Hongbin Zha
Rin-ichiro Taniguchi
Stephen Maybank (Eds.)

LNCS 5995

Computer Vision – ACCV 2009

9th Asian Conference on Computer Vision
Xi'an, September 2009
Revised Selected Papers, Part II

2 Part II

Springer

Lecture Notes in Computer Science 5995

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

Hongbin Zha Rin-ichiro Taniguchi
Stephen Maybank (Eds.)

Computer Vision – ACCV 2009

9th Asian Conference on Computer Vision
Xi'an, September 23-27, 2009
Revised Selected Papers, Part II

Springer

Volume Editors

Hongbin Zha
Peking University
Department of Machine Intelligence
Beijing, 100871, China
E-mail: zha@cis.pku.edu.cn

Rin-ichiro Taniguchi
Kyushu University
Department of Advanced Information Technology
Fukuoka, 819-0395, Japan
E-mail: rin@ait.kyushu-u.ac.jp

Stephen Maybank
University of London
Birkbeck College, Department of Computer Science
London, WC1E 7HX, UK
E-mail: sjmaybank@dcs.bbk.ac.uk

Library of Congress Control Number: 2010923506

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

ISSN 0302-9743
ISBN-10 3-642-12303-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-12303-0 Springer Berlin Heidelberg New York

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

Preface

It gives us great pleasure to present the proceedings of the 9th Asian Conference on Computer Vision (ACCV 2009), held in Xi'an, China, in September 2009. This was the first ACCV conference to take place in mainland China.

We received a total of 670 full submissions, which is a new record in the ACCV series. Overall, 35 papers were selected for oral presentation and 131 as posters, yielding acceptance rates of 5.2% for oral, 19.6% for poster, and 24.8% in total. In the paper reviewing, we continued the tradition of previous ACCVs by conducting the process in a double-blind manner. Each of the 33 Area Chairs received a pool of about 20 papers and nominated a number of potential reviewers for each paper. Then, Program Committee Chairs allocated at least three reviewers to each paper, taking into consideration any conflicts of interest and the balance of loads. Once the reviews were finished, the Area Chairs made summary reports for the papers in their pools, based on the reviewers' comments and on their own assessments of the papers.

The Area Chair meeting was held at Peking University, Beijing during July 6–7, 2009. Thirty-one Area Chairs attended the meeting. They were divided into eight groups. The reviews and summary reports for the papers were discussed within the groups, in order to establish the scientific contribution of each paper. Area Chairs were permitted to confer with pre-approved "consulting" Area Chairs outside their groups if needed. The final acceptance decisions were made at a meeting of all the Area Chairs. Finally, the Program Chairs drew up a single-track technical program which consisted of 12 oral sessions and three poster sessions for the three-day conference. We are glad to see that all of the oral speakers presented their papers at the conference.

The program included three plenary sessions in which world-leading researchers, Roberto Cipolla (University of Cambridge), Larry S. Davis (University of Maryland), and Long Quan (Hong Kong University of Science and Technology), gave their talks. We would like to thank them for their respective presentations on 3D shape acquisition, human tracking and image-based modeling, which were both inspiring and entertaining.

A conference like ACCV 2009 would not be possible without the concerted effort of many people and the support of various institutions. We would like to thank the ACCV 2009 Area Chairs and members of the Technical Program Committee for their time and effort spent in reviewing the submissions. The local arrangement team, led by Yanning Zhang, did a terrific job in organizing the conference. We also thank Katsushi Ikeuchi, Tieniu Tan, and Yasushi Yagi, whose help was critical at many stages of the conference organization. Last but

not least, we would like to thank all of the attendees of the conference. Due to their active participation, this was one of the most successful conferences in the history of the ACCV series.

December 2009

Hongbin Zha
Rin-ichiro Taniguchi
Stephen Maybank

Organization

Honorary Chairs	Yunhe Pan (Chinese Academy of Engineering, China) Songde Ma (Institute of Automation, Chinese Academy of Science, China) Katsushi Ikeuchi (University of Tokyo, Japan)
General Chairs	Tieniu Tan (Institute of Automation, Chinese Academy of Science, China) Nanning Zheng (Xi'an Jiaotong University, China) Yasushi Yagi (Osaka University, Japan)
Program Chairs	Hongbin Zha (Peking University, China) Rin-ichiro Taniguchi (Kyushu University, Japan) Stephen Maybank (University of London, UK)
Organization Chairs	Yanning Zhang (Northwestern Polytechnical University, China) Jianru Xue (Xi'an Jiaotong University, China)
Workshop Chairs	Octavia Camps (Northeastern University, USA) Yasuyuki Matsushita (Microsoft Research Asia, China)
Tutorial Chairs	Yunde Jia (Beijing Institute of Technology, China)
Demo Chairs	Dacheng Tao (Nanyang Technological University, Singapore)
Publication Chairs	Ying Li (Northwestern Polytechnical University, China) Kaiqi Huang (Institute of Automation, Chinese Academy of Science, China)

Publicity Chairs

Bin Luo (Anhui University, China)
Chil-Woo Lee (Chonnam National University, Korea)
Hichem Sahli (Vrije University Brussel, Belgium)

Area Chairs

Noboru Babaguchi (Osaka University)
Horst Bischof (Technical University Graz)
Chu-Song Chen (Institute of Information Science, Academia Sinica)
Jan-Michael Frahm (University of North Carolina at Chapel Hill)
Pascal Fua (EPFL: Ecole Polytechnique Fédérale de Lausanne)
Dewen Hu (National University of Defense Technology)
Zhanyi Hu (Institute of Automation, Chinese Academy of Science)
Yi-ping Hung (National Taiwan University)
Ron Kimmel (Technion - Israel Institute of Technology)
Reinhard Klette (University of Auckland)
Takio Kurita (National Institute of Advanced Industrial Science and Technology)
Chil-Woo Lee (Chonnam National University)
Kyoung Mu Lee (Seoul National University)
Fei-Fei Li (Stanford University)
Zhouchen Lin (Microsoft Research Asia)
Kai-Kuang Ma (Nanyang Technological University)
P.J. Narayanan (International Institute of Information Technology, Hyderabad)
Nassir Navab (Technische Universität München)
Tomas Pajdla (Czech Technical University in Prague)
Robert Pless (Washington University)
Long Quan (The Hong Kong University of Science and Technology)
Jim Rehg (Georgia Institute of Technology)
Ian Reid (Oxford University)
Wildes Richard (York University)
Hideo Saito (Keio University)
Nicu Sebe (University of Amsterdam)
Peter Sturm (INRIA)

Akihiro Sugimoto (National Institute of Informatics)
David Suter (University of Adelaide)
Guangyou Xu (Tsinghua University)
Yaser Yacoob (University of Maryland)
Ming-Hsuan Yang (University of California at Merced)
Hong Zhang (University of Alberta)

Committee Members

Zhonghua Fu (Northwestern Polytechnical University, China)
Dongmei Jiang (Northwestern Polytechnical University, China)
Kuizhi Mei (Xi'an Jiaotong University, China)
Yuru Pei (Peking University, China)
Jinqiu Sun (Northwestern Polytechnical University, China)
Fei Wang (Xi'an Jiaotong University, China)
Huai-Yu Wu (Peking University, China)
Runping Xi (Northwestern Polytechnical University, China)
Lei Xie (Northwestern Polytechnical University, China)
Xianghua Ying (Peking University, China)
Gang Zeng (Peking University, China)
Xinbo Zhao (Northwestern Polytechnical University, China)
Jiangbin Zheng (Northwestern Polytechnical University, China)

Reviewers

Abou Moustafa Karim
Achard Catherine
Ai Haizhou
Alahari Karteek
Allili Mohand Said
Andreas Koschan
Aoki Yoshmitsu
Argyros Antonis
Arica Nafiz
Ariki Yasuo
Arslan Abdullah
August Jonas
Awate Suyash
Azevedo-Marques Paulo
Bagdanov Andrew
Bai Xiang
Bajcsy Peter
Baltes Jacky
Banerjee Subhashis
Barbu Adrian
Barnes Nick
Barreto Joao
Bartoli Adrien
Baudrier Etienne
Baust Maximilian
Beichel Reinhard
Beng-Jin Andrew Teoh
Benhimane Selim
Benosman Ryad
Bibby Charles
Bicego Manuele
Blekas Konstantinos
Bo Liefeng
Bors Adrian
Boshra Michael
Bouaziz Sofien
Bouguila Nizar
Boutemedjet Sabri
Branzan Albu Alexandra

Bremond Francois
Bronstein Alex
Bronstein Michal
Brown Matthew
Brown Michael
Brun Luc
Buckley Michael
Caballero Rodrigo
Caglioti Vincenzo
Cagniart Cedric
Camastra Francesco
Cao Liangliang
Cao Liangliang
Carneiro Gustavo
Carr Peter
Castellani Umberto
Cattin Philippe
Celik Turgay
Chan Kap Luk
Chandran Sharat
Chellappa Rama
Chen Haifeng
Chen Hwann-Tzong
Chen Jiun-Hung
Chen Jun-Cheng
Chen Ling
Chen Pei
Chen Robin Bing-Yu
Chen Wei-Chao
Chen Xilin
Chen Yixin
Cheng Jian
Cheng Jian
Cheng Shyi-Chyi
Chetverikov Dmitry
Chia Liang-Tien
Chien Shao-Yi
Chin Tat-jun
Chu Wei-Ta
Chuang Yung-Yu
Chung Albert
Civera Javier
Clipp Brian
Coleman Sonya
Costeira Joao Paulo
Cousty Jean
Csaba Beleznai
Dang Xin
Daras Petros
De La Torre Fernando
Deguchi Koichiro
Demirci Fatih
Demirdjian David
Deng Hongli
Deniz Oscar
Denzler Joachim
Derpanis Konstantinos
Derrode Stephane
Destefanis Eduardo
Dick Anthony
Didas Stephan
Dong qiulei
Donoser Michael
Doretto Gianfranco
Drbohlav Ondrej
Drost Bertram
Duan Fuqing
Dueck Delbert
Duric Zoran
Dutagaci Helin
Dutta Roy Sumantra
Dutta Roy
Dvornychenko Vladimir
Dyer Charles
Eckhardt Ulrich
Eigensatz Michael
Einhauser Wolfgang
Eroglu Erdem Cigdem
Escolano Francisco
Fan Quanfu
Fang Wen-Pinn
Farenzena MIchela
Fasel Beat
Feng Jianjiang
Feris Rogerio
Ferri Francesc
Fidler Sanja
Fihl Preben
Filliat David
Flitti Farid
Floery Simon
Forstner Wolfgang
Franco Jean-Sebastien
Fraundorfer Friedrich
Fritz Mario
Frucci Maria
Fu Chi-Wing
Fuh Chiou-Shann
Fujiyoshi Hironobu
Fukui Kazuhiro
Fumera Giorgio
Furst Jacob
Furukawa Yasutaka
Fusiello Andrea
Gall Juergen
Gallagher Andrew
Gang Li
Garg Kshitiz
Georgel Pierre
Gertych Arkadiusz
Gevers Theo
Gherardi Riccardo
Godil Afzal
Goecke Roland
Goshtasby A.
Gou Gangpeng
Grabner Helmut
Grana Costantino
Guerrero Josechu
Guest Richard
Guliato Denise
Guo Feng
Guo Guodong
Gupta Abhinav
Gupta Mohit
Hadjileontiadis Leontios
Hamsici Onur
Han Bohyung
Han Chin-Chuan
Han Joon Hee
Hanbury Allan
Hao Wei
Hassab Elgawi Osman
Hautiere Nicolas
He Junfeng

Heitz Fabrice
Hinterstoisser Stefan
Ho Jeffrey
Holzer Stefan
Hong Hyun Ki
Hotta Kazuhiro
Hotta Seiji
Hou Zujun
Hsiao JenHao
Hsu Pai-Hui
Hsu Winston
Hu Qinghua
Hu Weimin
Hu Xuelei
Hu Yiqun
Hu Yu-Chen
Hua Xian-Sheng
Huang Fay
Huang Kaiqi
Huang Peter
Huang Tz-Huan
Huang Xiangsheng
Huband Jacalyn
Huele Ruben
Hung Hayley
Hung-Kuo Chu James
Huynh Cong
Iakovidis Dimitris
Ieng Sio Hoi
Ilic Slobodan
Imiya Atsushi
Inoue Kohei
Irschara Arnold
Ishikawa Hiroshi
Iwashita Yumi
Jaeger Stefan
Jafari Khouzani Kourosh
Jannin Pierre
Jawahar C. V.
Jean Frederic
Jia Jiaya
Jia Yunde
Jia Zhen
Jiang Shuqiang
Jiang Xiaoyi
Jin Lianwen
Juneho Yi
Jurie Frederic
Kagami Shingo
Kale Amit
Kamberov George
Kankanhalli Mohan
Kato Takekazu
Kato Zoltan
Kawasaki Hiroshi
Ke Qifa
Keil Andreas
Keysers Daniel
Khan Saad-Masood
Kim Hansung
Kim Kyungnam
Kim Tae Hoon
Kim Tae-Kyun
Kimia Benjamin
Kitahara Itaru
Koepfler Georges
Koeser Kevin
Kokkinos Iasonas
Kolesnikov Alexander
Kong Adams
Konolige Kurt
Koppal Sanjeev
Kotsiantis Sotiris
Kr Er Norbert
Kristan Matej
Kuijper Arjan
Kukelova Zuzana
Kulic Dana
Kulkarni Kaustubh
Kumar Ram
Kuno Yoshinori
Kwolek Bogdan
Kybic Jan
Ladikos Alexander
Lai Shang-Hong
Lao Shihong
Lao Zhiqiang
Lazebnik Svetlana
Le Duy-Dinh
Le Khoa
Lecue Guillaume
Lee Hyoung-Joo
Lee Ken-Yi
Lee Kyong Joon
Lee Sang-Chul
Leonardo Bocchi
Lepetit Vincent
Lerasle Frederic
Li Baihua
Li Bo
Li Hongdong
Li Teng
Li Xi
Li Yongmin
Liao T.Warren
Lie Wen-Nung
Lien Jenn-Jier James
Lim Jongwoo
Lim Joo-Hwee
Lin Dahua
Lin Huei-Yung
Lin Ruei-Sung
Lin Wei-Yang
Lin Wen Chieh (Steve)
Ling Haibin
Liu Jianzhuang
Liu Ming-Yu
Liu Qingshan
Liu Qingzhong
Liu Tianming
Liu Tyng-Luh
Liu Xiaoming
Liu Xiuwen
Liu Yuncai
Lopez-Nicolas Gonzalo
Lu Juwei
Lu Le
Luo Jiebo
Ma Yong
Macaire Ludovic
Maccormick John
Madabhushi Anant
Manabe Yoshitsugu
Manniesing Rashindra
Marchand Eric

Marcialis Gian-Luca
Martinet Jean
Martinez Aleix
Masuda Takeshi
Mauthner Thomas
McCarthy Chris
McHenry Kenton
Mei Christopher
Mei Tao
Mery Domingo
Mirmehdi Majid
Mitra Niloy
Mittal Anurag
Miyazaki Daisuke
Moeslund Thomas
Monaco Francisco
Montiel Jose
Mordohai Philippos
Moreno Francesc
Mori Kensaku
Moshe Ben-Ezra
Mudigonda Pawan
Mueller Henning
Murillo Ana Cris
Naegel Benoit
Nakajima Shin-ichi
Namboodiri Anoop
Nan Xiaofei
Nanni Loris
Narasimhan Srinivasa
Nevatia Ram
Ng Wai-Seng
Nguyen Minh Hoai
Nozick Vincent
Odone Francesca
Ohnishi Naoya
Okatani Takayuki
Okuma Kenji
Omachi Shinichiro
Pack Gary
Palagyi Kalman
Pan ChunHong
Pankanti Sharath
Paquet Thierry
Park In Kyu
Park Jong-Il
Park Rae-Hong
Passat Nicolas
Patras Yiannis
Patwardhan Kedar
Peers Pieter
Peleg Shmuel
Pernici Federico
Pilet Julien
Pless Robert
Pock Thomas
Prati Andrea
Prevost Lionel
Puig Luis
Qi Guojun
Qian Zhen
Radeva Petia
Rajashekar Umesh
Ramalingam Srikumar
Ren Xiaofeng
Reyes Edel Garcia
Reyes Aldasoro Constantino
Ribeiro Eraldo
Robles-Kelly Antonio
Rosenhahn Bodo
Rosman Guy
Ross Arun
Roth Peter
Rugis John
Ruvolo Paul
Sadri Javad
Saffari Amir
Sagawa Ryusuke
Salzmann Mathieu
Sang Nong
Santner Jakob
Sappa Angel
Sara Radim
Sarkis Michel
Sato Jun
Schmid Natalia
Schroff Florian
Shahrokni Ali
Shan Shiguang
Shen Chunhua
Shi Guangchuan
Shih Sheng-Wen
Shimizu Ikuko
Shimshoni Ilan
Sigal Leonid
Singhal Nitin
Sinha Sudipta
Snavely Noah
Sommerlade Eric
Steenstrup Pedersen Kim
Sugaya Yasuyuki
Sukno Federico
Sumi Yasushi
Sun Feng-Mei
Sun Weidong
Svoboda Tomas
Swaminathan Rahul
Takamatsu Jun
Tan Ping
Tan Robby
Tang Chi-Keung
Tang Ming
Teng Fei
Tian Jing
Tian Yingli
Tieu Kinh
Tobias Reichl
Toews Matt
Toldo Roberto
Tominaga Shoji
Torii Akihiko
Tosato Diego
Trobin Werner
Tsin Yanghai
Tu Jilin
Tuzel Oncel
Uchida Seiichi
Urahama K
Urschler Martin
Van den Hengel Anton
Vasseur Pascal
Veeraraghavan Ashok
Veksler Olga
Vitria Jordi

Wagan Asim
Wang Hanzi
Wang Hongcheng
Wang Jingdong
Wang Jue
Wang Meng
Wang Sen
Wang Yunhong
Wang Zhi-Heng
Wei Hong
Whitehill Jacob
Wilburn Bennett
Woehler Christian
Wolf Christian
Woo Young Woon
Wu Fuchao
Wu Hao
Wu Huai-Yu
Wu Jianxin
Wu Yihong
Xiong Ziyou
Xu Ning
Xue Jianru
Xue Jianxia
Yan Shuicheng
Yanai Keiji
Yang Herbert
Yang Ming-Hsuan
Yao Yi
Yaron Caspi
Yeh Che-Hua
Yilmaz Alper
Yin Pei
Yu Tianli
Yu Ting
Yuan Baozong
Yuan Lu
Zach Christopher
Zha Zheng-Jun
Zhang Changshui
Zhang Guangpeng
Zhang Hongbin
Zhang Li
Zhang Liqing
Zhang Xiaoqin
Zhang Zengyin
Zhao Deli
Zhao Yao
Zheng Wenming
Zhong Baojiang
Zhou Cathy
Zhou Howard
Zhou Jun
Zhou Rong
Zhou S.
Zhu Feng
Zhu Wenjun
Zitnick Lawrence

Sponsors

Key Laboratory of Machine Perception (MOE), Peking University.
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.
National Natural Science Foundation of China.
Microsoft Research.
Fujitsu Inc.
Microview Inc.
Luster Inc.

Table of Contents – Part II

Poster Session 1: Stereo, Motion Analysis, and Tracking

Poster Session 2: Segmentation, Detection, Color and Texture

A Dynamic Programming Approach to Maximizing Tracks for Structure from Motion

Jonathan Mooser[1], Suya You[1], Ulrich Neumann[1], Raphael Grasset[2], and Mark Billinghurst[2]

[1] CGIT Lab
University of Southern California, Los Angeles, California
{mooser,suyay,uneumann}@graphics.usc.edu
[2] HITLabNZ
University of Canterbury, Christchurch, New Zealand
{raphael.grasset,mark.billinghurst}@hitlabnz.org

Abstract. We present a novel algorithm for improving the accuracy of structure from motion on video sequences. Its goal is to efficiently recover scene structure and camera pose by using dynamic programming to maximize the lengths of putative keypoint tracks. By efficiently discarding poor correspondences while maintaining the largest possible set of inliers, it ultimately provides a robust and accurate scene reconstruction. Traditional outlier detection strategies, such as RANSAC and its derivatives, cannot handle high dimensional problems such as structure from motion over long image sequences. We prove that, given an estimate of the camera pose at a given frame, the outlier detection is optimal and runs in low order polynomial time. The algorithm is applied on-line, processing each frame in sequential order. Results are presented on several indoor and outdoor video sequences processed both with and without the proposed optimization. The improvement in average reprojection errors demonstrates its effectiveness.

1 Introduction

Structure from motion refers to the problem of processing sets of images with the goal of modeling the underlying scene geometry while simultaneously determining camera locations. In principle, the task is straightforward, the relevant computational geometry having been well documented [1,2]. In practice, however, the problem is substantially more challenging.

In almost all cases, the computations rely on identifying feature correspondences between images. These typically consist of single points [2,3,4], but may include more complex features as well [5]. Difficulties arise in real world applications because some putative correspondences are inevitably incorrect. The main contribution of this work is the novel method by which inaccurate correspondences are identified and removed, thus maximizing the accuracy of the final reconstruction.

The proposed system receives its input as an ordered sequence of video frames. Because the baseline between consecutive frames is small, two or three

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 1–10, 2010.

frame 5 frame 24 frame 51 frame 59 frame 91

Fig. 1. A point that tracks accurately over some frames, but not over the entire sequence. The correspondence between frames 5 and 91 is poor. The correspondence between Frames 5 through 24 are accurate, however, as is the correspondence between frames 59 and 91.

consecutive frames will rarely provide an accurate reconstruction on its own, and in many cases will only capture a small part of the scene. At the same time, no feature is likely to appear throughout the entire sequence, so looking for correspondences between, say, the first and last frames is not feasible. Hartley and Zisserman observed that the problem of structure from motion from video sequences remains a "black art" [1, p. 452].

The feature correspondences considered here consist of single points tracked over time using sparse optical flow. That is to say, we detect a set of pixels in one frame then estimate their locations in subsequent frames by taking advantage of the relative similarity between consecutive images. Specifically, we begin by detecting keypoints in the first image using Shi and Tomasi's method of identifying trackable point [6]. Then a variation of the Lucas-Kanade Optical flow algorithm [7] based on image pyramids [8] is used to update their locations as the sequence progresses. The reliability of the optical flow process is further improved by using bi-directional filtering, as described in [9].

Using optical flow to generate correspondences has both advantages and drawbacks. Optical flow is generally reliable, with the correspondences between consecutive frames will seldom off by more than one or two pixels.

On the other hand, points tracked by optical flow have a tendency to drift over long sequences and thus introduce a unique challenge. Figure 1 illustrates an example. Over the course of 91 frames, a point drifts significantly from the side of the statue to the lawn in the background. Clearly, any reconstruction that depends on this correspondence will suffer as a result.

One could attempt to identify such points and exclude them from the computation, hoping that enough correspondences remain to reconstruct the scene. Besides the difficulty of automatically detecting tracking errors, the problem is that over long sequences, almost *all* tracked keypoints will experience some drift. Simply labeling points as inlier or outlier is thus of limited value.

With this in mind, we set out to perform a somewhat more ambitious optimization. Examining Figure 1, the keypoint in question stays fixed to the same part of the statue between frames 5 and 24. It then drifts to another part of the

dog's head and finally onto the lawn in the background. Finally between frames 59 and 91, tracking regains stability, the keypoint remaining fixed to the same point on the grass. So while the keypoint is not useful over the entire sequence, it is useful for certain windows of time. Our goal is to determine which sets of frames contain accurate tracking for each keypoint and use only those.

Section 3 describes an algorithm for performing this optimization. This algorithm, called *subtrack optimization*, represents the main contribution of this work. Based on dynamic programming, it is guaranteed to output an optimal solution, and does so in low order polynomial time.

Section 4 discusses how this algorithm can be incorporated into a high level structure from motion system. It sequentially processes video frames to generate an accurate sparse scene structure as well as a camera pose at each frame. The proposed system offers several key advantages. One is that it operates as an online algorithm, which is to say it produces a solution for the first n frames before considering frame $n+1$. It also does not depend on extra hardware such as inertial sensors or a calibrated stereo rig; it relies only on a single calibrated camera. It assumes that the scene is rigid, but makes no other assumptions about scene structure or camera motion.

We present results on several real-world video sequences in section 5. Each sequence is processed with and without subtrack optimization. Numerical results demonstrate that the optimization substantially improves the quality of the overall reconstruction.

2 Related Work

Over the years, structure from motion has remained amongst the most widely studied topics in computer vision [2,3,4,5,10,11,12]. The high level of interest is hardly surprising, as it provides an invaluable tool in numerous application domains.

Snavely, *et al.*, for example use structure from motion as the basis for a virtual tourism application [3]. Zhu, *et al.*, describe a navigation system that uses structure from motion to build a database of landmarks, which can later be used to recover the location of an image in a large scale environment [12].

The present work focuses on sparse structure from motion, which is to say only a small set of landmark features from the target scene are modeled. In general, accurate sparse structure is a precondition for computing dense structure, which builds fully textured surfaces [10,11,2].

The goal of the present work is to separate accurate feature correspondences from inaccurate ones, and can thus be viewed as a kind of outlier detection. When only two views are available, outliers can be identified using random sampling methods such as RANSAC [13] or the more recent MLESAC [14] algorithm. The video sequences considered here, however, consist of hundreds of frames, leading to a very high dimensional solution space. RANSAC and its derivatives are not feasible as a means to optimize over all variable. While some systems use RANSAC to detect outliers between two or three consecutive frames, [2,4,15], it cannot be applied to an entire sequence at once.

Structure from motion algorithms often include bundle adjustment as a final step [1,2,4]. Traditionally, bundle adjustment is applied to the entire sequence at once, in which case it consumes most of the processing time and precludes online processing. More efficient versions can apply bundle adjustment to a few frames at a time. Still, bundle adjustment assumes that the putative 2D correspondences are nearly correct, and cannot determine when keypoints begin to drift, as we do here.

It is also important to draw a distinction between our algorithm and those that use assumptions about the structure of the scene, such as planar surfaces [11]. Our only assumption is that the scene is rigid.

Buchanan and Fitzgibbon [16] describe an approach to feature tracking that, like the method proposed here, is based on dynamic programming. That work, however, focuses on purely two-dimensional tracking. The algorithm described here is specifically designed to recover three-dimensional structure.

3 The Subtrack Optimization Algorithm

3.1 Terminology and Problem Definition

In order to describe the details of the optimization algorithm, the following terminology will be useful.

A *keypoint* will refer to a single point feature in a single image. Keypoints fall into two categories. Those that are initially identified by the detection process are referred to as *detected keypoints*. Those that have been tracked from the previous frame will be called *tracked keypoints*.

Points in 3D space from which keypoints arise are *structure points*. Conceptually, each keypoint represents a ray in 3D space, so there are an infinite number of possible structure points corresponding to a given keypoint. Any structure point that projects onto a keypoint within some margin of error, will be considered a *valid structure point* for that keypoint.

A detected keypoint along with all of the tracked keypoints generated from it are collectively referred to as a *keypoint track*. A keypoint track never skips frames; if the optical flow process fails to track a particular keypoint then the corresponding keypoint track ends. A keypoint track will also never contain more than one keypoint in any given frame.

Under ideal circumstances, all keypoints in a given track will share some valid structure point. Due to tracking errors, however, this will seldom be the case for long tracks. As illustrated in the examples from section 1, however, it will often be the case that a subset of a track's keypoints does in fact have a common valid structure point. Any set of two or more keypoints from consecutive frames of the same track will be called a *subtrack*. A subtrack whose keypoints share a valid structure point will be deemed *consistent*.

Using this terminology, the goal of the optimization algorithm is as follows:

> *Given a keypoint track and a camera matrix at each frame, find the partitioning that produces the longest possible disjoint consistent subtracks*

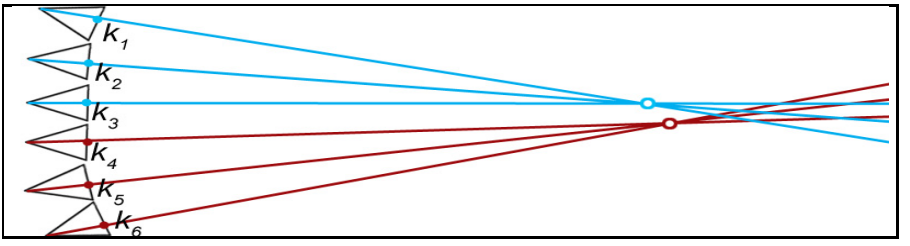


Fig. 2. A hypothetical keypoint track with six keypoints. On the left are six locations of a camera as it moves from the top downward. Each keypoint corresponds to a ray in space. The six rays do not meet at a single point, so there is no structure point that is valid for the entire track. However, subtracks $k_{1,2,3}$ and $k_{4,5,6}$ do have valid structure points. The goal of the optimization algorithm is to reliably perform this partitioning.

Favoring fewer, longer subtracks is important because it ensures that they span as wide a baseline as possible. If overly aggressive in partitioning a keypoint track, we risk losing valuable information and compromising the accuracy of the resulting structure.

To measure the consistency of a subtrack, we define an error function, $E(k_{a,b})$, as the RMS reprojection error generated by the optimal structure point $\mathbf{X}(k_{a,b})$ for subtrack $k_{a,b}$. If the subtrack is consistent $E(k_{a,b})$ will be small.

A naive approach might look to simply find subtracks that individually minimize $E(k_{a,b})$, which could be achieved by making a large number of short subtracks. This, however, would ignore the ultimate goal of maximizing subtrack lengths. To account for this constraint, a constant term δ is introduced representing the penalty of adding a new subtrack. For a keypoint track of length n, a given partitioning, $p = \{k_{1,a}, k_{a+1,b}, \ldots, k_{c,d}, k_{d+1,n}\}$, thus incurs a total cost of

$$C(p) = \sum_{k_{a,b} \in p} (\delta + E(k_{a,b})) \tag{1}$$

The optimal partitioning is the one that minimizes $C(p)$. Clearly, the number of possible partitionings is exponential in n, so a brute force search would be intractable. We will show, however, that it is it is possible to find an absolute minimum in $O(n^3)$ time using a dynamic programming algorithm.

3.2 A Dynamic Programming Solution

The insight behind the algorithm is the following lemma:

Lemma 1. *if $p = \{k_{1,a}, \ldots, k_{c,d}, k_{d+1,n}\}$ is the optimal partitioning of k, then $q = \{k_{1,a}, \ldots, k_{c,d}\}$ is the optimal partitioning of the subtrack $k_{1,d}$.*

Proof. Assume that q is not the optimal partitioning for $k_{1,d}$. That is to say there exists some other partitioning q' such that $C(q') < C(q)$. Now let p' be the partitioning of k given by $p' = \{q', k_{d+1,n}\}$. Because

$$\begin{aligned} C(p) &= C(q) + \delta + E(k_{d,n}) \\ &\text{and} \\ C(p') &= C(q') + \delta + E(k_{d,n}) \end{aligned} \tag{2}$$

we know that $C(p') < C(p)$, which implies that p is not optimal. □

Let $\hat{p}_n$ be the optimal partitioning of $k_{1,n}$. Its cost can now be defined recursively as

$$\begin{aligned} C(\hat{p}_0) &= 0 \\ C(\hat{p}_1) &= \delta \\ C(\hat{p}_n) &= \min_{1 \le a < n} [C(\hat{p}_{a-1}) + \delta + E(k_{a,n})] \end{aligned} \tag{3}$$

Formally, $\hat{p}_0$ and $\hat{p}_1$ are undefined because the corresponding subtracks, $k_{1,0}$, and $k_{1,1}$ do not exist; a subtrack must span at least two keypoints. Their costs $C(\hat{p}_0)$ and $C(\hat{p}_1)$ are explicitly defined, however, as a base case for the recursion.

A dynamic programming algorithm can efficiently compute $C(\hat{p}_n)$ for any value of n by evaluating the recursion from the bottom up. First compute $C(\hat{p}_2) = E(k_{1,2}) + \delta$, then $C(\hat{p}_b)$ for $b = 3, 4, \ldots, n$. At each iteration $C(\hat{p}_a)$ is known for all $a < b$, so equation (3) can be directly applied, computing $E(k_{a,b})$ for all a. The complexity of computing $E(k_{a-b})$ is linear in the length of k_{a-b}, so iteration b requires $O(b^2)$ time. Processing an entire keypoint track of length n thus requires $O(n^3)$ time.

The algorithm, as described, finds the cost of the optimal partition. From this, finding the partition itself is straightforward. The simplest way is to keep track of the values of a that produce the minimum value of $C(\hat{p}_b)$ for each b. Using these stored values, the algorithm can work backward from n to piece together the optimal partitioning.

Although the final partition is optimal in that it minimizes (1), it is not necessarily the case that each subtrack is consistent. Recall the ultimate goal of finding long consistent subtracks. After optimizing each keypoint track, those subtracks spanning at least three frames and having $E(k_{a,b}) < 1.0$ are deemed consistent; all others are deemed inconsistent. Only the structure points corresponding to consistent subtracks are included in the final reconstruction, as explained in the next section.

4 The Complete Structure from Motion Process

The previous section addressed the problem of optimally partitioning a single keypoint track. We will now show how this can be incorporated into a larger structure form motion system involving many tracks over long video sequences. The system will function as an online algorithm, computing reconstruction for the first n frames before frame $n + 1$ is considered.

From the first frame, a set of keypoints will be detected, each instantiating a keypoint track. As subsequent frames are processed, optical flow will be applied to extend existing tracks. In addition, new keypoint tracks will be periodically

added to the existing ones by rerunning the detection process. In our implementation, new keypoints are detected every seven frames, with the total number of keypoints in any frame never allowed to exceed 300.

Using the first frame and some other suitable frame early in the sequence, along with the known camera intrinsics, an essential matrix is fit using RANSAC. This, in turn, is used to estimate camera poses for the first few frames.

This provides enough information to run the subtrack optimization algorithm on each keypoint track. Because the tracks at this point will be short, most will consist of a single subtrack. In any case, each subtrack will be deemed consistent or inconsistent; the consistent subtracks will have a valid structure point which will be added to the reconstruction.

As each new frame is processed, optical flow is again used to extend all current tracks. At this point the system assumes that all consistent subtracks that ended at the previous frame will remain consistent through the current frame. Because all of those subtracks are associated with known structure points, they provides enough information to compute the camera pose of the new frame.

Having the new camera pose, the system now runs the subtrack optimization algorithm again. Structure points are updated, if necessary, for each consistent subtrack. If a previously inconsistent subtrack is now consistent, its structure point is added to the reconstruction. Likewise, if a previously consistent subtrack is now inconsistent, its structure point is removed.

Each frame is processed in this manner, first computing the camera pose then optimizing the subtracks to incorporate the new pose. The final output is a set of structure points along with a camera pose associated with every frame.

4.1 Performance Considerations

Computing $E(k_{a,b})$ and $\mathbf{X}(k_{a,b})$ for a general set of keypoints and cameras requires nonlinear optimization. However, this can be performed quickly because only the three components of $\mathbf{X}$ are allowed to vary; the cameras, in this case, remain fixed. From a reasonable estimate, the absolute minimum of $\mathbf{X}$ can be approximated very closely by a single iteration of the Levenberg-Marquardt algorithm. In practice, a good estimate is to consider only the subtrack's endpoints, k_a and k_b, and use linear triangulation, as described in [1].

The subtrack optimization runs in $O(n^3)$ time assuming that $\hat{p}_a$, is computed for all a. However, at the time that frame n is being processed $\hat{p}_a$ has already been computed for all $a < n$. By storing these values throughout the sequence, the processing time for each individual frame is reduced to $O(n^2)$.

Despite this improvement, the time required for each track still increases quadratically and eventually, over a long sequence, will become unacceptably slow. To keep the processing time approximately constant, the system imposes a maximum subtrack length of 30 frames. This effectively places an upper bound on the running time of the optimization algorithm by limiting the size of the search space needed to apply equation (3). While the result is no longer strictly optimal, 30 frames is generally long enough to produce an accurate structure point. The exact size of the limit can be adjusted to favor either speed or accuracy.

5 Results

We tested the complete system on two sequences, each consisting of at least 200 frames. For comparison, they were processed both with and without the subtrack optimization algorithm. When processing with the optimization, the constant δ was assigned a value of 2.0 pixels. Without the optimization, a keypoint track is extended until its reprojection error exceeds a threshold, also set to 2.0 pixels, and then terminated. In the non-optimized version, each keypoint track contains exactly one subtrack. In all cases no subtrack is allowed to exceed 30 frames.

Figures 3 and 4 show selected frames from both sequences along with the resulting reconstructions, including camera poses. For clarity, Only some cameras are rendered for clarity. Both cases present some inherent challenges. The paper house sequence in figure 3 is perhaps easier because the target object has a clear discernable texture. Note, however that the system successfully reconstructs part of the desktop surface, which has little or no texture. The tree sequence in figure 4 includes irregularly shaped plants and foliage, as well as objects at a wide range of distances.

Table 1. Average reprojection error in pixels for each of the test sequences

Sequence	**House**		**Tree**	
	without subtr. opt.	with subtr. opt.	without subtr. opt.	with subtr. opt.
Total Frames	202	202	262	262
Average Subtrack Length	25.26	28.05	22.13	24.78
RMS Reprojection Error	0.60	0.87	0.61	0.96

Table 1 shows the results of processing the sequences both with and without subtrack optimization. On both test sequences, applying the optimization substantially reduced the total reprojection error.

One might suspect that the reduction in total error was simply the result of creating shorter subtracks. If one test tends to generate subtracks that are much shorter, on average, than another test, then the first test will almost certainly return a smaller error. However, as shown in Table 1, the average subtrack length is actually longer when using subtrack optimization. To further emphasize this point, we plot reprojection error *as a function of subtrack length*. These results are shown in the graphs in figures 3 and 4. The graphs demonstrate that even when comparing subtracks of the same length, the subtrack optimization algorithm reduces the average reprojection error. It offers the dual advantages of producing subtracks that are longer (and thus span a wider baseline) yet more consistent in terms of reprojection error.

The non-optimized version will only stop tracking a keypoint when its error has already reached 2.0 pixels. At that point, it has likely already been drifting for several frames. The advantage of the subtrack optimization algorithm is that

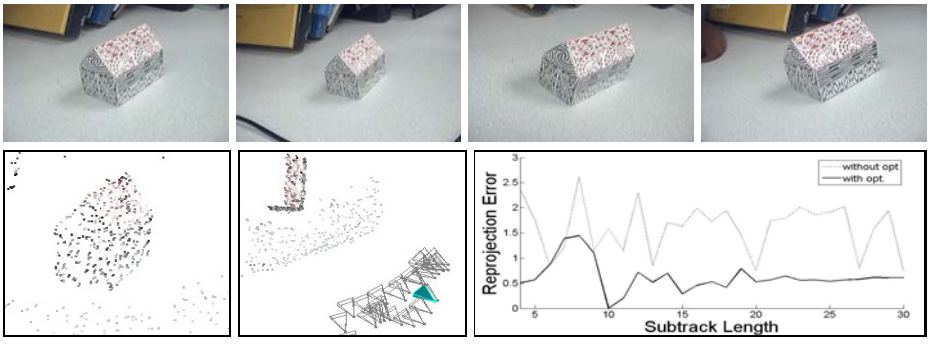


Fig. 3. A video sequence of a paper house and two views of the resulting reconstruction. The second is a top down view showing that the points on the vertical walls are coplanar, as expected.

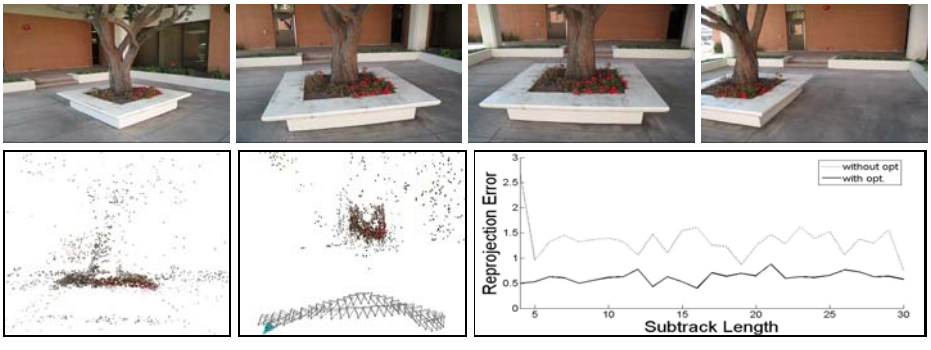


Fig. 4. A tree in the middle of a courtyard with two views of the resulting reconstruction. In the second view (top down) the square stone bench and square flower bed are clearly visible, as is a round space representing the volume occupied by the three trunk.

it identifies the precise moment when a keypoint begins drifting and partitions the track accordingly. The result is a more accurate reconstruction.

6 Conclusion

This paper has presented the subtrack optimization algorithm, which determines where to partition keypoint tracks so as to eliminate unreliable correspondences in structure from motion computations. Using dynamic programming, it performs this partitioning optimally. Because it makes few assumptions about the shape or appearance of the target scene, the optimization algorithm presented here is both effective and versatile.

References

1. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, New York (2003)
2. 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(3), 207–232 (2004)
3. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Papers, pp. 835–846 (2006)
4. 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, pp. 311–326. Springer, Heidelberg (1998)
5. Meltzer, J., Soatto, S.: Edge descriptors for robust wide-baseline correspondence (2008)
6. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
7. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
8. Bouguet, J.Y.: Pyramidal implementation of the Lucas-Kanade feature tracker. OpenCV library (2001), http://sourceforge.net/projects/opencvlibrary
9. Mooser, J., Wang, Q., You, S., Neumann, U.: Fast simultaneous tracking and recognition by incremental keypoint matching. In: 3D Data Processing, Visualization and Transmission (2008)
10. 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)
11. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions, pp. 1–8 (2007)
12. Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., Sawhney, H.: Real-time global localization with a pre-built visual landmark database (2008)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)
14. Torr, P.H.S., Zisserman, A.: Mlesac: a new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78(1), 138–156 (2000)
15. Nistér, D.: Preemptive ransac for live structure and motion estimation. In: IEEE International Conference on Computer Vision, p. 199 (2003)
16. Buchanan, A., Fitzgibbon, A.: Interactive feature tracking using k-d trees and dynamic programming. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 626–633 (2006)

Dense and Accurate Spatio-temporal Multi-view Stereovision

Jérôme Courchay[1], Jean-Philippe Pons[2],
Pascal Monasse[1], and Renaud Keriven[1]

[1] IMAGINE, 6 Av Blaise Pascal - Cité Descartes, Marne-la-Vallée, FR
{courchay,monasse,keriven}@imagine.enpc.fr
[2] CSTB, 290 route des Lucioles, Sophia Antipolis, FR
jean-philippe.pons@cstb.fr

Abstract. In this paper, we propose a novel method to simultaneously and accurately estimate the 3D shape and 3D motion of a dynamic scene from multiple-viewpoint calibrated videos. We follow a variational approach in the vein of previous work on stereo reconstruction and scene flow estimation. We adopt a representation of a dynamic scene by an animated mesh, i.e. a polygonal mesh with fixed connectivity whose time-varying vertex positions sample the trajectories of material points. Interestingly, this representation ensures a consistent coding of shape and motion by construction. Our method accurately recovers 3D shape and 3D motion by optimizing the positions of the vertices of the animated mesh. This optimization is driven by an energy function which incorporates multi-view and inter-frame photo-consistency, smoothness of the spatio-temporal surface and of the velocity field. Central to our work is an image-based photo-consistency score which can be efficiently computed and which fully handles projective distortion and partial occlusions. We demonstrate the effectiveness of our method on several challenging real-world dynamic scenes.

Keywords: Spatio-temporal stereovision, Scene flow, Motion capture.

1 Introduction

In recent years, several methods for automatic generation of complete spatio-temporal models of dynamic scenes from multiple videos have been proposed [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]. In particular, the most recent ones have proven effective for full-body marker-less motion capture, yielding visually impressive results. However, when taking a closer look at the aforementioned techniques, it becomes apparent that very few of them achieve a desirable *coupled, dense and accurate 3D shape and 3D motion estimation.*

Accurate 3D shape. Many recent techniques still produce an approximate geometry: free-form deformation of a template body model [2,11,15], visual hull [1,3,15], Laplacian deformation of a laser scan of the initial pose [4,5]. These methods are unable to recover genuine geometric details such as facial expressions and clothing folds and wrinkles.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 11–22, 2010.

Accurate 3D motion estimation is crucial in some applications like motion transfer and time interpolation. Also, a coarse motion estimation precludes the enforcement of temporal consistency constraints during coupled shape and motion estimation. However, in most existing performance capture techniques, 3D scene flow [16], i.e. the dense 3D motion field of the scene, is not accurately estimated. Often, it is interpolated from sparse 3D correspondences [3,4,12]. Some methods do not address 3D motion estimation whatsoever: [7] uses a four-dimensional level set representation which, beyond its very high computational and memory requirements, does not encode 3D correspondence. [10,15] produce animated meshes but, despite appearances, the underlying 3D correspondences are purely artifactual.

Coupled 3D shape and 3D motion estimation allows to exploit their redundancy, and has long been recognized [17] as a desirable way to improve their performance. However, most marker-less motion capture methods fail to integrate spatio-temporal consistency constraints. In [3,12,13], shape is computed independently in each time frame, prior to motion estimation. In [9], shape and motion are estimated sequentially, not simultaneously. In [5], an initial mesh is propagated by 3D scene flow, under silhouette constraints, but without any stereo cues; as a result, this method suffers from temporal drift. The latter is circumvented in [4] by substituting sparse 3D correspondences for dense 3D scene flow, but neither shape or motion are accurate enough to allow enforcing spatio-temporal consistency. In [1,7], a certain degree of spatio-temporal coherence is obtained through four-dimensional representations, but as these representations do not encode temporal correspondence, they cannot exploit inter-frame matching constraints. In [14], shape and motion are estimated simultaneously using a plane-sweep carving algorithm in a 6D space, but this approach has a very high computational and memory cost, is limited to two frames, and is unable to enforce the smoothness of the recovered shape and motion.

Thus, to our knowledge, two methods [6,8] achieve this highly desirable coupled, dense and accurate 3D shape and 3D motion estimation. In [8], shape and motion are represented through the detail coefficients of a time-varying subdivision surface. The latter coefficients are estimated by simultaneously optimizing multi-view and inter-frame photo-consistency. However, the non-linearity of the chosen multi-resolution representation makes this optimization intricate. Also, the required motion initialization relies on the spatio-temporal derivatives of the input images, thereby making it applicable mainly to slowly-moving Lambertian scenes under constant illumination.

[6] is the only work to date which can handle complex real-world dynamic scenes. Despite the effectiveness of this method, we believe that the expansion framework used does not allow to take into account the full visibility depending on occluding patch not computed yet.

In this paper, we propose a novel method to simultaneously and accurately estimate the 3D shape and 3D motion of a dynamic scene from multiple-viewpoint videos. First, we follow a **variational approach** in the vein of previous work on stereo reconstruction and scene flow estimation [9,17,18,19,20,21]. None of

these methods fits our applications in their current state: most are limited to a single time-varying depth map of the scene [17,18,19,20,21], while others do not enforce spatio-temporal consistency constraints [9,19].

Second, we adopt a representation of a dynamic scenes by an **animated mesh**, i.e. a polygonal mesh with fixed connectivity whose time-varying vertex positions sample the trajectories of material points. Interestingly, this representation ensures a consistent coding of shape and motion by construction. It is widely used in computer graphics, especially in computer animation. It is also popular for performance capture from video [3,4,5,6,10,11,15] or from time-varying point clouds [22,23] (the latter being obtained from video or from fast 3-D scanning hardware).

Our method accurately recovers 3D shape and 3D motion by optimizing the positions of the vertices of the animated mesh. This optimization is driven by an energy function which incorporates multi-view and inter-frame photo-consistency, smoothness of the spatio-temporal surface and of the velocity field. Central to our work is an image-based photo-consistency score which can be efficiently computed and which fully handles projective distortion and partial occlusions, in the spirit of [9].

The rest of this article is organized as follows. In Section 2, we describe in detail the discrete geometric representation, the variational formulation, the energy function and the associated minimization procedure which constitute our approach. In Section 3, we discuss implementation aspects and we demonstrate the effectiveness of our method on several challenging real-world dynamic scenes.

2 Our Approach

2.1 Discretize Then Optimize

An overwhelming majority of variational methods in this area [9,17,18,19,21] and more generally in computer vision, rely on an *optimize then discretize* approach: an energy functional depending on a continuous infinite-dimensional spatio-temporal representation is considered, the gradient of this energy functional is computed analytically, then the obtained evolution flow is discretized.

In contrast, we adopt a *discretize then optimize* approach: we define an energy function depending on a discrete finite-dimensional spatio-temporal representation, and we use standard non-convex optimization tools. The benefits of this approach have long been recognized in mesh processing, but have seldom been demonstrated in computer vision [24,25,26]. Thus, the choice of an adequate discrete spatio-temporal representation is crucial in our work.

2.2 Animated Mesh Representation

In our context, animated polygonal meshes present many significant advantages. Compared to unrelated meshes at different time instants, they are more compact, easier to store and to manipulate. They provide a direct access both to the shape

of the scene at a given time instant, and to motion trajectories. 3D shape and 3D motion are mutually consistent by construction.

Their fixed topology may be regarded as a limitation, as argued in [12]. We believe that it is not, since the human body has a constant - spherical, if disregarding pierces - topology. It is questionable to treat a character with hands on hips as a genus-2 torus. It should rather be regarded as a topological sphere with some temporary contact regions.

Furthermore, let us mention that our method is not limited to a spherical topology: while the topology of the animated mesh is constant *across time*, we are able to modify it *across our optimization process* using a spatio-temporal version of Delaunay deformable models [27].

2.3 Variational Formulation

In the following, we consider a dynamic scene, imaged by N calibrated and synchronized video sequences composed of T frames, and represented by an animated polygonal mesh with K vertices. We note:

- $I_{i,t} : \Omega_i \subset \mathbb{R}^2 \to \mathbb{R}^d$, $i \in \{1..N\}$, $t \in \{1..T\}$ the input images. In practice $d = 1$ for grayscale images and $d = 3$ for color images.
- $\boldsymbol{X} = \{\boldsymbol{x}_{k,t},\, k \in \{1..K\},\, t \in \{1..T\}\}$ the 3D positions of the vertices of the animated mesh at the different time instants,
- $\boldsymbol{X}_t = \{\boldsymbol{x}_{k,t},\, k \in \{1..K\}\}$ the t^{th} temporal slice of the animated mesh.

In the sequel, by a slight abuse of notation, we indistinctly use $\boldsymbol{X}$ and $\boldsymbol{X}_t$ to refer to the animated mesh and to the positions of its vertices.

The energy to minimize with respect to $\boldsymbol{X}$ is composed of a data attachment term, of a regularization term for the spatio-temporal surface and of a regularization term for the velocity field:

$$E(\boldsymbol{X}) = E_D(\boldsymbol{X}) + \lambda_S E_S(\boldsymbol{X}) + \lambda_V E_V(\boldsymbol{X})\ . \tag{1}$$

E_D encourages multi-view and frame-to-frame matching consistency. It is defined as the sum over camera pairs (i, j) and pairs of time frames (t, u) of a dissimilarity measure between image $I_{i,t}$ and the reprojection of $I_{j,u}$ via the animated mesh. The detailed description of this term is left to Section 2.4.

E_S favors the regularity of the spatio-temporal surface. We use the total area of the animated mesh. The minimization of this term by gradient descent yields a discrete version of the well known mean curvature motion, which we implement as described in [28].

E_V penalizes rapid variations of the velocity field along the animated mesh. It is the total squared L^2 norm over the animated mesh of the gradient of the velocity field. The detailed description of this term is left to Section 2.5.

We minimize the above energy function using a standard gradient descent on the spatio-temporal positions $\boldsymbol{X}$. In order to avoid unwanted local minima, we resort to a multi-resolution and chronological scheme. We first optimize the first two frames of a low-resolution animated mesh using low-resolution versions

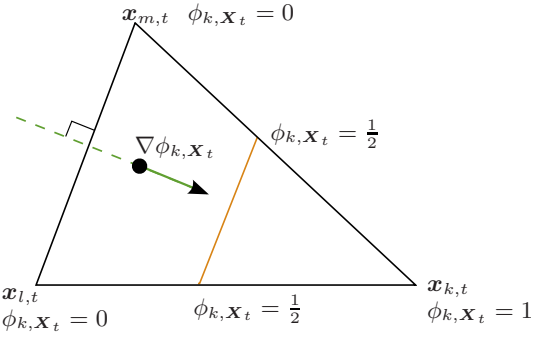


Fig. 1. Finite element representation over a facet (k, l, m) of the animated mesh

of input images. Then we initialize an additional time frame by extrapolating 3D position from speed and acceleration of previous frames. We iteratively add time frames, and optimize the sequence using a sliding time window of a few frames, until we reconstruct the whole temporal sequence at low resolution. We then refine the obtained spatio-temporal mesh with increased image and mesh resolutions, until we reach the desired accuracy.

2.4 Data Attachment Term

The formal definition of E_D and of its gradient requires some additional notations. The perspective projection performed by camera i is denoted by $\Pi_i : \mathbb{R}^3 \to \mathbb{R}^2$. Our method takes into account the visibility of the surface points. We refer to $\boldsymbol{X}_{i,t}$ as the part of the temporal slice $\boldsymbol{X}_t$ visible in image i. The back-projection of a point of camera i on the animated mesh at frame t is denoted by $\Pi^{-1}_{i,\boldsymbol{X}_t} : \Pi_i(\boldsymbol{X}_t) \to \boldsymbol{X}_{i,t}$.

We also define 3D transport functions $T_{\boldsymbol{X}_t \rightharpoonup \boldsymbol{X}_u}$ that map points in $\boldsymbol{X}_t$ to points in $\boldsymbol{X}_u$. This can be written formally using the linear finite-element representation depicted in Figure 1. For each vertex k of the animated mesh at some time frame t, we define a basis function $\phi_{k,\boldsymbol{X}_t}$ such that (i) $\phi_{k,\boldsymbol{X}_t}(\boldsymbol{x}_{k,t}) = 1$ (ii) $\forall l \neq k$, $\phi_{k,\boldsymbol{X}_t}(\boldsymbol{x}_{l,t}) = 0$ (iii) $\phi_{k,\boldsymbol{X}_t}$ varies linearly inside the triangular facets adjacent to the k^{th} vertex, and cancels outside this ring. We then have at pixel p_i in image i:

$$T_{\boldsymbol{X}_t \rightharpoonup \boldsymbol{X}_u} = \sum_k \boldsymbol{x}_{k,u}\, \phi_{k,\boldsymbol{X}_t} \ . \tag{2}$$

In a simpler way we can say that the back-projection Y_t of pixel p_i lies on a triangular facet f and has barycentric coordinates $\phi_{l,\boldsymbol{X}_t}(Y_t)$ at time t, l being a vertex of f. So the position of this particle at time u is $Y_u = \sum_{l \in f} \boldsymbol{x}_{l,u}\, \phi_{l,\boldsymbol{X}_t}(Y_t)$, that is $Y_u = \sum_{k \in \boldsymbol{X}} \boldsymbol{x}_{k,u}\, \phi_{k,\boldsymbol{X}_t}(Y_t)$ since $\phi_{k,\boldsymbol{X}_t}(Y_t)$ cancels if vertex k is outside facet f.

Finally, we define image transport functions $T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)}$ which map positions in $I_{i,t}$ to positions in $I_{j,u}$ via the animated mesh:

$$T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)} = \Pi_j \circ T_{\boldsymbol{X}_t \rightharpoonup \boldsymbol{X}_u} \circ \Pi^{-1}_{i,\boldsymbol{X}_t} \ . \tag{3}$$

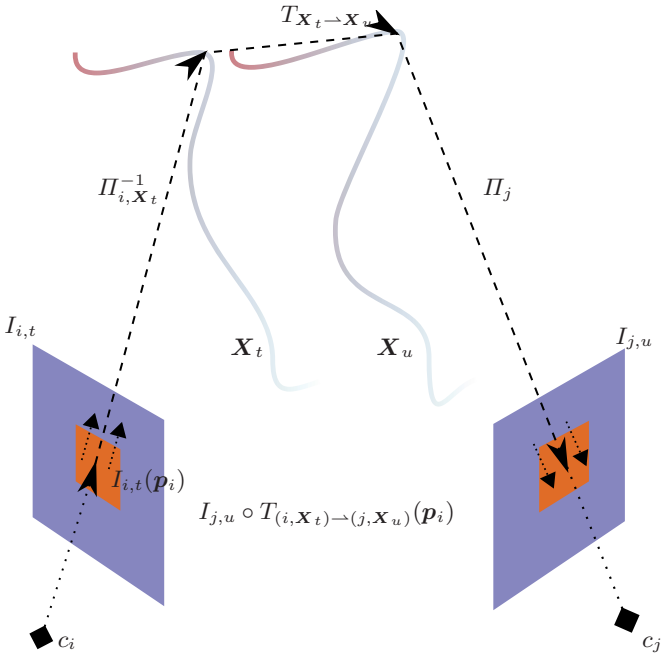


Fig. 2. Reprojection of image j at time u in image i at time t via the animated mesh

With these notations in hand the reprojection of image j at time u in image i at time t via the animated mesh writes $I_{j,u} \circ T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)}$. This is illustrated in Figure 2.

The data attachment term is the sum over oriented camera pairs (i, j) and oriented pairs (t, u) of time frames of a dissimilarity measure M between image $I_{i,t}$ and the above defined reprojection of $I_{j,u}$ via the animated mesh. The dissimilarity is computed only over the region of image plane i where both images are defined, i.e. after discarding semi-occluded regions. This image region writes $\Pi_i \left(\boldsymbol{X}_{i,t} \cap T_{\boldsymbol{X}_u \rightharpoonup \boldsymbol{X}_t}(\boldsymbol{X}_{j,u}) \right)$. More clearly, pixel p_i in image i is visible in both images, if its back-projection lies on the surface at time t, and this point on the surface once transported at time u is visible (nore occluded, nore outside the image frame) in image $I_{j,u}$. This visible image region is computed before each optimization step on graphics hardware. For conciseness, we will omit it in the equations below:

$$E_D(\boldsymbol{X}) = \sum_{i,j} \sum_{t,u} M \left[I_{i,t} \ , \ I_{j,u} \circ T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)} \right] \ . \tag{4}$$

We now compute the partial derivative of this energy term with respect to the variation of a single position $\boldsymbol{x}_{k,t}$ of the animated mesh. First, we note that the only oriented pairs of time frames affected by such a variation are (u, t) and (t, u), $u \in \{1..T\}$. Second, when the animated mesh moves, the reprojected

image changes. Hence the partial derivative of E_D involves the derivative of the similarity measure M with respect to its second argument, denoted by $\partial_2 M$.

Using the chain rule, and after some index manipulations, we get:

$$\frac{\partial E_D}{\partial \boldsymbol{x}_{k,t}} = \sum_{i,j} \sum_{u} \int_{\Omega_i} \partial_2 M \left[I_{i,t} \, , \, I_{j,u} \circ T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)} \right] DI_{j,u} \, \frac{\partial T_{(i,\boldsymbol{X}_t) \rightharpoonup (j,\boldsymbol{X}_u)}}{\partial \boldsymbol{x}_{k,t}} (\boldsymbol{p}_i) \, d\boldsymbol{p}_i + \int_{\Omega_j} \partial_2 M \left[I_{j,u} \, , \, I_{i,t} \circ T_{(j,\boldsymbol{X}_u) \rightharpoonup (i,\boldsymbol{X}_t)} \right] DI_{i,t} \, \frac{\partial T_{(j,\boldsymbol{X}_u) \rightharpoonup (i,\boldsymbol{X}_t)}}{\partial \boldsymbol{x}_{k,t}} (\boldsymbol{p}_j) \, d\boldsymbol{p}_j \ , \quad (5)$$

where $DI_{.,.}$ denotes the Jacobian matrices of the input images. For conciseness, we have omitted the points where the latter are evaluated in the above equation.

As regards the quantities $\frac{\partial T_{\ldots}}{\partial \boldsymbol{x}_{k,t}}$, we can make several observations. First, they are purely geometric, i.e. independent of image data. Second, they cancel outside the ring of triangular facets adjacent to the k^{th} vertex. Hence, despite appearances, integration is performed only over the visible projection of this ring in the different images, not over the full image domains. Third, these quantities involve the normal of the triangular facet visible at pixel p_i, and the barycentric coordinate of $\boldsymbol{x}_{k,t}$ in this facet. Complete expressions can be obtained using a non trivial geometric reasoning. We refer the interested reader to [29], where the detailed numerical computation, but also an additional intuitive solving are proposed. The numerical solving, mainly consist in computing how barycentric coordinates change for a small perturbation of the surface.

2.5 Velocity Field Regularization Term

The velocity field is unambiguously encoded by the animated mesh $\boldsymbol{X}$. Specifically, it is a continuous and piecewise linear vector field $\boldsymbol{X}_t \rightarrow \mathbb{R}^3$ defined by

$$\boldsymbol{v}_{\boldsymbol{X},t}(\boldsymbol{x}) = T_{\boldsymbol{X}_t \rightharpoonup \boldsymbol{X}_{t+1}}(\boldsymbol{x}) - \boldsymbol{x} \ , \quad (6)$$

or equivalently by

$$\boldsymbol{v}_{\boldsymbol{X},t} = \sum_k (\boldsymbol{x}_{k,t+1} - \boldsymbol{x}_{k,t}) \, \phi_{k,\boldsymbol{X}_t} \ . \quad (7)$$

The velocity field regularization term writes:

$$E_V(\boldsymbol{X}) = \sum_t \int_{\boldsymbol{X}_t} \|\nabla \boldsymbol{v}_{\boldsymbol{X},t}(\boldsymbol{x})\|^2 d\boldsymbol{x} \ . \quad (8)$$

To simplify this expression, we use the fact that $\nabla \phi_{k,\boldsymbol{X}_t}$ is constant in each triangular facet f of $\boldsymbol{X}_t$ and equals $\frac{\boldsymbol{h}_{k,f}}{\|\boldsymbol{h}_{k,f}\|^2}$, where $\boldsymbol{h}_{k,f}$ is triangle's height going through vertex k. A_f being the area of f, the energy term becomes:

$$E_V(\boldsymbol{X}) = \sum_t \sum_{f \in \boldsymbol{X}_t} A_f \left\| \sum_{k \in f} \frac{\boldsymbol{h}_{k,f}}{\|\boldsymbol{h}_{k,f}\|^2} (\boldsymbol{x}_{k,t+1} - \boldsymbol{x}_{k,t}) \right\|^2 \ . \quad (9)$$

If we neglect the variation of $\boldsymbol{h}_{k,f}$ with respect to vertex displacement, the partial derivatives $\frac{\partial E_V}{\partial \boldsymbol{x}_{k,t}}$ of this energy term can now easily be derived.

3 Numerical Experiments

3.1 Implementation Aspects

The computation of image reprojections via the animated mesh and of the gradient of the data attachment term are the most expensive parts of our algorithm. Hence, they are implemented on GPU using the OpenGL API and the Cg shading language.

In all our experiments, we choose the opposite of normalized cross correlation as the image dissimilarity measure M, in order to accommodate moving shadows and time-varying lighting conditions.

The storage of the animated mesh and the computation of spatio-temporal smoothing terms are based on the C++ Computational Geometry Algorithms Library (CGAL)[1].

The resolution of the mesh is controlled by a lower and an upper edge length thresholds, that are applied to the whole time sequence: an edge is bisected if it is longer than the upper threshold in *at least one* time frame; an edge is collapsed if it is shorter than the lower threshold in *all* time frames. The topology of the mesh is automatically corrected when needed by applying Delaunay deformable models [27] to the coordinates of the animated mesh at a reference time frame. The user chooses a reference frame that reflects the actual topology of the scene: e.g a pose with arms and legs slightly apart for human motion.

3.2 Experimental Results

We have tested our algorithm on two challenging multi-view video sequences of non-rigid scenes.

The "Pants" dataset is composed of 8 cameras 480×640 pixels. It is courtesy of R. White, K. Crane and D.A. Forsyth [30]. We have successfully applied our algorithm to the first 60 frames of this dataset. Due to the high image resolution, four multi-resolution scales have been used to obtain the accurate spatio-temporal reconstruction shown in Figure 3. It has taken 24 hours to teconstruct the 60 frames spatio-temporal model.

Figure 4 demonstrates the superiority of our spatio-temporal approach compared to a frame-by-frame multi-view stereovision method [9], on the "Pants" dataset. The improvements are three-fold: (i) our approach exploits speed and acceleration to make better initial guesses of the subsequent time frames, thus being less prone to unwanted local minima (ii) thanks to the enforcement of temporal coherence, our approach is less likely to fail in regions with low photo-consistency evidence (iii) our approach simultaneously and consistently estimates 3D shape and 3D scene flow.

[1] `http://www.cgal.org/`

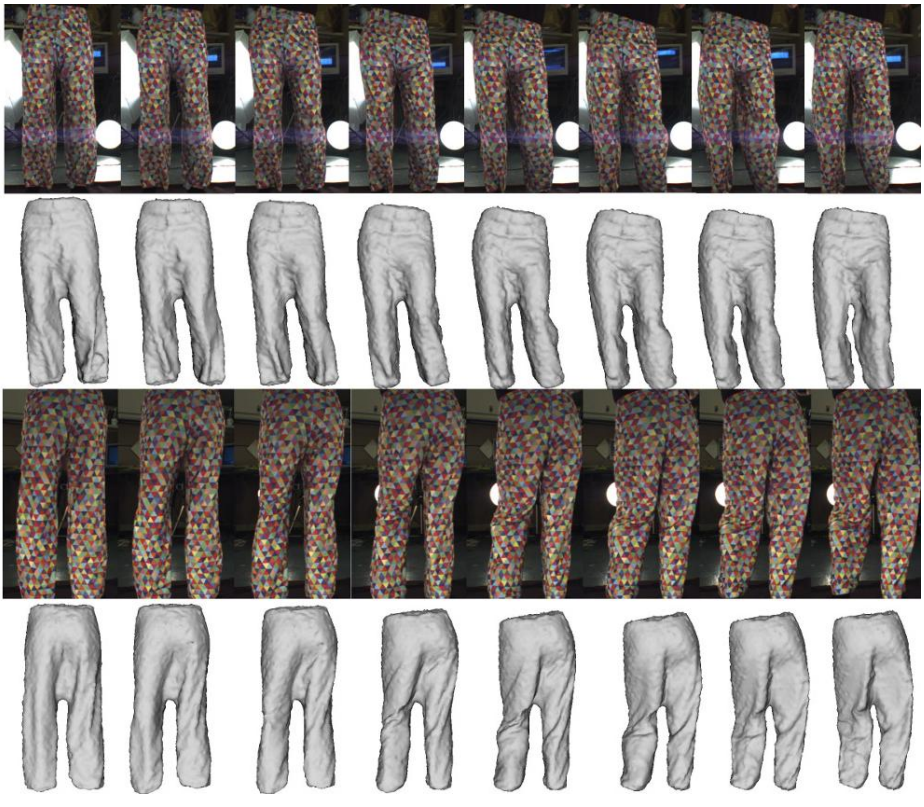


Fig. 3. Our results on the "Pants" dataset. See text for more details.

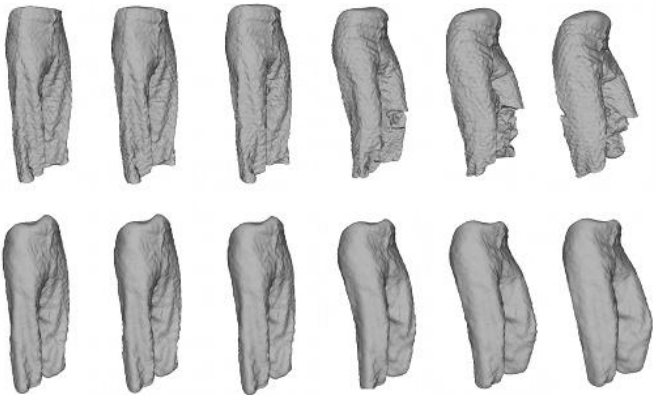


Fig. 4. Comparison between a frame-by-frame multi-view stereovision approach (top) and our spatio-temporal approach (bottom) on the "Pants" dataset. See text for details

Fig. 5. Our results on the "Dancer" dataset. See text for more details.

The "Dancer" dataset was made available to us by the 4Dviews company[2]. It was acquired by 14 calibrated and synchronized video cameras 1000×1000 pixels. We have applied our algorithm to the first 10 frames of this dataset. To bootstrap our multi-resolution and chronological optimization procedure, we have used a standard stereo-vision algorithm at the first time frame. The obtained reconstruction after processing three multi-resolution levels is displayed in Figure 5. We insist on the fact that we have not used silhouette information and that stereovision on such a dataset is quite challenging: because it was design for visual hull based techniques, many parts of the subject are textureless. It has taken 10 hours to teconstruct the full spatio-temporal model.

4 Conclusion

We have presented a novel variational approach to dense and accurate 3D shape and motion reconstruction from multi-view video sequences. Our method leverages the benefits of the animated mesh representation, of image-based photo-consistency, of discrete geometric optimization and of GPU computation. We have validated our algorithm on two challenging real datasets, and obtained results that rival state-of-the-art techniques.

[2] http://4dviews.com

References

1. Aganj, E., Pons, J.P., Ségonne, F., Keriven, R.: Spatio-temporal shape from silhouette using four-dimensional Delaunay meshing. In: IEEE International Conference on Computer Vision (2007)
2. Ahmed, N., de Aguiar, E., Theobalt, C., Magnor, M., Seidel, H.P.: Automatic generation of personalized human avatars from multi-view video. In: Proc. VRST 2005, pp. 257–260 (2005)
3. Ahmed, N., Theobalt, C., Rössl, C., Thrun, S., Seidel, H.P.: Dense correspondence finding for parametrization-free animation reconstruction from video. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
4. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: ACM SIGGRAPH (2008)
5. de Aguiar, E., Theobalt, C., Stoll, C., Seidel, H.P.: Marker-less deformable mesh tracking for human shape and motion capture. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
6. Furukawa, Y., Ponce, J.: Dense 3D motion capture from synchronized video streams. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
7. Goldlücke, B., Magnor, M.: Space-time isosurface evolution for temporally coherent 3D reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 350–355 (2004)
8. Neumann, J., Aloimonos, Y.: Spatio-temporal stereo using multi-resolution subdivision surfaces. The International Journal of Computer Vision 47(1-3), 181–193 (2002)
9. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. The International Journal of Computer Vision 72(2), 179–193 (2007)
10. Starck, J., Hilton, A.: Surface capture for performance based animation. IEEE Computer Graphics and Applications 27(3), 21–31 (2007)
11. Theobalt, C., Ahmed, N., Lensch, H., Magnor, M., Seidel, H.P.: Seeing people in different light-joint shape, motion, and reflectance capture. IEEE Transactions on Visualization and Computer Graphics 13(4), 663–674 (2007)
12. Varanasi, K., Zaharescu, A., Boyer, E., Horaud, R.P.: Temporal surface tracking using mesh evolution. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 30–43. Springer, Heidelberg (2008)
13. Vedula, S., Baker, S., Kanade, T.: Image-based spatio-temporal modeling and view interpolation of dynamic events. ACM Transactions on Graphics 24(2), 240–261 (2005)
14. Vedula, S., Baker, S., Seitz, S., Kanade, T.: Shape and motion carving in 6D. In: IEEE Conference on Computer Vision and Pattern Recognition (2000)
15. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. ACM Transactions on Graphics 27(3) (2008)
16. Vedula, S., Baker, S.: Three-dimensional scene flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 475–480 (2005)
17. Zhang, Y., Kambhamettu, C.: Integrated 3D scene flow and structure recovery from multiview image sequences. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 674–681 (2000)
18. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: IEEE International Conference on Computer Vision (2007)

19. Pons, J.P., Keriven, R., Faugeras, O., Hermosillo, G.: Variational stereovision and 3D scene flow estimation with statistical similarity measures. In: IEEE International conference on Computer Vision, vol. 2, pp. 597–602
20. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: European Conference on Computer Vision, pp. 739–751 (2008)
21. Zhang, Y., Kambhamettu, C.: On 3D scene flow and structure estimation. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 778–785 (2001)
22. Süßmuth, J., Winter, M., Greiner, G.: Reconstructing animated meshes from time-varying point clouds. Computer Graphics Forum (Proc. Eurographics Symposium on Geometry Processing) 27(5), 1469–1476 (2008)
23. Wand, M., Jenke, P., Huang, Q., Bokeloh, M., Guibas, L., Schilling, A.: Reconstruction of deforming geometry from time-varying point clouds. In: Eurographics Symposium on Geometry Processing, pp. 49–58 (2007)
24. Delaunoy, A., Prados, E., Gargallo, P., Pons, J.P., Sturm, P.: Minimizing the multi-view stereo reprojection error for triangular surface meshes. In: British Machine Vision Conference (2008)
25. Slabaugh, G.G., Unal, G.B.: Active polyhedron: Surface evolution theory applied to deformable meshes. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 84–91 (2005)
26. Vu, H.H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
27. Pons, J.P., Boissonnat, J.D.: Delaunay deformable models: Topology-adaptive meshes based on the restricted Delaunay triangulation. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
28. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: International Conference on Computer Graphics and Interactive Techniques, pp. 105–114 (1998)
29. Courchay, J., Pons, J.P., Keriven, R., Monasse, P.: Dense and accurate spatio-temporal multi-view stereovision. Technical Report 09-43, IMAGINE (July 2009)
30. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. In: SIGGRAPH (2007)

Semi-supervised Feature Selection for Gender Classification

Jing Wu, William A.P. Smith, and Edwin R. Hancock

Department of Computer Science, University of York, York YO10 5DD, UK
{jwu,wsmith,erh}@cs.york.ac.uk

Abstract. We apply a semi-supervised learning method to perform gender determination. The aim is to select the most discriminating feature components from the eigen-feature representation of faces. By making use of the information provided by both labeled and unlabeled data, we successfully reduce the size of the labeled data set required for gender feature selection, and improve the classification accuracy. Instead of using 2D brightness images, we use 2.5D facial needle-maps which reveal more directly facial shape information. Principal geodesic analysis (PGA), which is a generalization of principal component analysis (PCA) from data residing in a Euclidean space to data residing on a manifold, is used to obtain the eigen-feature representation of the facial needle-maps. In our experiments, we achieve 90.50% classification accuracy when 50% of the data are labeled. This performance demonstrates the effectiveness of this method for gender classification using a small labeled set, and the feasibility of gender classification using the facial shape information.

1 Introduction

Gender plays a significant role in both our social interactions and our interactions with machines. The ability to classify a user according to gender has many practical uses including making Human-Computer Interaction more user-friendly, access control, collecting demographic data, and improving the performance of face identity recognition by using gender specific models. In the last two decades appearance-based gender classification has attracted considerable attention in the literature [1], [2], [3], [4], [5], [6], [7], [8], [9] and particularly good performance has been reported using PCA-based features [10], [11]. However, the extracted PCA features still contain information that is redundant or even irrelevant for gender determination, and this limits gender classification accuracy. As a result feature selection is an important issue for gender classification. To select the gender discriminating feature subset, Sun et al. [3] has applied genetic algorithms to the extracted PCA features, and reported a best gender classification accuracy of 95.3%. Buchala et al. [4] used linear discriminant analysis (LDA) to explore which were the most important gender discriminating PCA feature components, and reported a 86.43% gender classification accuracy. However, these methods learn the discriminating features in a supervised way, and therefore require a large set of labeled data. For instance, Sun et al. [3] used

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 23–33, 2010.

300 images for training and 50 images for testing, while Buchala et al. used a training set of 2670 images.

In this paper, we apply a semi-supervised learning method to select the optimal set of gender discriminating feature components. The raw feature-vectors are extracted from facial images using shape-from-shading, and represent the modes of shape variation over a field of surface normals extracted using principal geodesic analysis [12]. Principal geodesic analysis is a generalization of PCA from data residing in Euclidean space to data residing on a non-linear Riemannian manifold (a unit hypersphere in the case of surface normals). By making use of the information provided by both labeled and unlabeled data, we successfully reduce the size of the labeled data (i.e. the number of facial images labelled as male or female) required. The semi-supervised learning method is based on a weighted graph representation of the data and employs harmonic functions over the graph to locate the optimal feature set. Each face is first represented by its PGA eigen-features, and is denoted by a vertex in a fully connected weighted graph. The edge weights are determined by a similarity measure for the corresponding pair of feature-vectors. The similarity measure weights each component of the PGA feature vector according to its significance for gender discrimination. By making use of harmonic functions and the entropy minimization strategy described by Zhu et. al in [14], we are able to learn the gender significance for each component of the PGA feature vectors. Experimental results demonstrate that using our method, the learned gender discriminating feature components are consistent with human perception. Moreover, we achieve 90.50% gender classification accuracy when 50% of the data are labeled.

A second noteworthy contribution of this paper is that we make use of fields of facial surface normals (facial needle-maps) instead of 2D brightness images for gender classification. There are two advantages of this approach. First, the 2.5D facial needle-maps reveal directly facial shape information. It has been shown by psychologists that gender classification is more effective using 3D shape than using 2D brightness [15]. Moreover, 3D facial shape provides more reliable information for surveillance purposes. The second advantage is that facial needle-maps can be recovered from single 2D images using the techniques such as shape-from-shading, and therefore avoid the expense of using a 3D sensor.

The outline of the paper is as follows. Section 2 reviews the statistical treatment of data residing on a Riemannian manifold using principal geodesic analysis. Section 3 commences by reviewing the harmonic functions and the semi-supervised learning method developed in [14], and then describes how to apply this method to gender discriminating feature selection. Experiments are presented in Section 4. Finally, Section 5 concludes the paper.

2 PGA on Facial Needle-Maps

A surface normal n may be considered as a point residing on a spherical manifold $n \in S^2$. Facial needle-maps, which are fields of N surface normals, may be considered as a point on the manifold $S^2(N) = \prod_{i=1}^{N} S^2$. Principal geodesic

analysis makes use of exponential and log maps, and intrinsic means to analyze data on this manifold.

If $u \in T_nS^2$ is a vector on the tangent plane to S^2 at n and $u \neq 0$, the exponential map, denoted $\exp_n(u)$, of u is the point on S^2 along the geodesic in the direction of u at distance $\| u \|$ from n. This is illustrated in Fig. 1. The log map, denoted $\log_n$ is the inverse of the exponential map. In the $S^2(N)$ space, the exponential and log maps are simply the products of N copies of the maps for S^2 given above.

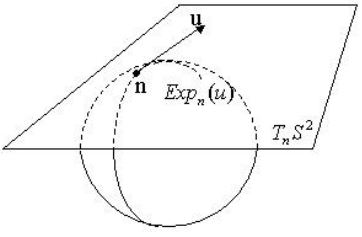


Fig. 1. The exponential map

The intrinsic mean is defined as $\mu = \arg\min_{n \in S^2} \sum_{i=1}^{N} d(n, n_i)$, where $d(n, n_i)$ is the geodesic distance between n and n_i on the manifold. For a spherical manifold, $d(n, n_i) = \arccos(n \cdot n_i)$. The intrinsic mean of data residing on a spherical manifold can be iteratively computed using the gradient descent method of Pennec [16]. Accordingly, the estimate $\mu^{(t)}$ at iteration t is updated as follows:

$$\mu^{(t+1)} = \exp_{\mu^{(t)}}(\frac{1}{N}\sum_{i=1}^{N}\log_{\mu^{(t)}}(n_i)).$$

In PGA each principal axis is a geodesic curve. In the spherical case this corresponds to a great circle. To project a point $n_1 \in S^2$ onto a geodesic G passing through the intrinsic mean, the projection π_G may be approximated linearly in the tangent plane $T_\mu S^2$: $\log_\mu(\pi_G(n_1)) \approx \sum_{i=1}^{m} V^i \cdot \log_\mu(n_1)$, where $V_1, \ldots V_m$ is an orthonormal basis for $T_\mu S^2$, which can be obtained using principal component analysis. Then, the principal geodesics for the S^2 space are obtained under the exponential map $\exp_\mu(v_i), i = 1...m$. This approximation enables the principal geodesics be computed by applying PCA in the tangent plane $T_\mu S^2$.

To apply PGA to facial needle-maps, we first make use of the log map to obtain the long vector representation of the faces in the tangent plane passing through the intrinsic mean. Then, we use the numerically efficient snap-shot method of Sirovich [17] to compute the eigenvectors and the according eigenvalues of the covariance of the long vectors. The leading m eigenvectors form the projection matrix $\Phi = (e_1|e_2|\ldots|e_m)$. Given a facial needle-map, we first obtain its long vector representation $u = [u_1, \ldots, u_N]^T$ in the tangent plane, and then we represent the face using its PGA feature vectors $b = \Phi^T u$.

3 Learning Gender Discriminating Features

After principal geodesic analysis, each face is represented by its m dimensional PGA feature vector. However, it has been well studied in [4] that not all of the PCA components are relevant to gender classification. The irrelevant or redundant information limits classification accuracy. We therefore select the most effective gender discriminating feature components from the PGA feature vectors. We make use of the learning strategy based on harmonic functions proposed in [14], and apply the method to gender classification.

3.1 Semi-supervised Learning Using Harmonic Functions

In [14], Zhu et. al represent the data $x_1, \ldots, x_l, x_{l+1}, \ldots, x_{l_u}$ as vertices in a connected weighted graph $G = (V, E)$. Here the first l data are labeled and the subsequent u data are unlabeled. The weight of each edge measures the similarity between the associated pair of data, and is calculated as,

$$w_{ij} = \exp\left[-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}\right], \tag{1}$$

where x_{id} is the dth component of the vector x_i, and σ_d is the length-scale of the dth component.

Each of the l labeled data have binary labels $y_i \in \{0, 1\}, i = 1, \ldots, l$. To assign labels to the unlabeled data, a real-valued function $f : V \rightarrow R$ is computed on the graph so as to satisfy two constraints. First, the function takes on the label as a value, and for the labeled data is $f(i) = y_i, i = 1, \ldots, l$. The second is that f minimizes the quadratic energy function

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij}(f(i) - f(j))^2.$$

It has been shown that the function f satisfying the above two constraints is harmonic, which means that the value of f at each unlabeled data point is the average of f over the neighboring vertices, i.e.

$$f(j) = \frac{1}{d_j} \sum_{i \sim j} w_{ij} f(i), j = l + 1, \ldots, l_u. \tag{2}$$

Equation (2) can be expressed as $f = Pf$, where $P = D^{-1}W$, $D = diag(d_i)$ is the diagonal matrix with the degree $d_i = \sum_j w_{ij}$ of each node i along the leading diagonal. Since the first l data are labeled and the subsequent u data are unlabeled, the weight matrix W (D and P similarly) can be split into 4 blocks after l rows and columns,

$$W = \begin{pmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{pmatrix}$$

The solution for f over the unlabeled data is

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l = (I - P_{uu})^{-1} P_{ul} f_l. \tag{3}$$

Hence from Equation (1) f_u is a function of the scale parameters σ_d. In [14], the scale parameters σ_ds are learned by minimizing the average label entropy,

$$H(f) = \frac{1}{u} \sum_{i=l+1}^{l+u} H_i(f(i)) \tag{4}$$

where $H_i(f(i)) = -f(i) \log f(i) - (1 - f(i)) \log(1 - f(i))$. Small values of the entropy indicate that the function values f for the unlabeled data are close to either 0 or 1, which means the labeling is relatively unequivocal. The authors also discussed the effectiveness of using entropy minimization together with the labeled data, and suggested to replace the transition matrix P in equation (3) with

$$\tilde{P} = \varepsilon U + (1 - \varepsilon) P, \tag{5}$$

where $U_{ij} = 1/(l + u)$, to avoid the complication that H has a minimum at 0 as $\sigma_d \to 0$. Gradient descent is used to learn the σ_ds. The computation is outlined as follows,

$$\frac{\partial H}{\partial \sigma_d} = \frac{1}{u} \sum_{i=l+1}^{l+u} \log\left(\frac{1 - f(i)}{f(i)}\right) \frac{\partial f(i)}{\partial \sigma_d}, \tag{6}$$

where $\partial f(i)/\partial \sigma_d$ is the ith component of $\partial f_u/\partial \sigma_d$, which is,

$$\frac{\partial f_u}{\partial \sigma_d} = (I - \tilde{P}_{uu})^{-1} \left(\frac{\partial \tilde{P}_{uu}}{\partial \sigma_d} f_u + \frac{\partial \tilde{P}_{ul}}{\partial \sigma_d} f_l\right), \tag{7}$$

where $\partial \tilde{P}_{uu}/\partial \sigma_d$ and $\partial \tilde{P}_{ul}/\partial \sigma_d$ are sub-matrices of $\partial \tilde{P}/\partial \sigma_d$, and,

$$\frac{\partial p_{ij}}{\partial \sigma_d} = \frac{\frac{\partial w_{ij}}{\partial \sigma_d} - p_{ij} \sum_{n=1}^{l+u} \frac{\partial w_{in}}{\partial \sigma_d}}{\sum_{n=1}^{l+u} w_{in}}. \tag{8}$$

Finally,

$$\frac{\partial w_{ij}}{\partial \sigma_d} = 2 w_{ij} (x_{id} - x_{jd})^2 / \sigma_d^3. \tag{9}$$

3.2 Application to Gender Classification

The aims in applying the above semi-supervised learning technique is as follows. By adjusting the parameters σ_d of the weight function according to equation (1), the influence of the gender discriminating feature components is increased, while that of the non-discriminating ones is decreased. The smaller value of σ_d, the greater the influence of component d in determining the similarity measure. As a result value of σ_d provides a means of gauging the significance of the different components of the PGA feature vector for gender classification.

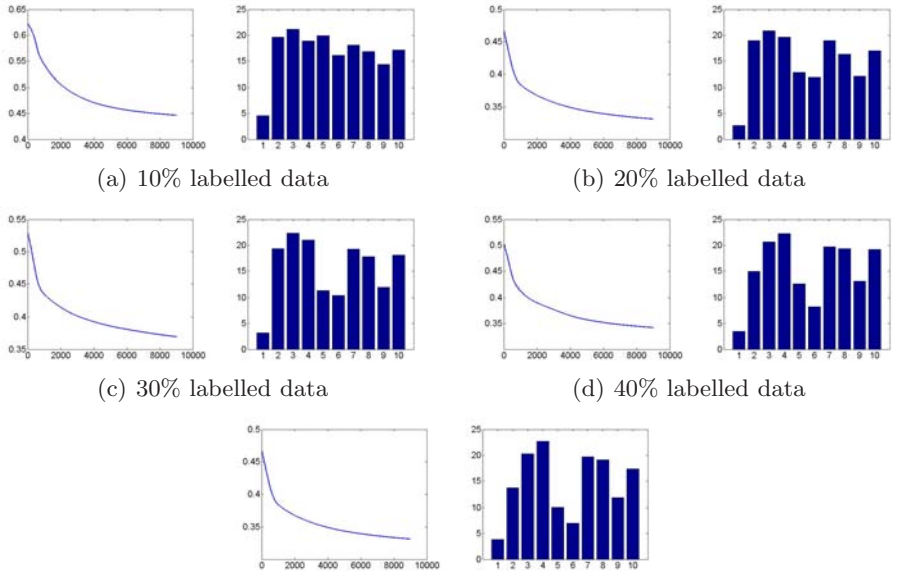
We set $f = 0$ for the labeled female data, and $f = 1$ for the labeled male data, and then apply the above semi-supervised learning method to learn the σ values for each component of the PGA feature vector. This strategy differs in two important ways from the feature selection methods described elsewhere in the literature ([3], [4]). First, the graph representation of the data and the harmonic functions enable us to make use of the entire set of available data, including both the labeled and unlabeled samples. As a result the size of required labeled dataset is reduced. Secondly, the σ values not only indicate which of the components of the PGA feature vector are discriminating feature components, but they also quantify the gender discriminating significance of each component. Incorporating this quantized significance into classification improves the gender classification performance.

After learning the σ values, the harmonic function values on the unlabeled data can be computed using Equation (3). From these values of $f(u)$, the genders of the unlabeled faces can be determined. When $f_u(i) < \frac{1}{2}$, the face i is assigned to be female, otherwise it is male.

4 Experimental Results

In this section, we evaluate the performance of gender discriminating feature selection using the semi-supervised learning method, and in particular we examine its performance for gender classification. The data used in our experiments are from the Max-Planck Face Database [18], [19]. The database comprises 200 laser scanned (Cyberware TM) heads without hair. There are 100 females and 100 males. The facial needle-maps are obtained by applying the following processing steps to the laser scans: a) the face is first orthographically projecting onto a frontal view plane, b) cropping the plane to 142-by-124 pixels so as to retain only the inner part of the face, and c) computing the surface normal at each pixel position. Principal geodesic analysis is applied to the 200 facial needle-maps. Each facial needle-map is then represented by its PGA feature vector.

We first apply the semi-supervised learning method to examine the gender discriminating feature components from the leading 10 PGA feature vectors. We examine the performance with 5 different fractions of labeled data (10%, 20%, 30%, 40%, 50%). The results for each fraction of labelled data are estimated with 10-fold cross validation. For learning the weights, we set the step size $\eta = 1.0$ for gradient descent, and set the smoothness parameter $\varepsilon = 0.01$ in Equation (5). We allow the learning process to run for 9000 iterations. The plot of the average label entropy (Equation (4)) is shown in the left panels of Figure 2. Irrespective of the the fraction of labeled data, the entropy decreases with the iteration number and converges after about 5000 iterations. The learned σ values for the leading 10 PGA eigenmodes are shown in the right panels of Figure 2. From this plot, when using 10% of the data as labeled, the σ value of the first component is significantly smaller than those of the remaining components. However, with an increasing fraction of labelled data, the σ values of the 5th and 6th components decrease rapidly, followed by those for the 2nd and 9th components. This indicates the



(a) 10% labelled data (b) 20% labelled data

(c) 30% labelled data (d) 40% labelled data

(e) 50% labelled data

Fig. 2. The performance of using semi-supervised learning for gender feature selection

importance of these components for gender discrimination. When 50% of the data is labeled, the 1st, 5th, 6th, followed by the 2nd and 9th PGA feature components are most gender discriminating. This is consistent with the results reported in [20].

To explore the data in more detail, we visualize the 1st, 5th, and 6th eigenmodes by showing the mean face together with its deviation along the 1st, 5th and 6th eigenmodes. The visualization is shown in Figure 3, and by inspection it seems plausible the three eigenmodes do convey some gender information. Turning our attention to the 1st eigenmode, the faces from left to right become more solid in appearance becoming larger and "squarer", while the cheeks become thinner. These are all masculine characteristics. In the case of the 5th component, the faces become more oval and the eyes wider. These are feminine characteristics. In the case of the 6th component, the faces again have more masculine appearance from left to right. Figure 3 therefore indicates that the gender discriminating features selected using the semi-supervised learning method are at least to some degree consistent with human perception.

After the determination of the parameters of the weight function, i.e. the σ values, we can use the values of f from Equation (3) to determine the gender for each unlabeled face. Again, the performance is examined with 5 different fractions of labeled data (10%, 20%, 30%, 40% 50%), and the average classification error rates for each fraction of labelled data are estimated with 10-fold cross validation. The classification results are shown in Table 1. From the table, it is clear

Fig. 3. Visualization of the 1st, 5th and 6th eigenmodes. From top to bottom are the 1st, 5th, 6th eigenmodes. The columns are according to the deviation from the mean, from left to right are λ=-30, λ=-20, λ=0 (the mean face), λ=20, and λ=30.

that the gender classification accuracy improves with the increasing fraction of labelled data. However, when 20% of the data are labeled, we can achieve over 81% gender classification accuracy. When 50% of the data are labeled, the classification accuracy reaches 90.50%, which is higher than the accuracy reported in [4] (86.43%), while with a much smaller volume of labeled data. These results demonstrate the effectiveness of using the semi-supervised learning method for gender classification, and the feasibility of gender classification using the facial shape information revealed by 2.5D facial needle-maps.

We also examine the gender classification performance using our method on facial needle-maps recovered from 2D face images using shape-from-shading (SFS). In our experiments, there are 140 2D images (70 females and 70 males), which are from the AR Face Database [21], with neutral expressions and no glasses. We use the principal geodesic SFS method proposed in [12] for the facial shape recovery. The statistical model required in this SFS method is constructed using the above 200 ground-truth needle-maps which are from the Max-Planck Face Database. Some examples of the recovered facial shapes are shown in Figure 4. From the figure, we can see that the recovered needle-maps and the surfaces give realistic shape, overcoming the well-known local convexity-concavity instability problem in previous SFS methods. Moreover, gender information is conveyed in the recovered facial needle-maps. This guarantees the feasibility of gender classification based on the recovered facial needle-maps.

Table 1. Classification accuracy using different fraction of labelled data

	10% labelled data	20% labelled data	30% labelled data	40% labelled data	50% labelled data
Accuracy	76.53%	81.17%	83.84%	86.15%	90.50%

(a) A female example (b) A male example

Fig. 4. Two examples of the recovered facial shapes. From left to right are the input images, the recovered needle-maps, the recovered surfaces.

Table 2. Classification accuracy for recovered facial needle-maps

	10% labelled data	20% labelled data	30% labelled data	40% labelled data	50% labelled data
Accuracy	74.44%	80.36%	87.35%	88.86%	89.52%

After the facial shape recovery, we first apply PGA to represent the recovered needle-maps using PGA feature vectors. Then, we apply the semi-supervised learning method to learn the values of σ_ds for the leading 10 PGA eigenmodes, and use the values of f to determine the gender. The classification performance is estimated with 5-fold cross validation for each fraction of labelled data (10%, 20%, 30%, 40%, 50%), and is shown in Table 2. From the table, we can see when only 30% of the data are labeled, we can achieve over 87% gender classification accuracy. When 50% of the data are labeled, we achieve the accuracy 89.52%. These results further confirm the effectiveness of using the semi-supervised learning method for gender classification, and demonstrate the feasibility of gender classification using the facial needle-maps recovered from 2D images using SFS.

5 Conclusions

In this paper we perform gender determination using PGA to parameterize 2.5D facial needle-maps and using a semi-supervised learning method [14] for the purposes of classification. The learning method is based on the graph representation of the data and harmonic label functions, and can be used to determine the most gender discriminating components of the PGA feature vectors. There are two novel contributions. First, we make use of the facial shape information conveyed by the facial needle-maps for gender classification. Second, by making use of the semi-supervised learning method, we are able to learn the gender discriminating features using a relatively small sample of labeled data and without sacrificing the classification accuracy. Experimental results demonstrate that the learned gender discriminating feature components are consistent with human perception. When 50% of data are labelled, the gender classification accuracy reaches

90.50% for ground-truth needle-maps, and 89.52% for needle-maps recovered using SFS.

There are a number of ways in which the graph representation can be enhanced for facial analysis problems including gender and ethnicity determination. Our immediate plans are to explore how to apply diffusion maps and graph-spectral probabilistic relaxation labeling to learn the gender discriminating features, and to determine the genders from facial images.

References

1. Golomb, B., Lawrence, D., Sejnowski, T.: SexNet: A Neural Network Identifies Sex from Human Faces. In: Advances in Neural Information Processing Systems, pp. 572–577 (1991)
2. Cottrell, G.W., Metcalfe, J.: Face, Emotion, and Gender Recognition Using Holons. In: Advances in Neural Information Processing Systems, vol. 3, pp. 564–571 (1991)
3. Sun, Z., Bebis, G., Yuan, X., Louis, S.J.: Genetic Feature Subset Selection for Gender Classification: A Comparison Study. In: WACV 2002, pp. 165–170 (2002)
4. Buchala, S., Davey, N., Gale, T.M., Frank, R.J.: Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images. In: Proc. IEEE ICMI 2005 (2005)
5. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
6. Jain, A., Huang, J.: Integrating Independent Components and Linear Discriminant Analysis for Gender Classification. In: FGR 2004, pp. 159–163 (2004)
7. Gutta, S., Weschler, H., Phillips, P.J.: Gender and Ethnic Classification of Human Faces using Hybrid Classifiers. In: Proc. of IEEE International Conf. on Automatic Face and Gesture Recognition, pp. 194–199 (1998)
8. Moghaddam, B., Yang, M.H.: Learning gender with support faces. IEEE Transaction Pattern Analysis and Machine Intelligence 24(5), 707–711 (2002)
9. Baluja, S., Rowley, H., Google Inc.: Boosting Sex Identification Performance. IJCV 1(71), 111–119 (2007)
10. Abdi, H., Valentin, D., Edelman, B., O'Toole, A.: More about the difference between men and women: evidence from linear neural networks and the principal component approach. Perception 24, 539–562 (1995)
11. O'Toole, A., Adbi, H., Deffenbacher, K., Valentin, D.: A low dimensional representation of faces in the higher dimensions of space. Journal of the Optical Society of America 10, 405–411 (1993)
12. Smith, W.A.P., Hancock, E.R.: Facial Shape-from-shading and Recognition using Principal Geodesic Analysis and Robust Statistics. International Journal of Computer Vision 76(1), 71–91 (2008)
13. Devijver, P., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, Englewood Cliffs (1982)
14. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In: ICML 2003 (2003)
15. O'Toole, A.J., Vetter, T., Troje, N.F., Bulthoff, H.H.: Sex Classification is Better with Three-Dimensional Structure than with Image Intensity Information. Perception 26, 75–84 (1997)

16. Pennec, X.: Probabilities and statistics on riemannian manifolds: A geometric approach. Technical Report RR-5093, INRIA (2004)
17. Sirovich, L.: Turbulence and the dynamics of coherent structures. Quart. Applied Mathematics XLV(3), 561–590 (1987)
18. Troje, N., Bulthoff, H.H.: Face recognition under varying poses: The role of texture and shape. Vision Research 36, 1761–1771 (1996)
19. Blanz, V., Vetter, T.: A Morphable Model for the Synthesis of 3D Faces. In: SIGGRAPH 1999 Conference Proceedings, pp. 187–194 (1999)
20. Wu, J., Smith, W.A.P., Hancock, E.R.: Learning Mixture Models for Gender Classification Based on Facial Surface Normals. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4477, pp. 39–46. Springer, Heidelberg (2007)
21. Martinez, A.M., Benavente, R.: The AR Face Database. CVC Technical Report, 24 (June 1998)

Planar Scene Modeling from Quasiconvex Subproblems

Visesh Chari[1,2], Anil Nelakanti[1,3], Chetan Jakkoju[1], and C.V. Jawahar[1]

[1] Center for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India-500032
[2] INRIA Rhône Alpes, Grenoble, France
[3] MirriAd Limited, London, UK

Abstract. In this paper, we propose a convex optimization based approach for piecewise planar reconstruction. We show that the task of reconstructing a piecewise planar environment can be set in an L_∞ based Homographic framework that iteratively computes scene plane and camera pose parameters. Instead of image points, the algorithm optimizes over inter-image homographies. The resultant objective functions are minimized using Second Order Cone Programming algorithms. Apart from showing the convergence of the algorithm, we also empirically verify its robustness to error in initialization through various experiments on synthetic and real data. We intend this algorithm to be in between initialization approaches like decomposition methods and iterative non-linear minimization methods like Bundle Adjustment.

1 Introduction and Related Work

In this paper, we describe a convex optimization based approach for piecewise planar reconstruction by optimizing inter-image homographies. This work is motivated by both the recent success of convex optimization based methods in various geometric problems like triangulation, resectioning [1,2], and the available sophistication in robust estimation of homographies across views [2].

Convex optimization methods have achieved recent success in the estimation of various geometric quantities like homography, pose, 3D point cloud (triangulation) [1,2] etc., and are even shown to be reasonably robust to noise [2]. There are even works on outlier estimation and removal using convex optimization [3] . On the other hand, there also has been progress on robust estimation of homographies from multiple views of a scene plane [2]. However, even though homographies can also be expressed as a function of the camera pose, and can be decomposed using SVD in a similar manner to fundamental matrices [4,5], piecewise planar reconstruction as a 3D reconstruction pipeline has not received much attention.

To this extent, we intend to develop an algorithm that can be a useful "bridge" between SVD based initialization methods mentioned above and non-linear optimization methods like Bundle Adjustment (BA). We focus on the iterative reconstruction process, that alternates between optimizing a six parameter camera pose vector for each view, and a four element plane parameter vector for each scene plane, by optimizing over the resulting inter-image homographies.

We make the following contributions in this work. First, we introduce objective functions for producing optimal estimates of pose and plane parameters, along the lines of

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 34–43, 2010.

[2]. Then, we show how a Branch and Bound (BnB) algorithm may be formulated for the computation of optimal rotation between views [4].

Some of the recently proposed frameworks on L_∞ based quasi-convex cost functions problems form the motivation for our work [1,6], while some closely related works include projective Bundle Adjustment (pBA) [7] and BA with constraints [8]. However, we differ from these works in the kinds of objective functions minimized (quasiconvex as opposed to non-linear) and in the quantities we optimize (homographies as opposed to 3D points). Recent study of bi-linear problems also has relevance to our work [9] since plane and pose parameters are combined together in a bi-linear form in the expansion of a homography (Equation 1). However, the formulation proposed in [9] requires that the entire set of plane and pose parameters need to be optimized together. Also, estimation of rotation parameters becomes infeasible in such a scenario. Thus we do not resort to a formulation along the lines of [9].

The rest of this paper is organized in the following manner. Section 2 sets the problem of pose estimation in a homographic framework and motivates the need for the use of optimization. Section 3 presents our solution and algorithm details. Experimental analysis on synthetic and real-world sequences are done in Section 4 and finally, we conclude with a discussion on future directions and applications in Sections 5.

2 SVD Based Initializations

Let there be m planes in the world, characterized by the parameters $\left[n^1, d^1, \ldots, n^m, d^m\right]$. The jth plane is characterized by the parameters (n^j, d^j), where n^j represents the normal of the plane and d^j represents the perpendicular distance from world origin. Let there be two cameras with external parameters $[\mathbf{I} \mid \mathbf{0}]$ and $[\mathbf{R} \mid \mathbf{t}]$. For simplicity, let us assume that the internal parameters of the cameras are set to identity ($\mathbf{K} = \mathbf{I}$). Thus the homography induced by the jth plane between the two views [10] is given by

$$\mathrm{H^j} = \left[\mathbf{R} \; - \; \frac{\mathbf{t} {n^j}^T}{d^j} \right] \tag{1}$$

Decomposition algorithms for obtaining camera pose and plane normals from homography matrix using Equation 1 are well known [11,5]. However, since, the process of pose computation from correspondences through the homography matrix involves two SVDs, a theoretical sensitivity analysis of such algorithms is difficult and approximate [12]. Thus it is more advantageous to do an empirical study of the error in the estimation of plane and pose parameters, given noise in image correspondences.

Figures(1a-1c), depict the poor performance of one of the SVD based decomposition algorithms [5]. The experiments consisted of adding increasing amounts of noise to a previously determined set of normalized image correspondences. Homographies obtained after a standard RANSAC routine were then decomposed to obtain estimates of the plane and pose parameters. Variances are plotted against error in pixel coordinates, with a maximum variance of 5 pixels which corresponds to approximately 1% of the image size. As can be seen, translation and normal estimations are adversely affected by image noise. The errors for the other algorithm [11], were similar.

The variances in Figures(1a) plot the error in estimation of rotation parameters when noise is introduced into the system. As is seen, the maximum variation of rotation parameters in the Euler angle space is 6 degrees, for as high as one percent image noise. Comparison with the translation and normal errors, which are as high as 40 degrees in the polar space Figures(1b-1c), show that the decomposition algorithm produces much more robust estimates of rotation than either translation or normal parameters. This explains the greater need for better estimates of translation and normal parameters compared to that of rotation parameters that are much close to the actual values.

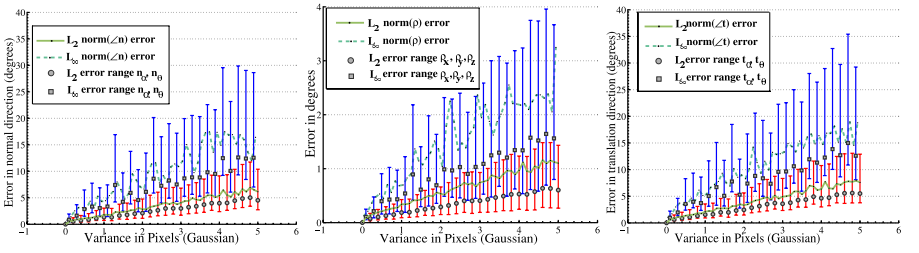


Fig. 1. (a,b,c) Plot the $L2$ and L_∞ errors in the rotation angles, translation direction and normal direction respectively. Also are plotted the maximum error ranges for these quantities. The translation and normal direction errors are computed as Euclidean distances in polar space.

3 Optimization Framework

In this section, we describe our algorithm. First, we start with the simple case when rotation is assumed known, and the rest of the parameters are optimized (Section 3.1). The reason for this is the non-convexity of the orthonormality constraints of the rotation matrix. Since algorithms for estimating the rotation already exist [4], and since we have shown rotation parameters to be robustly recovered from SVD decompositions as compared to other parameters (Figure 1a), we treat rotation separately (Section 3.3). Finally, in order to bring all the SVD decomposition estimates into a single coordinate system, we describe a convex function in Section 3.2.

3.1 Formulation of the Objective Function

We wish to find plane and pose parameters that best fits Equation 1 which is non-linear in terms of quantities $(\mathbf{R}, \mathbf{t}, n^j, d^j)$ that need to be computed. However, observe that when either the plane or the pose parameters are known, Equation 1 is linear in the remaining unknowns. This simple fact is used to define an objective function that measures the geometric distance between the homography computed from plane/pose parameters and the homography estimated from point correspondences. If the homography matrix with varying pose parameters and fixed plane parameters is defined as $\mathcal{H}rt^j = \left[\mathbf{R} - \frac{\mathbf{t}{n_c^j}^T}{d_c^j}\right]$ for the jth plane then the corresponding objective function is

$$\mathcal{F}_{(\mathbf{R},\mathbf{t})} = \sum_{i=1}^{8} \frac{H_i^j}{H_9^j} - \frac{\mathcal{H}rt_i^j}{\mathcal{H}rt_9^j} \tag{2}$$

Similarly, when the plane parameters are allowed to vary fixing pose parameters the homography function is $\mathcal{H}nd^j = \left[\mathbf{d^j} R_c - t_c \mathbf{n^j}^\top\right]$ and the objective function

$$\mathcal{F}_{(\mathbf{n},\mathbf{d})} = \sum_{i=1}^{8} \frac{H_i^j}{H_9^j} - \frac{\mathcal{H}nd_i^j}{\mathcal{H}nd_9^j} \tag{3}$$

(R_c, t_c, n_c^j, d_c^j) are fixed and the optimization runs over free variables denoted by bold letters. There are two important observations to make at this point. Firstly, equations (2, 3) are both linear fractional: both the numerator and denominator are affine in terms of the unknowns. Secondly, it is possible to optimize all parameters by alternatively minimizing Equation 2 and Equation 3 till convergence.

The proposed algorithm is a two step process. An initial estimate of the parameters is acquired using SVD-based decomposition in the first. However, estimates from SVD decomposition in the first step do not all have the same scale factor. Such estimates need to be threaded together and brought down to a common universal scale before carrying out the optimization. This is done by minimizing the difference between various estimates of a single quantity as described in Section 3.2.

Subsequently, in the second step, this estimate is improved in an optimization framework. However, minimizing Equation 2 without enforcing the constraints inherent to a rotation matrix will not lead to a physically valid rotation matrix. Equation 2 fails to be a linear fractional with rotation constraints enforced complicating its minimization. Hence, rotation is handled separately as explained in Section 3.3 and Equation 2 is minimized by varying only the translation as in Step 7 of Algorithm 1.

The optimization takes advantage of the fact that the objective functions are quasiconvex and employs convex optimization techniques at minimizing them. Variables t^i and (n^j, d^j) are minimized in alternating iterations. Optimization of t^i takes into account information from all visible planes. Similarly, optimization for (n^j, d^j) is done with information from all views in which the plane is visible. This two step process ensures the quasiconvexity of the objective functions. The complete method is summarized in Algorithm 1.

Algorithm 1. Complete Algorithm Summarized.

1: Input: Homographies ${}^kH_j$ for $j = 1, \ldots, J$ and $k = 1, \ldots, K$ of plane Π_j between the camera views ${}^kP$ and reference view ${}^0P = [I|0]$.
2: SVD-based decomposition: Decompose ${}^kH_j$ to get ${}^kR_j$, $\frac{{}^kt_j}{{}^kd_j}$, ${}^kn_j$.
3: Initialization: ${}^kR =$ median$_j$ $\{{}^kR_j\}$ and $t =$ median$_j\{{}^kt_j\}$.
4: Set to universal scale: Assume each actual camera translation to be a unit vector in the direction of $\frac{{}^kt_j}{d_j}$, i.e., $\|{}^kt\| = 1$. Let ${}^kG_j = [{}^kR - \frac{{}^kt{n_j}^T}{{}^kd_j}]$ and ${}^kG_j^s = (g_1, g_2, \ldots, g_9)^T$.
5: Iterative Minimization:
6: $\Sigma_k \Sigma_j \left\{{}^kH_j^s - {}^kG_j^s\right\} \leq \delta$
7: Update $({}^kt)$: $({}^kt) = \arg\min_{{}^kt} \max_{j=1}^{J} \sqrt{\Sigma_i [\frac{{}^jh_i}{{}^jh_9} - \frac{{}^jg_i}{{}^jg_9}]^2} \forall k = 1, \ldots, K.$
8: Update (n_j, d_j): $(n_j, d_j) = \arg\min_{n_j, d_j} \max_{k=1}^{K} \sqrt{\Sigma_i [\frac{{}^kh_i}{{}^kh_9} - \frac{{}^kg_i}{{}^kg_9}]^2} \forall j = 1, \ldots, J.$

3.2 Universal Scale

Each decomposition by the algorithms of Faugeras [11] and Zhang [5] produces estimates of $\{R, t, n\}$ assuming d (perpendicular distance of plane from origin) to be unity. Thus estimates vary by a scale factor and need to be tied down to a single universal scale which in the presence of noise has to be computed using optimization.

Let the solutions of translation obtained by decomposing homography H_i^j be t_i^j . Ideally, the actual translation is $t_i = t_i^j d^j$. Since various estimates of the same quantity must be consistent, we find an $x = \left[t_1, t_2, \ldots, t_k, d^1, d^2, \ldots, d^m\right]^\top$ for which an error $|f(x)|_\infty$ is minimum. $f(x)$ is a vector with elements of the set $\{t_i - t_i^j d^j \mid i \in [1, k], j \in [1, m]\}$ stacked up. Optimal estimates are found by performing the minimization $x^* = \arg\min_x |f(x)|_\infty$.

The considered error function is convex [13], made from the pointwise maximum of the convex function $(t_i - t_i^j d^j)$. An unconstrained optimization in this case could lead to the trivial solution of all zeros for x which is undesirable. To avoid this we *fix* perpendicular distance of anyone of the planes (say, d^1) to unity. This also sets the overall scale of the minimization process.

3.3 Retrieving Rotation

Constraints inherent to rotations and normals like orthonormality constraints of the rotation matrix are non-convex and do fit into a convex framework. Such constraints have been handled in the literature [4,14] using under estimators and over estimators of the non-convex function with a Branch and Bound algorithm. We, thus, handle rotation separately rather than in the above optimization. We use image coordinates of planes available on the lines of [4] to solve for rotation R_i of the i^{th} view. The objective function to be minimized is

$$\mathcal{F}_{(\mathbf{R}_i, \mathbf{t}_i)} \equiv \mathbf{Find}(R_i, \mathbf{t}_i) \qquad \text{s.t. } \angle(H_i^j \mathbf{x}_1^j, (\mathbf{R}_i - \mathbf{t}_i \frac{{n^j}^T}{d^j})\mathbf{x}_1^j) < \epsilon_{min} \tag{4}$$

which can be alternatively posed as

$$\mathcal{F}_{(\mathbf{R}_i, \mathbf{t}_i)} \equiv \mathbf{Find}(\mathbf{R}_i, \mathbf{t}_i) \qquad \text{s.t. } \angle(H_i^j \mathbf{x}_1^j, \mathbf{R}_i(\mathbf{I} - \mathbf{t}_i \frac{{n^j}^T}{d^j})\mathbf{x}_1^j) < \epsilon_{min} \tag{5}$$

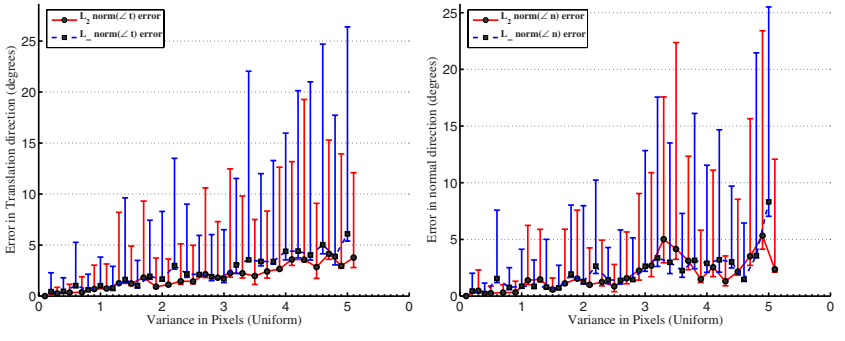
where x_1^j are points from the jth plane in the first view. Arguments of bounds and in general the branching strategy of [4] can now be incorporated into the current framework. The analysis that estimates of rotation from SVD-based methods are more robust than that of translations and normals as noted in Section 2 practically helps the idea of handling rotation separately at a later stage. Figure 3c shows the performance of the objective function described above in the presence of varying noise. The L_2 norm in angular space (roll-pitch-yaw) is plotted against increasing amounts of noise in image pixels.

4 Experimental Analysis

In order to test the proposed algorithm, we have conducted experiments using SeDuMi [15] on both synthetic and real-world data. Synthetic data is obtained by generating points on planes and projecting them onto camera matrices. Real world data sets tested include the Oxford Model House, Corridor, and UNC datasets. In all these cases, the real world is assumed to be segmented into planes apriori *i.e.* interest points and hence correspondences computed are assumed to be clustered into planes. However, there are automatic algorithms to achieve such a classification [16].

4.1 Synthetic Data

Generation. Random points are generated on the XY-plane which is then re-positioned at a random location. Two random camera matrices are generated and the world points of many such planes are projected using them to generate image points. Gaussian noise of varying standard deviation is added to these image points to create synthetic correspondence data. Homographies are then computed using the RANSAC after normalization [10] which can alternatively be generated by [1]. The generated Homographies are decomposed using Faugeras' and Zhang's algorithms [11,5] to generate data for both initialization and comparison. Algorithm 1 is then run with this data, to produce our estimate and is compared with the SVD-based algorithms and Bundle Adjustment in the 6-parameter pose space by plotting the euclidean distance between estimated and ground truth values.



(a) Translation (b) Normal

Fig. 2. Plot of L_2 and L_∞ norms of the distance in pose space between estimated and ground truth quantities from Algorithm 1 against increase in variance of Gaussian error in point correspondences. Comparison with the two SVD based methods is shown.

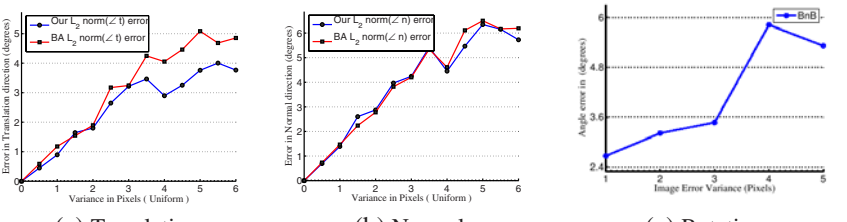
Effect of noise. Figures (2a,2b) show the effect of increasing image noise on the accuracy of estimation. Two observations can be made for both translations and normals. First, the average error in the estimation of both parameters is less than 5 degrees even for a 1% error in the image coordinates, which is a considerable amount of error. This justifies the robustness of our algorithm to image noise. The second observation is that

the mean errors (averaged for 100 trials) in all these cases are located close to the minimum errors represented by the lower end of the error bar. We can conclude that most of the estimations center around the mean, with only a few deviating towards the higher end. Another interesting observation is that even the resilience to noise is apparent till about 3 pixel error after which the maximum error in both cases seems to increase. This can be attributed to the fact that after a point the algorithm possibly settles into a local minima because of the inaccurate initialization. However, this is still better than the results of SVD-based methods in Figures 1b, 1c.

Comparison with Bundle Adjustment. We empirically compare our algorithm with standard iterative non-linear optimization technique of Bundle Adjustment (BA) [17], which uses Levenberg-Marquardt internally. BA is initialized by the output of the SVD-based approaches similar to ours. This initialization is used to minimize the following error over the normals and the translations

$$(R, t, n_j, d_j) = \arg \min_{{}^kR, {}^kt, n_j, d_j} \sum_k \sum_j \sum_i [\frac{h_i}{h_9} - \frac{x^T A_i x}{\overline{x}^T A_9 \overline{x}}]^2 \tag{6}$$

where, $x = ({}^1R^s, \ldots, {}^KR^s, {}^1t^T, \ldots, {}^Kt^T, n_1^T, \ldots, n_J^T, d_1, \ldots, d_J)$ and A_i is a matrix s.t. $x^T A_i x = g_i$ and $\overline{x}$ is x with the initial SVD estimates of ${}^kR, {}^kt, n_j, d_j$ substituted. The improvement in translations is shown in Fig (3a) and that of normals in Fig (3b). They are shown for varying levels of variance each of which has been tested for 100 trials. They clearly show our algorithm performing better than BA.



(a) Translation (b) Normal (c) Rotation

Fig. 3. (a-b)Plot of L_2 norm of the distance in pose space between estimated and ground truth quantities from Algorithm 1 and Bundle adjustment against increase in variance of Gaussian error in point correspondences.(c) Error in recovery of rotation parameters using the objective function of Section 3.3

Effect of planes and views. Figures (4a,4c,4b,4d) show the effect of the number of planes and views on the performance of the algorithm. Contrary to intuition, increasing the number of planes does not seem to have much effect on the accuracy of the estimates of translation parameters. On the other hand, increasing the number of views increases the parameter size, and the accuracy of translation estimates dwindles since the number of planes and hence, measurements is kept constant. In the case of normals, however, increasing the number of views results in a marked improvement in the accuracy of their estimates.

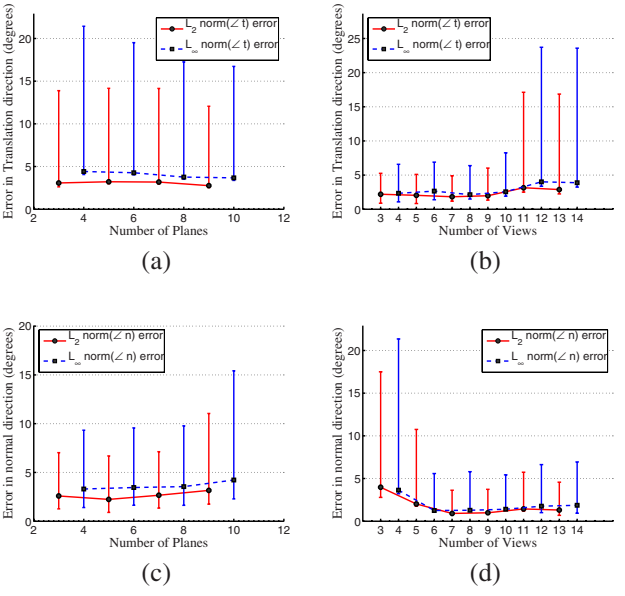


Fig. 4. The above figures plot the effect of planes and views on the accuracy in estimation of the translation and normal parameters. First two figures plot the effect on translations and last two plot the effect on normals. For the experiment with increasing planes, the number of views was kept constant at 10, and that for views, the number of planes was set to be 3.

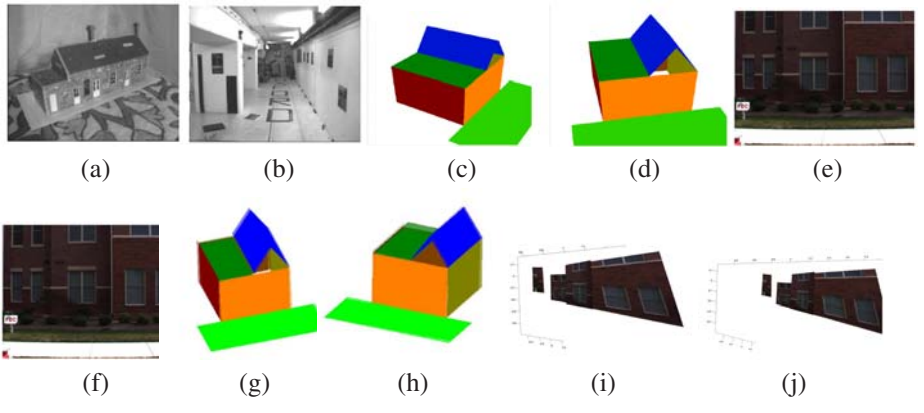


Fig. 5. Sample images of scenes reconstructed using our approach. (House(a), Corridor(b), synthetic(c-d), UNC((e-f))). (g-h) illustrates the accuracy of our reconstruction, the ground truth and reconstructed models are overlapping. (i-j) Texture mapped UNC reconstructions.

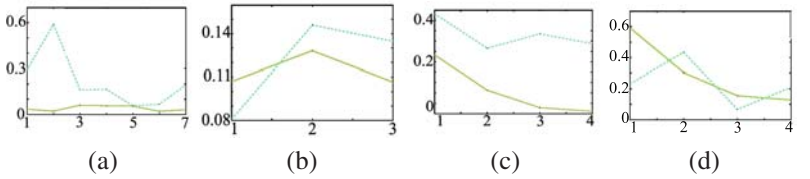


Fig. 6. Plots of the L_∞ error between plane and pose parameters with respect to the ground truth, for the House and Corridor sequence. $L2$ error shows similar plots. Y-axis of plots (a),(b),(c) and (d) is the angular error in radians, X-axis of (a) and (c) is the number of views, where as X-axis of (b) and (d) is the number of planes. In the plots (a),(b),(c) and (d), dotted curve represents the Faugeras initialization and other curve represents our approach.

4.2 Real Data

In order to test on data from the real-world, we chose two Oxford data sets and the UNC data set. The House, and Corridor data sets (Figures (5a,5b)) are accompanied by correspondences and estimates of the camera matrices, while the UNC data set only comprises camera matrices.

Figures 6a-6b show the comparison between our estimation and that of the decomposition of Faugeras for the Oxford data sets. The L_2 and L_∞ errors between the estimated and ground truth quantities are plotted. In order to compare normals, we took the best estimate of normals from the available decompositions. As can be seen from the plots, estimates of translation from our algorithm are far better than the corresponding algorithm by Faugeras. We found that Zhang's algorithm produces estimates similar to that of Faugeras' algorithm in most cases. The same situation is repeated in the Corridor sequence (Figures 6c-6d), where translation is very accurately obtained. An explanation of why certain plane parameters are "perturbed" by a higher error is that some of the homographies are erroneous and the error in a particularly bad homography is distributed across planes. Finally, the UNC data set (Figures 5i,5j) show the visual accuracy of our reconstruction.

5 Discussion and Conclusion

We proposed a framework that reconstructs piecewise planar scenes in much the same way as Bundle Adjustment for point sets. The algorithm incorporates both multiple planes and views and does not constrain all the planes to be visible in any single view. This makes it a useful bridge between initialization approaches and non-linear minimization methods

The existing framework is not without its drawbacks. Currently, though the objective functions show robustness to noise, it does not work very well in the presence of outliers. Existing literature in convex optimization that handles outliers may be used for this purpose [3]. Similarly, uncertainty of correspondences can also be handled with techniques like [18]. Secondly, constraints *between* planes like orthogonality may help in stabilizing the overall reconstruction [8]. One other issue related to this algorithm is its practical applicability. Recent results reported in [6,19] are very relevant to our

work and may be used to improve the run time of our algorithm, making it suitable for faster computation required by videos. We believe that our current contribution lays down a useful framework for practically viable optimization over planes, and wish to investigate further into its use for large scale optimization.

References

1. Kahl, F., Henrion, D.: Globally optimal estimates for geometric reconstruction problems. In: ICCV (2005)
2. Kahl, F.: Multiple view geometry and the l-infinity norm. In: ICCV (2005)
3. Sim, K., Hartley, R.: Removing outliers using the l-infinity norm. In: CVPR (1), pp. 485–494 (2006)
4. Hartley, R., Kahl, F.: Global optimization through searching rotation space and optimal estimation of the essential matrix. In: ICCV (2007)
5. Zhang, Z., Hanson, A.R.: 3d reconstruction based on homography mapping. In: ARPA (1996)
6. Kahl, F., Agarwal, S., Chandraker, M.K., Kriegman, D.J., Belongie, S.: Practical global optimization for multiview geometry. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 592–605. Springer, Heidelberg (2006)
7. Mitra, K., Chellappa., R.: A scalable projective bundle adjustment algorithm using the 1 norm. In: ICVGIP (2008)
8. Bartoli, A., Sturm, P.: Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. In: IJCV (2003)
9. Chandraker, M., Kreigman, D.: Convex optimization for bilinear problems in computer vision. In: CVPR (2008)
10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
11. Faugeras, O., Lustman, F.: Motion and structure from motion in a piecewise planar environment. In: IJPRAI (1988)
12. Criminisi, A., Reid, I.D., Zisserman, A.: A plane measuring device. In: Image Vision Comput., pp. 625–634 (1999)
13. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
14. Olsson, C., Kahl, F., Oskarsson, M.: Optimal estimation of perspective camera pose. In: ICPR (2006)
15. Sturm, J.F.: Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones (1999)
16. Bartoli, A.: A random sampling strategy for piecewise planar scene segmentation. In: CVIU, vol. 105(1), pp. 42–59 (2007)
17. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Workshop on Vision Algorithms, pp. 298–372 (1999)
18. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: ICCV (2005)
19. Agarwal, S., Snavely, N., Seitz, S.: Fast algorithms for l-inf problems in multiview geometry. In: CVPR (2008)

Fast Depth Map Compression and Meshing with Compressed Tritree

Michel Sarkis, Waqar Zia, and Klaus Diepold*

Institute for Data Processing, Technische Universität München
Arcisstr. 21, 80290 Munich, Germany
sarkis@mytum.de, waqar.zia@tum.de, kldi@tum.de

Abstract. We propose in this paper a new method based on binary space partitions to simultaneously mesh and compress a depth map. The method divides the map adaptively into a mesh that has the form of a binary triangular tree (tritree). The nodes of the mesh are the sparse non-uniform samples of the depth map and are able to interpolate the other pixels with minimal error. We apply differential coding after that to represent the sparse disparities at the mesh nodes. We then use entropy coding to compress the encoded disparities. We finally benefit from the binary tree and compress the mesh via binary tree coding to condense its representation. The results we obtained on various depth images show that the proposed scheme leads to lower depth error rate at higher compression ratios when compared to standard compression techniques like JPEG 2000. Moreover, using our method, a depth map is represented with a compressed adaptive mesh that can be directly applied to render the 3D scene.

1 Introduction

Determining the depth of a scene depicted by a stereo image is a well established research area in computer vision. The depth is usually represented by a disparity or a depth map that reflects the movement of the pixels between the two images. In order to compute this map, it is necessary to first evaluate some matching costs among the pixels of the images. Then, an energy function is defined over these costs and optimized. A disparity map can be used in a variety of applications: 3D scene reconstruction, image based rendering and 3D-television (3D-TV). In order to use this entity for such purposes, the depth map needs to be as accurate possible so that the visualization errors are minimal. This is why stereo matching is still an active area of research. For the interested reader, an excellent survey about this topic is found in [1].

Nowadays, it is possible to find numerous stereo matching algorithms that are able to result in high quality depth maps. Irrespective of the algorithm used, a disparity map has to be compressed at a later stage to save the storage requirement or to limit the needed bandwidth if it has to be transmitted over a network, e.g. telepresence and 3D-TV. A typical way to compress a depth map is by applying standard image or video compression techniques like JPEG 2000 or MPEG4 ASP. Such schemes process a depth image

* This research is sponsored by the German Research Foundation (DFG) as a part of the SFB 453 project, High-Fidelity Telepresence and Teleaction.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 44–55, 2010.

while taking only the visual quality into consideration. This is why they result in a high amount of error in the reconstructed depth values especially at high compression rates, see [2,3,4] for more details.

Motivated by this limitation, we propose in this work an algorithm for depth map compression and meshing based binary space partitions. Using this concept, we first divide the disparity image into a triangular tree (Tritree). This tree has a binary format and is actually the content adaptive mesh approximation of the depth map. The nodes of the mesh are the non-uniform samples of the map and can reconstruct the other depth values with a minimum error. We then apply suitable entropy coding on the samples and the binary tree in order to further compress the data. The results we obtain on several disparity images show that our scheme leads to a noticeable improvement in the quality when compared to other techniques even at high compression ratio. Our scheme is fast and can be applied in real-time to simultaneously mesh and compress a depth map.

The method we propose is based on binary space partitions (BSP) to subdivide an image. This concept was described in detail in our previous work [5]. There, we apply three variants of BSP to approximate normal images with a mesh taking into consideration the visual quality of the results. In this paper, there are two main contributions that make our work original. Firstly, we adopt the fastest BSP variant and tailor it specifically for depth images. To do that, we take the depth error rate of the compressed depth image into account when building the mesh and not its visual quality, i.e. we drop out Peak Signal to Noise Ratio (PSNR). Secondly, we post-process the mesh with several lossless coding schemes to achieve a very efficient representation of the disparity map.

The rest of this paper is organized as follows. We present in Section 2 a brief review on depth map compression techniques. We derive the proposed method for depth map meshing and compression in Section 3. We evaluate the proposed scheme and compare it to other methods in Section 4. In the end, we draw some conclusions in Section 5.

2 Related Work

The depth map is an image where the intensity values represent the displacement of the corresponding pixels between the stereo images. The simplest way to compress this entity is by applying a state of the art image compression method like JPEG, JPEG 2000 or MPEG4 ASP. While compressing an image, these techniques take into account the visual quality of the result and not the amount of errors in the intensity values of the compressed image. This is usually represented with the term Mean Squared Error (MSE), equivalently the PSNR, between the compressed image and its uncompressed version. The MSE actually can be low even if all the reconstructed pixels are erroneous. For example, a compressed disparity image where each pixel is reconstructed with an absolute difference of 2 has a MSE of 4 while the pixel error rate is 100%. This explains why JPEG and MPEG4 ASP result in blocking artifacts while JPEG 2000 blurs the edges or the depth discontinuities if a high compression ratio is desired [2,3,4]. Taking the MSE as a metric is not recommended for the compression of a depth image since the latter is a piecewise smooth surface, i.e. it has discontinuities along the edges while it is smooth otherwise [1]. Applying it might not be harmful when visualizing the depth image but leads to a lot of artifacts and errors in 3D reconstruction and rendering.

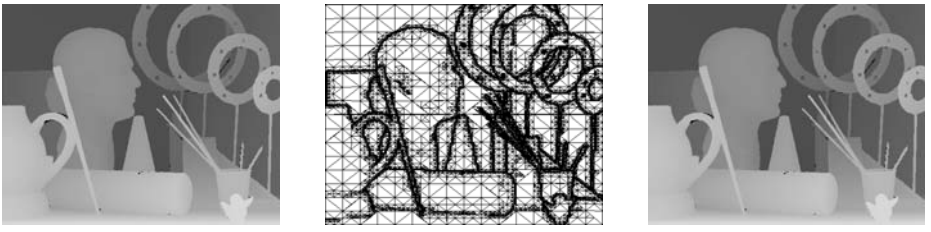
The limitations of the standard image compression techniques motivated the research to develop more sophisticated schemes for depth maps. In [2], JPEG 2000 was modified to accommodate for region of interest coding and reshaping the dynamic range of the depth map. In [3,6], a disparity image was meshed using height fields meshing schemes. Then, the mesh was coded at different resolutions to obtain the compressed representation of the disparity map. In [7], a depth map was hierarchically decomposed into four regions depending on the locations of the edges. These regions were then merged and fed to an H264/AVC encoder. In [8], the context of a depth map was classified and then compressed depending on its class via predictive image coding schemes.

The proposed method is based on adaptive or irregular mesh generation. It is thus important to state some recent development in this domain since such methods can be adapted to mesh and compress a depth map. Irregular meshes are usually generated based on coarse to fine strategies, see [9,10] for some examples. One important algorithm is the quadtree since it has been applied in various image types. Quadtree was applied to adaptively mesh and code videos in [11]. It was also applied to approximate and visualize terrains in [12]. More recently, quadtree was used in [4] to compress and mesh disparity images. Another way to obtain irregular meshes is by applying the concept of non-uniform sampling or content adaptive meshing of images. Content adaptive meshing is the art of approximating an image with an adaptive mesh. The nodes of the mesh are called the non-uniform samples of the image. These samples are able to interpolate all the other pixels of the image via the mesh up to a predefined error. Some of the techniques developed in this direction are [5,13,14,11].

3 Proposed Scheme

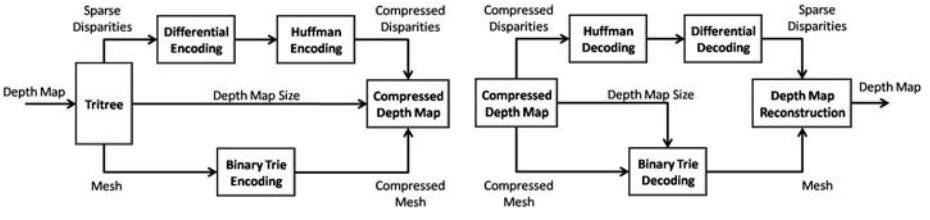
In this work, we will be applying the concept of content adaptive meshing to generate an irregular mesh. The advantage of such a scheme is its ability to preserve the depth discontinuities. The sampling rate will be high near the edges or any kind of depth discontinuities, hence, the size of a triangle in such regions will be small. Otherwise, the sampling rate will be low and the size of the corresponding triangle in this case will be big. An example is shown in Fig. 1. Applying a mesh as a first step to compress a depth map has been also used in [3]. There, however, the generated mesh does not deal with depth discontinuities but on small details within the objects. This is why the algorithm was later improved in [6] to handle these issues by detecting the discontinuities, modeling them and using a constrained triangulation in such regions. With the proposed content adaptive meshing strategy, the depth discontinuities will be already taken into account for the sampling rate will be very high at these locations. This makes edge modeling or enforcing some constraints on the triangulation not required anymore.

Therefore, the main idea of our proposed scheme is to first approximate a depth map with an adaptive mesh by detecting its non-uniform samples. This mesh will be able to approximate the original content of the depth map with a minimal error. To further reduce the size, the obtained mesh and the corresponding disparity values of the nodes will be encoded in a lossless fashion to obtain an efficient representation of the map. The block diagram of our compression/decompression scheme is depicted in Fig. 2 and will be explained in the remainder of this section.



(a) (b) (c)

Fig. 1. (a) The original Art disparity map of [15]. (b) The corresponding adaptive mesh using the proposed tritree in Section 3.1. (c) The recovered disparity map from the mesh.



(a) (b)

Fig. 2. (a) The tritree based meshing/compression scheme of a depth map. (b) The decompression process and depth map reconstruction.

3.1 Content Adaptive Meshing with Tritree

The pixels of a disparity map form a 3D space represented by the 2D-coordinates of the pixel in the map and the corresponding disparity value. Each triangle T of the desired mesh is formed by three vertices. Let $\mathbf{v}_i(x_i, y_i, d_i)$ with $i = 1, 2, 3$ be the three vertices of T. The plane Π described by T is defined using the normal equation

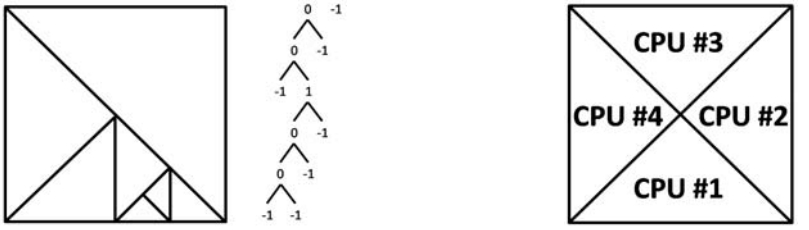
$$\overrightarrow{\eta} \cdot \mathbf{p}_n + k = 0, \tag{1}$$

where $\mathbf{p}_n\,(x_n, y_n, d_n)$ denotes a pixel with coordinates (x_n, y_n) and depth value d_n lying on the plane, k is a real constant such that $k = -\overrightarrow{\eta} \cdot \mathbf{v}_i$ and $\overrightarrow{\eta}$ is the normal vector to the plane. The vector $\overrightarrow{\eta} = (\eta_1, \eta_2, \eta_3)$ can be computed as the cross product of any two edges of the triangle. To recover the disparity value $\hat{d}_n$ of a pixel lying inside T, we should write Equation (1) in the form

$$\hat{d}_n = -\left(\eta_1 x_n + \eta_2 y_n + k\right)/\eta_3. \tag{2}$$

To determine the content adaptive mesh of a disparity map, it is necessary that we find all the triangles of the map whose nodes can reconstruct the depth values of the pixels lying within them using Equation (2). To determine the quality of the reconstructed values, we will be using the percentage of the disparity errors PERR inside each triangle as a measure. PERR is defined in a triangle T of the mesh to be

$$\mathrm{PERR} = \frac{1}{\lfloor \triangle T \rfloor} \sum_{i=0}^{\lfloor \triangle T \rfloor - 1} f\left(d\left(x_i, y_i\right), \hat{d}\left(x_i, y_i\right)\right)\%, \tag{3}$$



(a) (b)

Fig. 3. (a) The tritree subdivision principle applied to an example depth image along with the corresponding binary tree. An upper or right triangle is encoded with 1 if divided, a lower or left triangle is encoded with 0 if divided. A -1 indicates the end of the branch, i.e. no further divisions of the triangle. (b): The concept of tritree subdivisions working in parallel on several CPUs. The depth image is pre-divided into 4 parts and each is processed by a different CPU.

where $\triangle$ is a symbol denoting the area of a triangle, i.e. $\triangle T$ is the area of T, $\lfloor \cdot \rfloor$ is the floor operator, (x_i, y_i) are the coordinates of a pixel in T, $\hat{d}$ is the disparity value in the reconstructed disparity map from the compressed image and d is the corresponding disparity in the uncompressed depth image. The function f in (3) returns the value 0 if the difference between d and $\hat{d}$ is strictly less than 1; otherwise, f returns the value 1.

To obtain the adaptive mesh, we should minimize PERR over each triangle of the mesh. Using the BSP concept, we first divide the disparity map along one of the diagonals into two triangles. We then check each of the triangles if it satisfies the predefined PERR threshold ϵ. If it does not, we recursively divide the triangle into two smaller triangles from the longest edge until ϵ is satisfied. We then repeat this step until no further subdivisions are possible. This method leads to a Triangular Tree which is why we called it *tritree*. The difference in applying tritree in this work as compared to [5] is the minimization of PERR instead of PSNR. PERR ensures that the depth error rate is minimized across each triangle in the mesh. In [5], however, the aim was to compress texture images. For that, PSNR was used since it is known to reflect the visual quality of a compressed image. In other words, applying PSNR to depth images does not necessarily reduce the depth error rate but only guarantees that a compressed depth map is visually close to its uncompressed version.

One advantage of tritree is the fact that the obtained adaptive mesh is nothing but a binary tree due to the incurred property form BSP. Let us assume that when dividing a triangle, an upper or right triangle is assigned the code 1 while a lower or left triangle is given the code 0. With tritree, we first divide an image from the diagonal into two triangles. If the PERR is not satisfied, a 1 is written in the code tree if the upper/left triangle is divided while a 0 is written in the code tree if a lower/right triangle is divided. If a triangle satisfies PERR, it is not divided and a -1 is written instead of 0 or 1 to indicate that the tree does not extend anymore at this node. An example tritree subdivision is shown in Fig. 3a along with the corresponding binary tree.

As a consequence, a content adaptive mesh with tritree has a corresponding binary tree that can be directly generated with no extra effort to represent the mesh. This will allow us to save the mesh as will be seen later in a very compact format. Another

advantage of tritree is that it can be easily parallelized to operate on multi-core processors as seen in Fig. 3b. This is because the processing of each child triangle is totally independent from the other triangles. Hence, we can pre-divide a depth map into several pieces and let each one operate on a different CPU. The result can be then padded at the end to obtain the overall mesh (binary tree).

Depth Map Reconstruction: Until now, we have represented a disparity map with a binary tree and the corresponding sparse disparity values of the mesh nodes. The inverse process to reconstruct back the depth map is also possible. Using the binary tree and the image size, we can easily reconstruct back the mesh interconnections in a hierarchical manner. We first create an image space of the size of the depth map. We then scan through the binary tree and add a triangle in the image when 0 or 1 is encountered in the code. From the mesh connectivity in the image, we can get back the coordinates of the mesh nodes. Using the mesh and the depth values at the nodes, we can recover back the overall depth map with Equation (2).

3.2 Residual Coding

In order to make the algorithm shown in Fig. 2 complete, we still need to make the depth map representation more compact. We have to code the residual data, i.e. binary tree (sometimes referred to as binary trie) and the sparse disparity values, to remove the remaining redundancies in the data. Therefore, the purpose of this residual coding stage is to further compress the remaining redundancy in the data, either via lossy or lossless coding. Lossy encoding is typically based on a constrained rate-distortion minimization. Since our approach already enables this tradeoff in the content adaptive meshing stage, we will apply lossless coding techniques (specifically, entropy coding) at this stage. This ensures that this step is fully reversible.

The most popular entropy coding techniques are the Huffman and the Arithmetic coding. Arithmetic coding can theoretically achieve the lower bound given by entropy $H(P)$, and defined by Shannon's source coding theorem as

$$H(P) = \sum_{k=1}^{n} -p\{e_k\} \log_2 p\{e_k\}, \tag{4}$$

where P is the probability distributor of the symbols and $p\{e_k\}$ is the probability of an event e_k. Obviously numerical inaccuracies will prevent achieving this limit. Huffman coding, which can be considered as a simplified case of Arithmetic coding [16], stays further away from this limit since its codes have an integral length. As reported in [16,17], both Huffman and Arithmetic coding have a complexity of $\mathcal{O}(M \log_2 M)$, where M is the dimension of the symbol set used to represent the data. Still, Arithmetic coding has been traditionally considered far more complex than Huffman coding for it uses complex operations like division. With modern hardware, however, this difference is not significant anymore [17].

Entropy coding is a variable length codeword scheme. We first have to represent the target data by symbols with a specific probability distribution. We then assign variable length codewords to the symbols according to their probabilities. The more-peak shaped

histogram the symbols have, the more compression can be achieved. Hence, histogram shaping of the data is an important cornerstone of this technique. In order to keep the complexity as low as possible, we will consider simple and reversible histogram shaping schemes like differential coding. For the statistical modeling of the data, we will apply a *semi-adaptive* scheme [18] since it has an intermediate complexity.

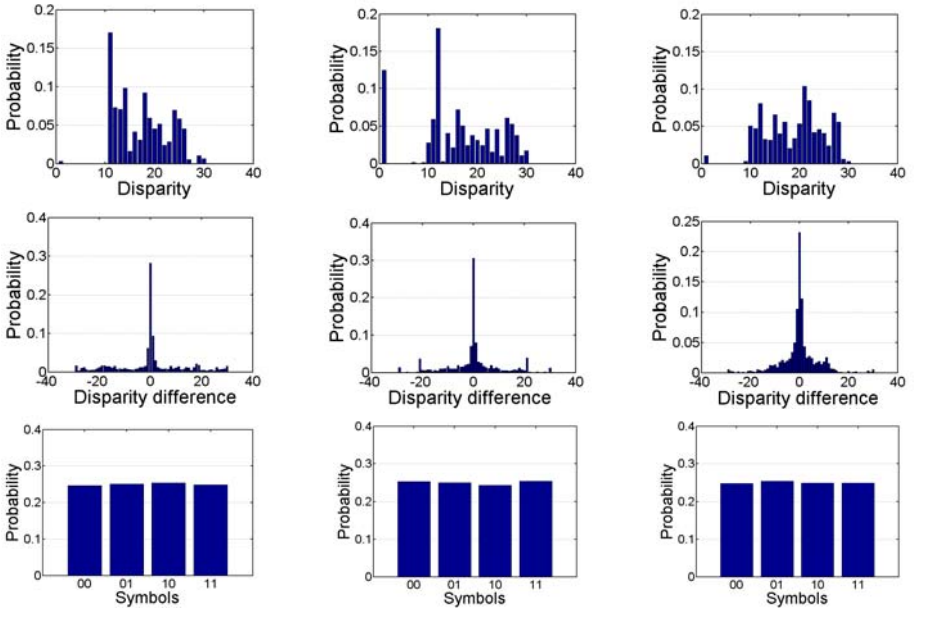
Sparse Disparity Coding: In a depth map, the co-located depth values are highly correlated since this entity is a piecewise smooth surface [1]. This is also true for sparse depth maps which makes differential coding handy to obtain a histogram with a peak shape. Differential coding is done in this work by predicting the next disparity from the immediately preceding one. Moreover, the sparse values are scanned from top to bottom in a raster order. By looking at the first two rows in Fig. 4, we can see that differentially coded disparities have peak shaped histograms and are much better suited for entropy coding after this step. Note that the results shown in this figure were obtained by taking tritree at 50% error threshold, hence the depth maps are very sparse. At a lower compression ratio, hence error rate, the sparse maps will be denser and will have even stronger correlation between the neighboring depth values.

Binary tree coding: We first code the binary tree as a *pre-ordered bitstream*, which as reported in [19,20] is a very compact representation. To do that, we express each node by a codeword that has only four possible binary symbols, namely 0 0, 0 1, 1 0, and 1 1, which correspond to the four possibilities in the binary tree: 0 1, -1 1, 0 -1 and -1 -1. The histogram of some sparse binary trees obtained with tritree is shown in the last row of Fig. 4. One can notice that the tree histogram is almost flat and no considerable compression gain can be expected if entropy coding is performed. This is actually expected since the shape of the binary tree (adaptive mesh) depends only on the characteristics of the depth map, i.e. some regions might have more discontinuities than others. This cannot be predicted in coding and leads to little redundancy that can be exploited. Thus, the pre-ordered bitstream is enough to represent the binary tree.

4 Results

We will perform some tests that consist of evaluating the performance of the proposed tritree based compression scheme. We will use as a test data set the ground truth depth maps of the Middlebury test bench [21,15] and the depth maps of the Microsoft Breakdancer and Ballet sequences which were computed using the stereo matching technique of [22]. We will compare our method with the JPEG and the JPEG 2000 image compression standards. We will be using three quality measures in the comparisons. The first one is the PERR of the compressed depth map, see (3). The second one is the mean squared error (MSE) of the compressed map and the third one is the rate distortion curve or the average number of bits used to represent each pixel or bits per pixel (BPP). We will also make these measurements while varying the compression ratio.

In Fig. 5a, we show the outcome of the algorithms on the Teddy and Art ground truth depth maps. For the proposed scheme, we show the results using the Huffman coding to compress the sparse disparities. We also show the outcome of the Arithmetic coding to



(a) Art (b) Cones (c) Dolls

Fig. 4. Histograms of the sparse disparity values in the upper row, the differentially coded disparity values in the middle row and of the binary trees in the lower row for various depth images

compress the disparities since it is closer to the theoretical bound with which they can be compressed. As we can see, the proposed scheme leads to the best performance when compared to the others. The PERR in the compressed disparity map is always lower. This is because JPEG and JPEG 2000 target the MSE as a measure in the compression. To visualize that, we also show the results of the MSE versus the compression ratio in Fig. 5b. We can see that the MSE in the compressed disparity maps with our scheme is higher. This might seem contradicting at the beginning but our scheme minimizes the PERR and not the MSE. So although the MSE is higher, it is not the case with PERR. Moreover, the amount of bad depth values better reflects the quality of a depth map since MSE can be low even if many depth values are erroneous as we previously said. Looking at the rate distortion curves, we can also notice that the performance of our scheme is much better. Using less the 1 BPP on average, we can now represent a depth map with less than 1% error rate. This was also the case when we tested our algorithm on the depth maps of the Breakdancer and Ballet sequences obtained with stereo algorithm of [22]. The outcome is depicted in Fig. 6. By comparing the Huffman coding to the Arithmetic coding, we can see that they lead to almost the same outcome. This justifies the employment of the Huffman scheme since it has less complexity.

In all the obtained results, the proposed tritree based compression scheme has shown a better performance since the optimization takes the PERR into account. In other words, it does not optimize to only maintain the visual quality of the depth maps with MSE as the others do. This allows us to obtain a higher compression ratio with an error

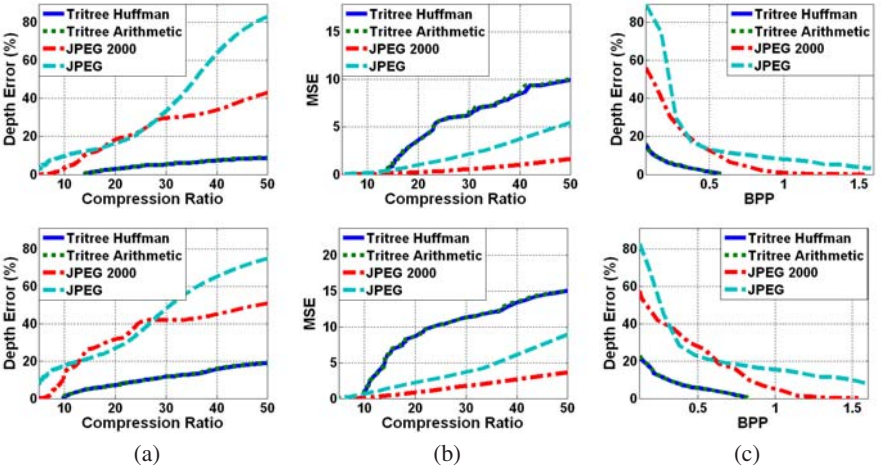


Fig. 5. Comparison of the compression algorithms on the *ground truth* Middlebury depth images of [21,15]. First row: Teddy. Second row: Art. (a): Depth error rate in % versus the compression ratio. (b): MSE versus the compression ratio. (c): The rate distortion curve.

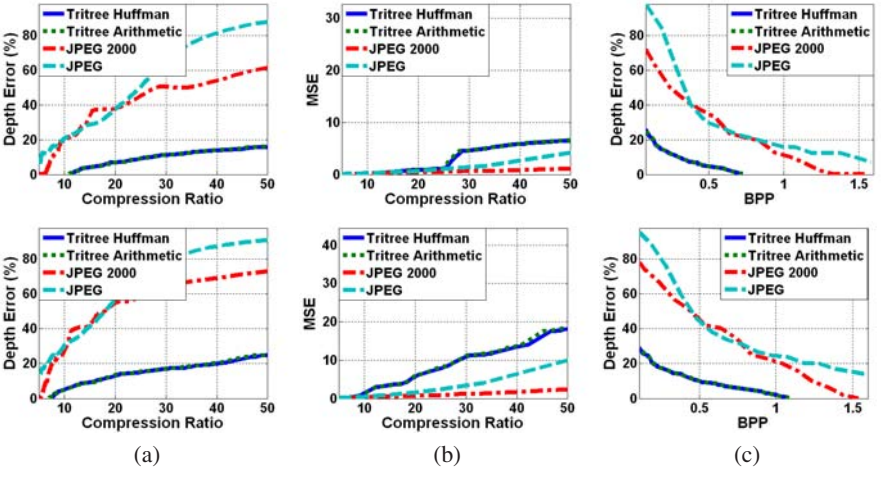
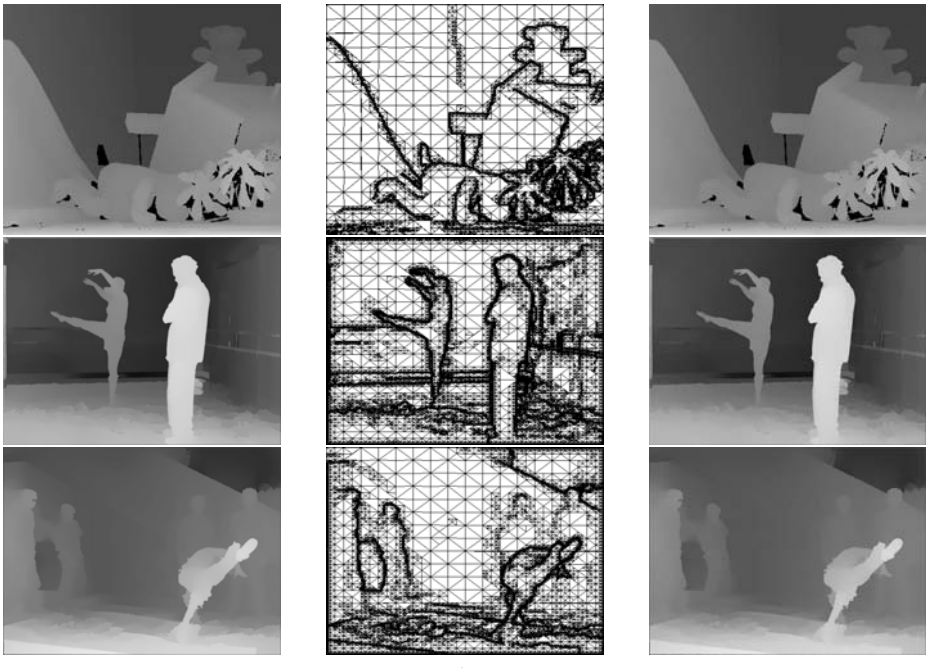


Fig. 6. Comparison of the compression algorithms on the depth images computed by the *stereo scheme* of [22]. First row: Breakdancer. Second row: Ballet. (a): Depth error in % versus the compression ratio. (b): MSE versus the compression ratio. (c): The rate distortion curve.

rate less than 1% of the total depth values of the map. In Fig. 7, we show the original Teddy depth map of [21] and the output of the stereo scheme of [22] on Ballet and Breakdancer. We also show the adaptive mesh at less than 1% PERR threshold obtained with tritree and the reconstructed depth maps. As one can see, the adaptive mesh preserves the content of a depth map by generating small triangles along the discontinuities and big triangles elsewhere. Thus, non-uniform sampling with tritree removes the

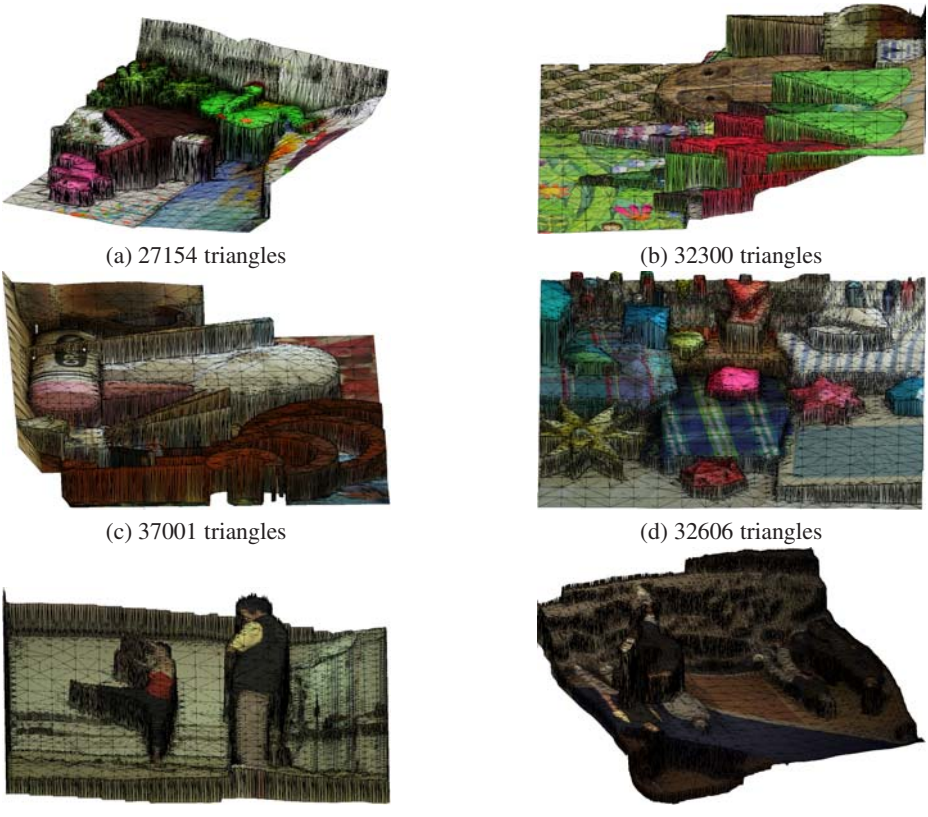


(a) (b) (c)

Fig. 7. Visual outcome of the scheme on the ground truth Teddy depth map of [21] and the resulting depth maps form the stereo scheme of [22] on Ballet and Breakdancer. From up to down: Teddy, Ballet and Breakdancer. (a): The original depth maps. (b): The adaptive meshes. (c): The reconstructed depth maps with our scheme. The depth error rate was set to 1%. The compression ratio is: 13.8, 7.3 and 11.1. The obtained error rate in % is: 0.43, 0.45 and 0.31.

major redundancies in the depth image and then the coding schemes remove the residuals ones. This leads to a more compact representation which can be seen from the obtained compression ratio at these values in Fig. 5 and Fig. 6. We also present in Fig. 8 rendered 3D views of some depth images overlayed with the adaptive meshes obtained with our scheme. This shows that our algorithm can be used not only to compress the depth maps but to create a mesh representation that can be applied in 3D rendering.

Concerning the timing, our method attains real-time operation using the efficient implementation described in Section 3.1. It requires around 80 ms with the Teddy and Cones images, 90 ms for the Dolls image, 88 ms for the Art and Moebius images. It takes 110 ms for the Ballet image at half the resolution and 390 ms at the full resolution while it requires 100 ms for the Breakdancer at half the resolution and 350 ms at the full resolution. These measurement are made on an AMD Opteron 64 bit quad core PC of 2.2 GHz speed. The algorithm is written using the C++ programming language.



(a) 27154 triangles (b) 32300 triangles

(c) 37001 triangles (d) 32606 triangles

(e) 52481 triangles (f) 35994 triangles

Fig. 8. Rendered 3D views overlayed with the corresponding meshes obtained using the proposed scheme. (a): Teddy, (b): Cones, (c): Art, (d): Moebius, (e): Ballet and (f): Breakdancer.

5 Conclusion

We derived in this paper a method to simultaneously mesh and compress a depth map. The technique is based on BSP. It generates a mesh in the form of a binary tree by locating the non-uniform samples of the depth map. These samples are the nodes of the mesh and are able to interpolate the other pixels with minimal error. To minimize the representation of the sparse pixels, we apply differential coding followed by entropy coding to compress the sparse disparities. We also code the binary tree as a pre-ordered bitstream. The compressed depth map is thus the combination of the compressed mesh and the compressed disparities. Our algorithm leads to lower depth error rate at higher compression ratios when compared to compression techniques like JPEG and JPEG 2000. We are now able to represent a depth map using less than 1 BPP on average while having less than 1% errors in the depth values. Our method attains real-time and the mesh can be easily applied to render the 3D scene represented by the depth map.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision 47(1), 7–42 (2002)
2. Krishnamurthy, R., Chai, B., Tao, H., Sethuraman, S.: Compression and transmission of depth maps for image-based rendering. In: IEEE Conf. Image Processing (October 2001)
3. Chai, B., Sethuraman, S., Sawhney, H., Hatrack, P.: Depth map compression for real-time view-based rendering. Pattern Recognition L 25(7), 755–766 (2004)
4. Morvan, Y., de With, H.N., Farin, D.: Depth-image representation employing meshes for intermediate-view rendering and coding. In: SPIE Conf. Electronic Imaging: Stereoscopic Displays and Applications (January 2006)
5. Sarkis, M., Diepold, K.: Content adaptive mesh representation of images using binary space partitions. IEEE T. Image Processing 18(5), 1069–1079 (2009)
6. Farin, D., Peerlings, R., de With, H.N.: Depth-image representation employing meshes for intermediate-view rendering and coding. In: 3DTV Conf. (May 2007)
7. Kim, S.Y., Ho, Y.S.: Mesh-based depth coding for 3D video using hierarchical decomposition of depth maps. In: IEEE Conf. Image Processing (October 2007)
8. Bao, P., Gourlay, D., Li, Y.: Context based depth image compression for distributed virtual environment. In: IEEE Conf. Cyberworlds (December 2003)
9. Sappa, A.D., Garcia, M.A.: Coarse-to-fine approximation of range images with bounded error adaptive triangular meshes. SPIE J. Electronic Imaging 16(2) (April 2007)
10. Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L., Faust, N., Turner, G.: Real-time, continuous level detail rendering of height fields. In: ACM SIGGRAPH, August 1996, pp. 109–118 (1996)
11. Wang, Y., Lee, O.: Use of two-dimensional deformable mesh structures for video coding, part II–the analysis problem and a region-based coder employing an active mesh representation. IEEE T. Circuits and Systems for Video Technology 6(6), 647–659 (1996)
12. Pajarola, R.: Overview of quadtree-based terrain triangulation and visualization. Technical Report UCI-ICS-02-01, Information and Computer Science, Uni. California Irvine (2002)
13. Demaret, L., Dyn, N., Iske, A.: Image compression by linear splines over adaptive triangulations. Signal Processing J. 86(7), 1604–1616 (2006)
14. Yang, Y., Wernick, M.N., Brankov, J.G.: A fast approach for accurate content-adaptive mesh generation. IEEE T. Image Processing 12(8), 866–880 (2003)
15. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: IEEE Conf. Computer Vision and Pattern Recognition (June 2007)
16. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press, Cambridge (2001)
17. Said, A.: Introduction to arithmetic coding theory and practice. Technical Report HPL-2004-76, Hewlett-Packard Laboratories (2004)
18. Said, A.: Comparative analysis of arithmetic coding computational complexity. In: Data Compression Conf. (March 2004)
19. De Jonge, W., Tanenbaum, A., Van De Riet, R.: Two access methods using compact binary trees. IEEE T. Software Engineering SE-13(7), 799–810 (1987)
20. Shishibori, M., Mochizuki, H., Arita, T., Aoe, J.I.: An efficient method of compressing binary tries. In: IEEE Conf. Systems, Man, and Cybernetics (October 1996)
21. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: IEEE Conf. Computer Vision and Pattern Recognition (June 2003)
22. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. ACM T. Graphics 23(3), 600–608 (2004)

A Three-Phase Approach to Photometric Calibration for Multi-projector Display Using LCD Projectors

Lei Zhang, Siyu Liang, Bo Qin, and Zhongding Jiang

Computer Graphics Laboratory, Software School, Fudan University, China
zdjiang@fudan.edu.cn

Abstract. Photometric calibration plays important role when building seamless appearance of multi-projector display. In this paper, we address photometric issues on chrominance variation and luminance nonuniformity in multi-display system constructed using LCD projectors. A three-phase approach is proposed to construct imaging models, which makes transformations among them when formulating the whole imaging procedure. These models are named as single-projector model, normalized-projector model and display-wall model. Single-projector model describes the imaging procedure from the projector's input color to its measured tristimulus values in CIEXYZ. Normalized-projector model denotes the common gamut of projectors, which normalizes each single-projector model, and makes every projector have the same ranges of chrominance and luminance. The display-wall model treats the whole display as one projector, which has similar photometric model to single LCD projector. Weighting light contributions from all projectors using the display wall model, our method can achieve visually plausible seamlessness.

1 Introduction

Multi-Projector displays become popular in scientific visualization, military simulation, CAD, multimedia exhibition, and entertainment fields [1]. For achieving visual seamlessness, geometric alignment and photometric calibration must be addressed when building such displays [1].

There are many methods [2] for geometry calibration, we focus on photometric calibration in this paper. Photometric calibration is very important for seamless and uniform display. Some work [3,4,5,6,7] have been done on this topic, but it still lacks of one practical method. About photometric problems, there are chrominance variation and luminance nonuniformity [6,7].

Stone introduced the color and brightness issues of multi-projector display, explained projector characteristics, and brought forward common standard gamut method [3,4]. Wallace et al. extended Stone's work, and presented one non-parametric full-gamut color matching algorithm [5].

Majumder and Steven gave a more detailed explanation about color nonuniformity issues [6]. They pointed out that chrominance and luminance problems

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 56–65, 2010.

exist in multi-projector display system. The luminance problem was more important, which includes inter-projector and intra-projector luminance variations. They proposed one Luminance Attenuation Map(LAM) method to equalize the luminance output across the display wall, which achieved better image quality than Alpha Blending [8]. Using human contrast sensitivity function, they further proposed one gradient domain smoothing method to smooth the input image [9]. These two methods mainly focused on the luminance variation across the multi-projector display, but the chrominance variation problem was overlooked [7]. In practice, luminance adjustment can generate good result only for projectors with the same model and brand [6,9]. In practical system, projectors with different bands and model are common.

Two-phase method[7] and color gamut mapping method[10] considered both chrominance and luminance problems. These methods employed colorimeter for light measurement, and HDR[11] technology to capture lots of photos. Without explicitly modelling black offset of multi-display system, they could not resolve the dark background with low end input values. In fact, the black offset of LCD projectors is much larger than black offset of DLP projectors used in [7,10], the black offset of multi-projector display should be modelled explicitly for achieving visual seamlessness for low end input values.

Among previous work on photometric calibration, a few of them focused on the luminance variation[6,9]. For light measurement, some used special hardware [3,4,5] or HDR technology with tedious work to capture images [7,10]. They did not explicitly model the black offset of multi-projector display system to solve the dark background problem. For handing black offset, we proposed a three-phase approach to photometric calibration, which explicitly models the black offset of multi-projector display constructed using LCD projectors.

The rest of this paper is organized as follows. Section 2 describes the details of the three-phase photometric calibration method. Experimental results are given in Section 3. Finally, we draw the conclusions and point out the future work.

2 Three-Phase Photometric Model

Previous methods usually constructed the common display gamut [3,4,5,7], or smooth the input image [9] for achieving seamless display. Our three-phase photometric model is another description of the multi-projector display wall system. Using three-phase model, the chrominance and luminance problems can be described clearly. The model provides a better way for solving photometric problem, especially solving the black offset and color shift problems of display wall [6]. Previous work showed camera could be measurement tool for photometric calibration[12] [2], so our experiment use it instead of spectroradiometer or colorimeter.

The whole imaging process of displaying input RGB using multi-projector display involves three models in different phases. They are namely single-projector model, normalized-projector model, and display-wall model. Single-projector model characterizes the photometric characteristic of one projector.

Normalized-projector model describes how to transform each projector to a normalized one. Display-wall model shows how to form a large-scale seamless display wall system using several normalized projectors.

After transforming single-projector model to normalized-projector one, all projectors will have the same chrominance and holistic luminance. They should look like the same one, which can also be called standard projectors. Using standard projectors, the display wall model is constructed to solve the photometric problems, especially for handling low end input values. Using standard projectors to project content on screen, the multi-projector display can be regarded as one projector with the similar photometric properties with standard prjector.

This paper exploits the relationship among these models, and constructs a seamless multi-projector display. In the following, we first describe the generalized color matching process. Then, how to measure single-projector model is described. Based on the single-projector model, how to generate normalized-projector model and display-wall projector model are described step by step. Since LCD projector has simple parametric representation, we use it for problem formuation. For DLP projector [3,4], complex non-parametric description should be be used.

2.1 Generalized Color Matching Process

The color matching process of the projector can be regarded as a mapping between the RGB input in CIERGB space and the tristimulus values of output light in CIEXYZ space[3,4,5]. The entire process can be characterized as a Color Transfer Function $F : R^3 \rightarrow R^3$, $(X, Y, Z) = F(r, g, b)$ [5]. F is the mapping function, and r,g,b ϵ {0,1,...,255}.

Due to the channel independence [3,6] and regardless of black offset of LCD projector, the following equation is obtained [5]:

$$F(r, g, b) = F(r, 0, 0) + F(0, g, 0) + F(0, 0, b). \tag{1}$$

The operating system, graphics card, and projector have influences on each primary channel. Due to those factors, the mapping is nonlinear, and the Intensity Transfer Functions(ITFs) are commonly used to describe these influences. ITFs map pixel values to normalized intensity values for each primary color [3,4,6].

Three parameters(Y, x, y) could be used to described the color value, which could be measured by optical device. Y is the luminance and (x, y) are the chromaticity coordinates. The triangle formed by the chromaticity coordinates of three primaries is called the color gamuts of the display [6]. Each projector has its color gamut, which defines the color range that could be displayed. Due to the nonidentical color gamut of each projector, color differences appear when projecting the same input. The common color gamut of display should be determined to guarantee that any input from different projectors will have the same output chrominance response [10].

When the tristimulus values (X, Y, Z) in CIEXYZ are known, the corresponding chrominance values could be computed as:

$$x = X/(X+Y+Z) \quad y = Y/(X+Y+Z) \quad z = Z/(X+Y+Z) \tag{2}$$

The three-phase model described in following subsections will be used to express this process and solve the photometric problem existing in multi-projector display.

2.2 Single-Projector Model

The following equation is used to define the imaging procedure from the projector's input color values to its measured tristimulus values in CIEXYZ [3,12].

$$\mathbf{IM} + \mathbf{T_K} = \mathbf{T} \tag{3}$$

where

$$\mathbf{I} = \begin{pmatrix} ITF_R(r) \\ ITF_G(g) \\ ITF_B(b) \end{pmatrix}' \quad \mathbf{M} = \begin{pmatrix} X'_R & Y'_R & Z'_R \\ X'_G & Y'_G & Z'_G \\ X'_B & Y'_B & Z'_B \end{pmatrix} \quad \mathbf{T_K} = \begin{pmatrix} X_K \\ Y_K \\ Z_K \end{pmatrix}' \quad \mathbf{T} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}' \tag{4}$$

Given an input color $\mathbf{C} = (r, g, b)$, the vector $\mathbf{I} = (I_r, I_g, I_b)$ denotes the color intensity vector computed from the ITFs. The ITFs of projector for each primary channel are denoted as ITF_R, ITF_G and ITF_B respectively, and $\mathbf{I} = \text{ITFs(C)} = (ITF_R(r), ITF_G(g), ITF_B(b))$. Since the black offset of the projector is considered independent of the pixel value input into the projector, $\mathbf{T_K}$ is regarded as the tristimulus values for black. The matrix $\mathbf{M}$ is referred to as the color mixing matrix; (X_R, Y_R, Z_R), (X_G, Y_G, Z_G) and (X_B, Y_B, Z_B) are the tristimulus values for the primaries, subtracting black offset to form $\mathbf{M}$. The vector $\mathbf{T}$ represents the final tristimulus values which could be measured. This model is valid at every point in the projector plane.

Using (X, Y, Z) representation, Eq. (3) could also be described as:

$$\begin{aligned} X &= I_r X_R + I_g X_G + I_b X_B + X_K(1 - (I_r + I_g + I_b)) \\ Y &= I_r Y_R + I_g Y_G + I_b Y_B + Y_K(1 - (I_r + I_g + I_b)) \\ Z &= I_r Z_R + I_g Z_G + I_b Z_B + Z_K(1 - (I_r + I_g + I_b)) \end{aligned} \tag{5}$$

As following, we use $\mathbf{M_{P_i}}$ instead of $\mathbf{M}$ and $\mathbf{T_{K,P_i}}$ instead of $\mathbf{T_K}$ to describe single-projector model, and P_i represents the projector's ID. Combined with $\mathbf{T_K}$, we could also express this transformation as a single 4x4 homogeneous transformation matrix. So each projector model $\mathbf{M_{P_i}}$ could be described as:

$$M_{P_i} = \begin{pmatrix} X_{R,P_i} - X_{K,P_i} & Y_{R,P_i} - Y_{K,P_i} & Z_{R,P_i} - Z_{K,P_i} & 0 \\ X_{G,P_i} - X_{K,P_i} & Y_{G,P_i} - Y_{K,P_i} & Z_{G,P_i} - Z_{K,P_i} & 0 \\ X_{B,P_i} - X_{K,P_i} & Y_{B,P_i} - Y_{K,P_i} & Z_{B,P_i} - Z_{K,P_i} & 0 \\ X_{K,P_i} & Y_{K,P_i} & Z_{K,P_i} & 1 \end{pmatrix}$$

For constructing single-projector model, the ITFs and color mixing matrix of each projector is measured. Since the raw image of digital camera records the scene irradiance, so HDR technique is unnecessary [12]. After sample points are recorded, curve fitting can be used to restore the ITF curve approximately.

2.3 Normalized-Projector Model

From the single-projector model, the characteristics of one projector defines its gamut, or the set of realizable colors for that display. Each projector has its gamut, common gamut of all projectors should be defined [3]. Tsai et al. [7] use algorithm to adjust common gamut for chrominance problem. In the same spirit, we define normalized-projector model, which normalize each projector to make them have the same photometric characteristics.

To make every projector look like the same one, we should make all projectors reach the common color gamut. The standard gamut has similar shape and characteristics as the projector gamut, and must fit completely inside the gamuts [3] of all projectors. We use $\mathbf{M_s}$ and $\mathbf{T_{K,s}}$ to describe normalized-projector model. The normalized projector model $\mathbf{M_s}$ could also be described as:

$$M_s = \begin{pmatrix} X_{R,s} - X_{K,s} & Y_{R,s} - Y_{K,s} & Z_{R,s} - Z_{K,s} & 0 \\ X_{G,s} - X_{K,s} & Y_{G,s} - Y_{K,s} & Z_{G,s} - Z_{K,s} & 0 \\ X_{B,s} - X_{K,s} & Y_{B,s} - Y_{K,s} & Z_{B,s} - Z_{K,s} & 0 \\ X_{K,s} & Y_{K,s} & Z_{K,s} & 1 \end{pmatrix}$$

Here $\mathbf{T_{K,s}}$ uses the maximum value of $\mathbf{T_{K,P_i}}$, and i ϵ {1,2,...,n}, so each projector could reach it. When we compute values from captured images, all values are the average ones of selected regions. Since camera is our measurement tool, the average luminance could better describe the characteristics of each projector than single pixel. Using optical hardware, Stone used values read from device [3,4]. We specify the chromaticity coordinates of white to avoid color shift.

For obtaining $\mathbf{M_s}$ and $\mathbf{T_{K,s}}$, one variant algorithm similar to that in [3] is design as following:

1. Compute the maximum intersected chromaticity triangle who lies inside of the chromaticity triangles of all projectors. For each projector, the vertex's chromaticity coordinate of its triangle is that of the primary color red, green and blue, respectively. We select the maximum triangle which allow gamut to contain more realizable colors, while Stone select an inside one which just keeps reasonable.
2. Compute the chromaticity coordinates of black by averaging the ones of all projectors. Set the luminance Y of black as the maximum of all projectors. $\mathbf{T_{K,s}}$ is computed using Y and the chromaticity coordinates of black.
3. For each primary red, green, blue, the Y is selected as the minimum one of each primary channel of all projectors. Using Y and the chromaticity coordinates obtained in step 1, compute its (X, Z).
4. $(X_{R,s}, Y_{R,s}, Z_{R,s})$, $(X_{G,s}, Y_{G,s}, Z_{G,s})$ and $(X_{B,s}, Y_{B,s}, Z_{B,s})$ are vectors which stand for the tristimulus values measured for the primaries. $\mathbf{M_s}$ is constructed by subtracting $\mathbf{T_{K,s}}$ from the above vectors. To make sure that $\mathbf{M_s}$ is the suitable model, the following equation is used to adjust the matrix.

$$\mathbf{I_{P_i} M_{P_i}} = \mathbf{I_s M_s} \tag{6}$$

$\mathbf{I_s}$ is the color intensity of normalized projector, $\mathbf{I_{P_i}}$ is the color intensity of each projector. We use $\mathbf{s_R}$, $\mathbf{s_G}$ and $\mathbf{s_B}$ to stand for each row scale factors. After these row scale factors multiply each $\mathbf{M_s}$ row, a new $\mathbf{M_s}$ is obtained.

5. Specify the white point in the chromaticity coordinate graph as standard white. The specified white is used to adjust $\mathbf{M_s}$ again to reach white balance.

2.4 Display-Wall Model

Gradient domain smoothing algorithm generates good result in most cases after chrominance issues has been solved [7]. Without explicitly and accurately modelling the black offset of display wall, it is no easy to handle the low end input of LCD projectors (e.g. the black background in Fig.5(b)). Since the photometric model of display wall could be treated as one projector [7], any color characteristics could be redefined, including ITFs, chromaticity coordinates of display gamut and its own color mixing model. Inspired by these work, we construct the display-wall model using normalized-projector model.

We use $\mathbf{M_{dw}}$ to indicate the color matrix of display wall model, and $\mathbf{T_{K,dw}}$ to represent the black offset of this model. $\mathbf{M_{dw}}$ is expressed as:

$$M_{dw} = \begin{pmatrix} X_{R,dw} - X_{K,dw} & Y_{R,dw} - Y_{K,dw} & Z_{R,dw} - Z_{K,dw} & 0 \\ X_{G,dw} - X_{K,dw} & Y_{G,dw} - Y_{K,dw} & Z_{G,dw} - Z_{K,dw} & 0 \\ X_{B,dw} - X_{K,dw} & Y_{B,dw} - Y_{K,dw} & Z_{B,dw} - Z_{K,dw} & 0 \\ X_{K,dw} & Y_{K,dw} & Z_{K,dw} & 1 \end{pmatrix}$$

$\mathbf{M_{dw}}$ contains the tristimulus values for the primaries like single-projector model/ normalized-projector model. $\mathbf{T_{K,dw}} = (X_{K,dw}\ Y_{K,dw}\ Z_{K,dw}) \geq \mathbf{mT_{k,s}}$, where $\mathbf{T_{k,s}} = (X_{K,s}\ Y_{K,s}\ Z_{K,s})$ is described in normalized-projector model, which stands for the tristimulus value for black offset of display-wall model; $\mathbf{m}$ is the maximum number of overlap projectors on screen.

Using Eq. (5) in section 2.2, an expression between normalized-projector model and display-wall model can be derived. Take X component in the output tristimulus value as an example:

$$\begin{aligned} &I_{r,dw}X_{R,dw} + I_{g,dw}X_{G,dw} + I_{b,dw}X_{B,dw} + [1 - (I_{r,dw} + I_{g,dw} + I_{b,dw})]mX_{K,s} \\ &= (\sum_{i=1}^{n} \lambda_{P_i})(I_{r,s}X_{R,s} + I_{g,s}X_{G,s} + I_{b,s}X_{B,s}) + [n - (\sum_{i=1}^{n} \lambda_{P_i})(I_{r,s} + I_{g,s} + I_{b,s})]X_{K,s} \end{aligned} \tag{7}$$

The equations of Y and Z components can be derived similarly. These three components' equations establish a relationship between normalized-projector model and display-wall model. $\lambda_{\mathbf{P_i}}$ represent the contribution of each normalized projector; $\mathbf{m}$ stands for the maximum number of normalized projectors overlapped on one display pixel; $\mathbf{n}$ stands for the number of all projectors. $\mathbf{M_{dw}}$ is computed in the following ways:

1. Specify a new chromaticity triangle whose vertices' coordinates are inside of the normalized-projector. The original one could be directly used.

2. In $\mathbf{M_s}$, select Y in normalized-projector model as Y in display-wall model for each primary. Compute (X, Z) using Y and chromaticity coordinate to construct $\mathbf{M_{dw}}$.
3. $(X_{R,dw}, Y_{R,dw}, Z_{R,dw})$, $(X_{G,dw}, Y_{G,dw}, Z_{G,dw})$ and $(X_{B,dw}, Y_{B,dw}, Z_{B,dw})$ are are row vectors which stand for the tristimulus values for the primaries. We set $\mathbf{s_R}$, $\mathbf{s_G}$ and $\mathbf{s_B}$ are each row scale factor. Use Eq. (7) to get these scale factor values and recompute $\mathbf{M_{dw}}$.
4. Use the same algorithm like normalized-projector model to do white balance to adjust $\mathbf{M_{dw}}$ again.

2.5 Photometric Calibration Algorithm

With three models given above, we could compute re-input pixel color from given input one to achieve photometric calibration. The equation in the following explains the whole process.

$$\mathbf{C}' = ITFs_{P_i}^{-1}(\mathbf{M_{P_i}^{-1}}(\lambda_{\mathbf{P_i}}\mathbf{M_s}(G_{dw2n}(ITFs_{dw}(\mathbf{C}))))) \tag{8}$$

$\mathbf{C} = (r, g, b)$ is the input pixel value; $\mathbf{C}' = (r'_{P_i}, g'_{P_i}, b'_{P_i})$ is the re-input pixel value; $\mathrm{ITFs_{dw}}$ is the ITF of the display wall, and we use $y = x^2$; $\mathrm{G_{dw2n}}$ transforms the color intensity value from display-wall model to normalized-projector model; $\mathbf{M_s}$ simulates the normalized projector photometric process; $\lambda_{\mathbf{P_i}}$ is the weight of each normalized projector radiance, which is set in Eq. (7); $\mathbf{M_{P_i}^{-1}}$ will transform CIEXYZ tristimulus values to each projector intensity values. Compared with two-phase model[7], our algorithm could describe the problems more clearly, and explicitly model the black offset for handing low end input values. Although the single-projector models are different for different projectors, the normalized-projectors could be calibrated to a normalized one to guarantee the chrominance consistent and luminance uniformity. Based on this, the display-wall model could achieve visual seamlessness and color consistent display.

3 Experimental Results

3.1 Setup

Two brands of LCD projectors are used. one is SANYO PLC-XT3200, the other is EPSON EMP-8300. The camera is Cannon EOS 20D. The experiment is carried out on one cylindrical screen. All experiments are done in a dark room, and the environment light has slight influence.

To facilitate color balancing on a multi-projector display, the color characteristics of the projectors should be measured. Traditionally, a sophisticated mechanical device such as colorimeter or spectroradiometer is utilizedc [6,9]. we use a camera to capture raw images for achieving the same results [12].

3.2 Result

Fig.1 describes one chromaticity coordinate calibration result using raw images. Fig.1(a) describes red, green, blue and white chromaticity coordinates of two projectors before calibration, Fig.1(b) shows the result after calibration.

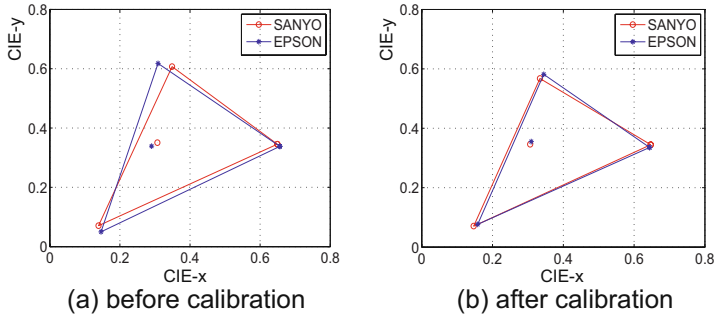


Fig. 1. Comparison of two projector chromaticity coordinates before and after correction

Fig.2, Fig.3 and Fig.4 show the snapshots of one display wall constructed using the aforementioned two projectors. (a) shows snapshots of using single-projector model with no photometric calibration. From the captured image in (a), there are obvious chrominance and luminance problems. (b) shows the results using blending algorithm [8] after applying ITFs. (c) shows the results with three-phase approach, which are visually uniform and seamless.

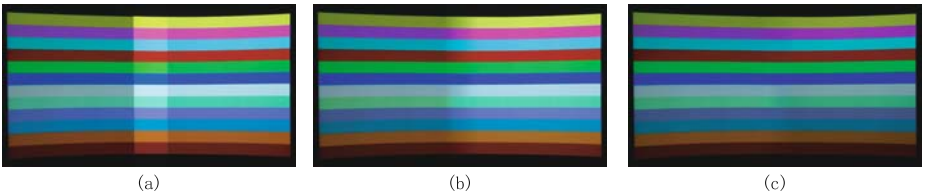


Fig. 2. Experimental result. (a) The result without any calibration. (b) The result with blending Method. (c) The result with the three-phase approach.

For dark background image with low end input values, the photometric calibration is difficult without explicitly modelling the black offset. Fig.5 shows one case that has noticeable seam and chrominance difference. Even if chrominance between two projectors has little difference after calibration [7] and blending operation [8] is used, noticeable beam still exits (Fig.5(b)) due to the black offset of projectors. Using three-phase approach to refine the black level, dark background image appears visually seamless (Fig.5(c)).

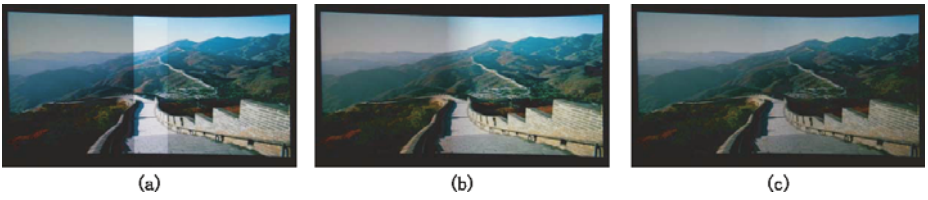


Fig. 3. Experimental result. (a) The result without any calibration. (b) The result with blending Method. (c) The result with the three-phase approach.

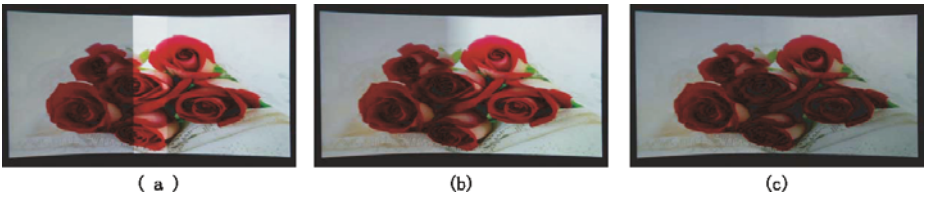


Fig. 4. Experimental result. (a) The result without any calibration. (b) The result with blending Method. (c) The result with the three-phase approach.

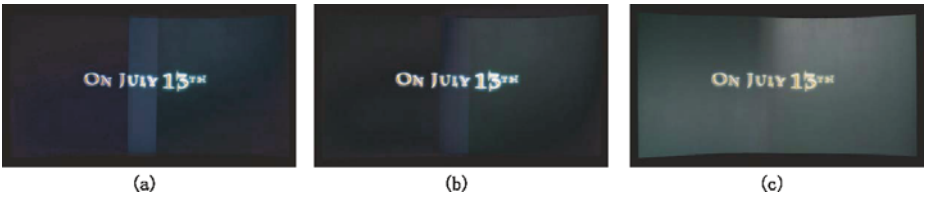


Fig. 5. Three photos were captured with a very low exposure to show dark background situation. (a) The result without any calibration. (b)The result with chrominance calibration and blending. (c) The result with the three-phase approach.

4 Conclusions and Future Work

In this paper, we proposed a three-phase approach to do photometric calibration for multi-projector display using LCD projectors. From the experiments, our approach can handle photometric problems of chrominance variation and luminance nonuniformity. It models the whole imaging process from pixel input to projector out clearly. It explicitly models the black offset of multi-projector display, and can achieve visual seamlessness for low end input values. The luminance control of display wall is implemented by changing the whole luminance of one projector, the main reason is that the current commercial projector has good holistic and uniform luminance. In the future, we will generalize the three-phase approach to solve photometric problems of DLP projectors.

Acknowledgments

Correspondence to Zhongding Jiang. This work is partially supported by NSFC and 863 Program of China (Grant No.60803064 and 2006AA01Z325).

References

1. Ni, T., Schmidt, G., Staadt, O., Livingston, M., Ball, R., May, R.: A survey of large high-resolution display technologies, techniques, and applications. In: IEEE Virtual Reality, pp. 223–236 (2006)
2. Brown, M., Majumder, A., Yang, R.: Camera-based calibration techniques for seamless multiprojector displays. IEEE Transactions on Visualization and Computer Graphics 11(2), 193–206 (2005)
3. Stone, M.C.: Color and brightness appearance issues in tiled displays. IEEE Computer Graphics and Applications 21, 58–66 (2001)
4. Stone, M.: Color balancing experimental projection displays. In: 9th IST/SID Color Imaging Conference (2001)
5. Wallace, G., Chen, H., Li, K.: Color gamut matching for tiled display walls. In: ACM International Conference Proceeding Series, vol. 39, pp. 293–302 (2003)
6. Majumder, A., Stevens, R.: Color nonuniformity in projection-based displays: analysis and solutions. IEEE Transactions on Visualization and Computer Graphics 10(2), 177–188 (2004)
7. Tsai, Y., Liao, S., Shih, Z., Hung, Y.: Two-phase photometric calibration for multiprojector displays. In: Third Taiwanese-French Conference on Information Technology (2006)
8. Raskar, R., Brown, M.S., Yang, R., Chen, W., Welch, G., Towles, H., Seales, B., Fuchs, H.: Multi-projector displays using camera-based registration. In: IEEE Visualization (1999)
9. Majumder, A., Steven, R.: Perceptual photometric seamlessness in projection-based tiled displays. ACM Transactions on Graphics 24(1), 118–139 (2005)
10. Tsai, Y., Wu, Y., Liao, S., Shih, Z., Hung, Y.: Automatic geometric and photometric calibration for tiling multiple projectors with a pan-tilt-zoom camera. In: IEEE International Conference on Multimedia and Expo., pp. 401–404 (2006)
11. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: SIGGRAPH, pp. 369–378 (1997)
12. Jiang, Z., Mao, Y., Qin, B., Zang, B.: A high resolution video display system by seamlessly tiling multiple projectors. In: IEEE International Conference on Multimedia and Expo., pp. 2070–2073 (2007)

Twisted Cubic: Degeneracy Degree and Relationship with General Degeneracy

Tian Lan, YiHong Wu, and Zhanyi Hu

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, P.O. Box 2728, 100190, Beijing, China
{tlan,yhwu,zyhu}@nlpr.ia.ac.cn

Abstract. Fundamental matrix, drawing geometric relationship between two images, plays an important role in 3-dimensional computer vision. Degenerate configurations of space points and two camera optical centers affect stability of computation for fundamental matrix. In order to robustly estimate fundamental matrix, it is necessary to study these degenerate configurations. We analyze all possible degenerate configurations caused by twisted cubic and give the corresponding degenerate rank for each case. Relationships with general degeneracies, the previous ruled quadric degeneracy and the homography degeneracy, are also reported in theory, where some interesting results are obtained such as a complete homography relation between two views. Based on the result of the paper, by applying RANSAC for degenerate data, we could obtain more robust estimations for fundamental matrix.

1 Introduction

Fundamental matrix describes geometric relation between two 2-dimensional views. It plays an important role in image matching, epipolar geometry, camera motion determination, camera self-calibration and 3-dimensional reconstruction. Robust and accurate estimation for fundamental matrix has been the research focus of extensive researchers[1,2,3,4,5,6,7,8].

From at least seven pairs of point-point correspondences between two views, the fundamental matrix can be estimated. Sometimes, a reliable estimation cannot be obtained, no matter how many correspondences are used. One of the main reasons is that the cameras and the scene lie on a degenerate or quasi-degenerate configuration. If a space configuration is degenerate mathematically but the noise from the measured image makes it non-degenerate, any estimation under such a configuration would be useless [9]. It follows that we should know what configurations might cause degeneracy for estimating the fundamental matrix. Moreover in order for a robust RANSAC like [10,7], we still need to know how great the degenerate degree is, namely, to know the degenerate rank of the coefficient matrix of the equations for computing fundamental matrix.In [3], RANSAC loop to estimate relation from quasi-degenerate data is reported,

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 66–77, 2010.

where the degenerate configurations need not be known. This is not equivalent to say the studies on the degenerate configurations are useless. At least, such studies can give more geometric intuition, which could be as guidance for placing cameras to avoid degeneracy in practice. Furthermore, if we can judge the degeneracy by applying geometric knowledge, RANSAC work will be much easier.

Due to the importance of degeneracy analysis, many of such works have been reported previously. The planar scene is a trivial degenerate configuration for computing fundamental matrix, where the images can provide only six independent constraints [11,7] but the general fundamental matrix has seven degrees of freedom. Degeneracy from twisted cubic configuration has also been discussed. In [12], Buchanan stated that camera calibration from known space points under a single view is not unique if the optical center and the space points lie on a twisted cubic. The corresponding detection as well as emendations including other unreliability was given by Wu et al [13]. Then, under two views, Maybank [14] analyzed the characterizations of horopter curve and the relations between the curve and the ambiguous case of reconstruction. The horopter curve is regarded as a twisted cubic, which intersects the plane at infinity at three particular points. The ambiguous case of reconstruction implies ambiguity of fundamental matrix. Luong and Faugeras reported the stability for computing fundamental matrix caused by quadric critical surface in [15]. Hartley and Zisserman [11] also gave systematic discussions for degeneracy of camera projection estimation from twisted cubic under a single view and for degeneracy from ruled quadric surface under two views. Under three views, critical configurations are provided in [16], which is an extension of the critical surface under two views. Degeneracy under a sequence of images is also investigated [17,18]. Maybank and Shashua [18] pointed out there is a three-way ambiguity for reconstruction from images of six points when the six points and the camera optical centers lie on a hyperboloid of one sheet. In [17], Hartley and Kahl presented a classification of all possible critical configurations for any number of points from three images and showed that in most cases, the ambiguity could extend to any number of cameras.

Relative to the above works on degenerate configurations, there are fewer deep studies on degeneracy degrees of degenerate configurations. Torr et al [7] catalogued all two-view non-degenerate and degenerate cases in a logical way by dimensions of the right null space of equations on fundamental matrix and then proposed a PLUNDER-DL method to detect degeneracy and outliers. Chum et al [10] also analyzed those dimensions when the two views or most of the point correspondences are related by a homography and presented an algorithm to estimate fundamental matrix through detecting the homography degeneracy. They all [7,10] generalized the robust estimator RANSAC [19]. The plane degeneracy in [7,10] is consistent with the ruled quadric degeneracy proposed by Hartley and Zisserman [11] because a plane and two camera optical centers always lie on a degenerate ruled quadric. What are the degeneracy degrees when estimating the fundamental matrix for other non-trivial degenerate configurations? In this

paper, we discuss all possible degenerate situations caused by twisted cubic and give the corresponding degeneracy degrees. Let **SO** be a set of space points and the two camera optical centers. We find that if all the points of **SO** lie on a twisted cubic, the configuration is degenerate for estimating fundamental matrix and the corresponding rank of coefficient matrix is five; if all the points other than one lie on a twisted cubic, the corresponding rank is six; if all the points other than two lie on a twisted cubic, the corresponding rank is seven. The previous general degeneracies are ruled quadric degeneracy and homography degeneracy. Few studies are given on relationships of twisted cubic degeneracy with them. We investigate the relationships in detail and then present our contribution relative to the general degeneracies.

The organization of the paper is as follows. Some preliminaries are listed in Section 2. The complete and unified degeneracy study from twisted cubic is elaborated in Section 3. Some experimental results are displayed in Section 4 and Section 5 makes some conclusions.

2 Preliminaries

The camera model used is a perspective camera. A space point or its homogeneous coordinates is denoted by $\mathbf{M}$, an image point or its homogeneous coordinates is denoted by $\mathbf{m}$, $\mathbf{P}$ denotes the camera projection matrix, and $\mathbf{O}$ denotes the camera optical center. Under two views, $\mathbf{P}'$ denotes the second camera projection matrix, $\mathbf{O}'$ denotes its optical center, and $\mathbf{m}'$ denotes the corresponding image point of $\mathbf{m}$. Let $\mathbf{F}$ be the fundamental matrix between the two views. Other vectors or matrices are also denoted in boldface. The symbol $\approx$ means equality up to a scale.

Camera Projection Matrix: $\mathbf{M_i}$, $i = 1 \ldots N$ are 3-dimensional space points. And their corresponding image points are $\mathbf{m_i}$, $i = 1 \ldots N$. The camera projection matrix $\mathbf{P}$ is a 3×4 matrix such that $\mathbf{m_i} \approx \mathbf{PM_i}$. For the camera optical center $\mathbf{O}$, we have the equation:

$$\mathbf{PO} = \mathbf{0} \tag{1}$$

Fundamental matrix: Let $\mathbf{m_i'}$ be the corresponding image points of the space points $\mathbf{M_i}$ under another view. Then, $\mathbf{m_i}$ and $\mathbf{m_i'}$ are related by the fundamental matrix $\mathbf{F}$ through:

$$\mathbf{m_i'^T F m_i} = \mathbf{0}, \quad i = 1 \ldots N \tag{2}$$

We denote $\mathbf{F}$ as $\begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}$ If $\mathbf{m_i} \approx \begin{pmatrix} u_i & v_i & w_i \end{pmatrix}$ and $\mathbf{m_i'} \approx \begin{pmatrix} u_i' & v_i' & w_i' \end{pmatrix}$, we expand (2) and have:

$$\begin{pmatrix} \cdots \\ u_i'u_i \;\; u_i'v_i \;\; u_i'w_i \;\; v_i'u_i \;\; v_i'v_i \;\; v_i'w_i \;\; w_i'u_i \;\; w_i'v_i \;\; w_i'w_i \\ \cdots \end{pmatrix}_{N \times 9} \mathbf{f} = \mathbf{0} \tag{3}$$

where $\mathbf{f} = \begin{pmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 \end{pmatrix}^T$ is the vector consisting of all elements in $\mathbf{F}$. The $N \times 9$ coefficient matrix of $\mathbf{f}$ is denoted by $\mathbf{G}$.

Twisted cubic: The locus of points $\mathbf{X} = \begin{pmatrix} X & Y & Z & T \end{pmatrix}^T$ in a 3-dimensional projective space satisfying the parametric equation:

$$\begin{pmatrix} X & Y & Z & T \end{pmatrix}^T \approx \mathbf{H} \begin{pmatrix} \theta^3 & \theta^2 & \theta & 1 \end{pmatrix}^T \tag{4}$$

is a twisted cubic, where $\mathbf{H}$ is a 4×4 matrix and θ is the parameter[20]. Twisted cubic is an extension of a conic to 3-dimensional space by increasing the degree of curve parameter from two to three. The properties of twisted cubic underlie many of the ambiguous cases that arise in 3-dimensional reconstruction.

3 Degeneracies from Twisted Cubic

The previously known degenerate configuration of two views for fundamental matrix or projective reconstruction is that two camera optical centers and all space points lie on a ruled quadric. For such a general ruled quadric, the right null space of $\mathbf{G}$ in (3) is of dimension two as given in the section 2 of [7] and in the paragraph five of the introduction section of [16]. The more critically degenerate configuration is from a plane, of which the right null space of $\mathbf{G}$ in (3) is of dimension three [10,7]. This is not at the most since the nontrivial degenerate configuration—-twisted cubic can cause more critically degeneracy than a plane as shown below.

3.1 Degeneracy Degree from Twisted Cubic

In (3), if the rank of the coefficient matrix $\mathbf{G}$ is 8, then $\mathbf{F}$ can be determined uniquely by linear 8-point algorithm. Otherwise, if the rank of $\mathbf{G}$ is 7, the solution of $\mathbf{f}$ from (3) has one degree of freedom and the freedom can be removed by $\det(\mathbf{F}) = 0$ to obtain three or one solution. But if we only rely on the linear equations (3), the freedom cannot be removed. If the rank is 6 or less than 6, solutions of $\mathbf{f}$ has two or more degrees of freedom and so $\mathbf{F}$ cannot be determined finitely. The configuration making the rank of $\mathbf{G}$ deficient is degenerate for computing $\mathbf{F}$. Due to noise of image data, generally we always can calculate a unique solution of $\mathbf{f}$ from (3) with 8 corresponding points. However, the degenerate configurations or the configurations near to degeneracy will terribly influence stability of the calculation. Therefore, in order for robust estimation of fundamental matrix, we need to know the degenerate configurations. The degenerate configurations from twisted cubic and the corresponding degeneracy degrees are provided in the following theorem.

Theorem 1. *Let* **SO** *be a set of space points and two camera optical centers for capturing these points. If all the points of* **SO** *are on a twisted cubic, then the rank of the coefficient matrix* $\mathbf{G}$ *for computing* $\mathbf{F}$ *is five. If all the points other than one of* **SO** *are on a twisted cubic, the rank of* $\mathbf{G}$ *is six. If all the points other than two of* **SO** *are on a twisted cubic, the rank of* $\mathbf{G}$ *is seven.*

Proof: Firstly, we give the proof when **SO** are all on a twisted cubic.
According to (4), assume the parametric equation of this twisted cubic is $\mathbf{H}\left(\theta^3\ \theta^2\ \theta\ 1\right)^T$, where $\mathbf{H}$ is a 4×4 matrix. Let the parameter of the space point $\mathbf{M_i}$ be θ_i and the parameters of the two camera optical centers be θ_0, θ'_0. Then,$\mathbf{M_i} = \mathbf{H}\left(\theta_i^3\ \theta_i^2\ \theta_i\ 1\right)^T$, $\mathbf{O} = \mathbf{H}\left(\theta_0^3\ \theta_0^2\ \theta_0\ 1\right)^T$, $\mathbf{O'} = \mathbf{H}\left(\theta_0'^3\ \theta_0'^2\ \theta'_0\ 1\right)^T$. By (1), we have:

$$\mathbf{0} = \mathbf{PO} = \mathbf{PH}\left(\theta_0^3\ \theta_0^2\ \theta_0\ 1\right)^T, \quad \mathbf{0} = \mathbf{P'O'} = \mathbf{P'H}\left(\theta_0'^3\ \theta_0'^2\ \theta'_0\ 1\right)^T \tag{5}$$

where $\mathbf{P}$, $\mathbf{P'}$ are the two camera projection matrices. So we also have:

$$\mathbf{m_i} \approx \mathbf{PM_i} = \mathbf{PH}\left(\theta_i^3\ \theta_i^2\ \theta_i\ 1\right)^T, \quad \mathbf{m'_i} \approx \mathbf{P'M_i} = \mathbf{P'H}\left(\theta_i'^3\ \theta_i'^2\ \theta'_i\ 1\right)^T \tag{6}$$

Do subtraction from both sides for (5) and (6), we obtain:

$$\begin{aligned}\mathbf{m_i} &\approx \mathbf{PH}\left(\theta_i^3\ \theta_i^2\ \theta_i\ 1\right)^T - \mathbf{PH}\left(\theta_0^3\ \theta_0^2\ \theta_0\ 1\right)^T \\ &\approx \mathbf{PH}(\theta_i-\theta_0)\left(\theta_i^2+\theta_0^2+\theta_0\theta_i\ \ \theta_i+\theta_0\ \ 1\ 0\right)^T \\ &\approx \mathbf{PH}\left(\theta_i^2+\theta_0^2+\theta_0\theta_i\ \ \theta_i+\theta_0\ \ 1\ 0\right)^T\end{aligned} \tag{7}$$

Denote $\mathbf{PH}$ as $\mathbf{Q} = \begin{pmatrix} q_1 & q_2 & q_3 & q_4 \\ q_5 & q_6 & q_7 & q_8 \\ q_9 & q_{10} & q_{11} & q_{12}\end{pmatrix}$, and $\mathbf{P'H}$ as $\mathbf{Q'} = \begin{pmatrix} q'_1 & q'_2 & q'_3 & q'_4 \\ q'_5 & q'_6 & q'_7 & q'_8 \\ q'_9 & q'_{10} & q'_{11} & q'_{12}\end{pmatrix}$.
Then, (7) is changed into:

$$\mathbf{m_i} \approx \begin{pmatrix} q_1 \\ q_5 \\ q_9 \end{pmatrix}(\theta_i^2+\theta_0^2+\theta_0\theta_i) + \begin{pmatrix} q_2 \\ q_6 \\ q_{10} \end{pmatrix}(\theta_i+\theta_0) + \begin{pmatrix} q_3 \\ q_7 \\ q_{11} \end{pmatrix} \tag{8}$$

Similarly, do subtraction from both sides for (5) and (6), there is:

$$\mathbf{m'_i} \approx \begin{pmatrix} q'_1 \\ q'_5 \\ q'_9 \end{pmatrix}(\theta_i^2+{\theta'_0}^2+\theta'_0\theta_i) + \begin{pmatrix} q'_2 \\ q'_6 \\ q'_{10} \end{pmatrix}(\theta_i+\theta'_0) + \begin{pmatrix} q'_3 \\ q'_7 \\ q'_{11} \end{pmatrix} \tag{9}$$

θ_i varies with $\left(\mathbf{m_i}\ \mathbf{m'_i}\right)$, while q_k, q'_k, θ_0, θ'_0 are unchanged.
Substitute (8) and (9) into (3), we get the coefficient matrix $\mathbf{G}$ with each element of the i-th row being a four-order polynomial in θ_i as $c_1\theta_i^4 + c_2\theta_i^3 + c_3\theta_i^2 + c_4\theta_i + c_5$. The coefficients c_s of θ_i in these four-order polynomials are functions on while q_k, q'_k, θ_0, θ'_0. Since while q_k, q'_k, θ_0, θ'_0 are not varying with image pair varying, c_s are also not varying with the row number varying. It follows that $\mathbf{G}$ is in this form:

$$\mathbf{G} = \begin{pmatrix} \cdots \\ g_1(\theta_i)\ g_2(\theta_i)\ g_3(\theta_i)\ g_4(\theta_i)\ g_5(\theta_i)\ g_6(\theta_i)\ g_7(\theta_i)\ g_8(\theta_i)\ g_9(\theta_i) \\ \cdots \end{pmatrix} \tag{10}$$

where $g_j(\theta) = c_{1j}\theta^4 + c_{2j}\theta^3 + c_{3j}\theta^2 + c_{4j}\theta + c_{5j}, j = 1 \ldots N$. We equivalently change $\mathbf{G}$ into: $\mathbf{G} = \begin{pmatrix} \theta_1^4 & \theta_1^3 & \theta_1^2 & \theta_1 & 1 \\ \cdots & & & & \\ \theta_i^4 & \theta_i^3 & \theta_i^2 & \theta_i & 1 \\ \cdots & & & & \end{pmatrix}_{N\times 5} \begin{pmatrix} c_{11} & \ldots & c_{1j} & \ldots & c_{19} \\ c_{21} & \ldots & c_{2j} & \ldots & c_{29} \\ c_{31} & \ldots & c_{3j} & \ldots & c_{39} \\ c_{41} & \ldots & c_{4j} & \ldots & c_{49} \\ c_{51} & \ldots & c_{5j} & \ldots & c_{59} \end{pmatrix}_{5\times 9}$

From the expression, we know the rank of $\mathbf{G}$ is generally five. By now, we proved that if all the space points and the optical centers of the two cameras are on a twisted cubic, the rank of the coefficient matrix G is five.

If a camera optical center does not lie on the twisted cubic determined by another camera optical center and the space points, assumed to be $\mathbf{O}$, then the degree of θ_i for representing $\mathbf{m_i}$ in (6) can not decrease to two but the degree for representing $\mathbf{m_i'}$ can do, i.e. $\mathbf{m_i'}$ is still in the form (9). Thus, the degrees of θ_i in the obtained coefficient matrix $\mathbf{G}$ of (10) become into 5. Then by the same reason as above, we have the corresponding rank 6. If the point not lying on the twisted cubic is one of the space points other than one of the camera optical center, assumed to be $\mathbf{M_{i_0}}$, then the row in $\mathbf{G}$ from the image pair $\mathbf{m_{i_0}}$, $\mathbf{m'_{i_0}}$ is not in the polynomial form of some θ. It follows that this row is not linearly related to other rows in general. Thus, the rank of $\mathbf{G}$ increases from five to six.

Similarly, if all the points other than two of $\mathbf{SO}$ are in a twisted cubic, the rank of $\mathbf{G}$ is seven. The theorem is proved.

In the above theorem, we analyze all possible degenerate configurations for computing $\mathbf{F}$ caused from twisted cubic. In all the cases, $\mathbf{F}$ can not be determined finitely by linear 8-point algorithm and the dimensions of the right null space of $\mathbf{G}$ in (3) are respectively 4, 3, 2. By 7-point algorithm, $\mathbf{F}$ still can not be solved in rank 5, 6 cases but can be solved in rank 7 case.

3.2 Relationship with Ruled Quadric Degeneracy

The degenerate configuration of two views for reconstruction is well known as a ruled quadric [11]. The theorem in Section 3.1 is consistent with the ruled quadric degeneracy. In this subsection, we at first give two lemmas about twisted cubic and ruled quadric for the consistency. Then, the contribution of our work is discussed.

In projective space, quadrics are classified into ruled and unruled ones. Quadrics with positive index of inertia 2 are ruled quadrics and the degenerate quadrics except one point case are all ruled ones [11]. Here the positive index of inertia means the number of positive entries in the canonical form for a quadric.

Lemma 1. *In a 3-dimensional projective space, a proper real twisted cubic can always be embedded on a ruled quadric, conversely, any quadric containing a proper real twisted cubic is a ruled one.*

Due to space limit, the proof is omitted. It is similar for the following lemmas.

Lemma 2. *In a 3-dimensional projective space, if seven points of a real proper twisted cubic lie on a quadric, then the whole twisted cubic lies on the quadric.*

Remark 1. By Lemma 1 and Lemma 2, we conclude that a twisted cubic plus one or two points can be embedded on a ruled quadric. We take seven points on the twisted cubic and combine the additional one or two points to generate a quadric. This is reasonable because generally nine space points uniquely determine a quadric. Since this quadric contains seven points of the twisted cubic, by Lemma 2, we know it contains the whole twisted cubic. Furthermore by Lemma 1, we know the generated quadric is ruled. It follows that the theorem in Section 3.1 is consistent with the previous ruled quadric degeneracy.

Remark 2. The contribution of Theorem 1 is that it gives more intuitive degeneracy and the degeneracy degrees for all possible cases caused by twisted cubic. For the general ruled quadric degeneracy, there are a finite number of solutions for the fundamental matrix by combining with the additional constraint of $\det(\mathbf{F}) = 0$. This degeneracy degree is the same as the rank 7 case in the theorem. For rank 5, 6 cases in the theorem, the degeneracy is more critical which makes the fundamental matrix free in a four- or three-dimensional space. Even though by the additional constraint $\det(\mathbf{F}) = 0$, it cannot be solved. These details are not discussed in the previous ruled quadric degeneracy. Usually, six points determine a unique twisted cubic and nine points determine a unique quadric. A twisted cubic is not a class in the ruled quadrics. Therefore, from fewer non-incidence points to make $\mathbf{F}$ computations, quadric degeneracy may not come to mind, which also could ignore the twisted cubic degeneracy. However indeed the twisted cubic can make the $\mathbf{F}$ computation degenerate severely as shown in the theorem in Section 3.1.

3.3 Relationship with H-degeneracy

One previous work closely related to ours is the H-degeneracy studied by Chum et al. [10], where the H-degeneracy means the degeneracy caused by a 3×3 homography between two views. They also discussed the degeneracy degrees for the $\mathbf{F}$ computation and mentioned the twisted cubic degeneracy. There are differences between our work and theirs. In this subsection, we discuss the contribution of our work relative to the study [10].

Firstly, we give complete cases that two views are related by a 3×3 homography.

Lemma 3. *If the image point correspondences* $(\mathbf{m_i}, \mathbf{m'_i})$ *between two views are related by a homography* $\mathbf{H}$ *, that is* $\mathbf{m'_i} = \mathbf{Hm_i}$*, then generally there are the following complete three situations:*
1) *The camera performs a pure rotation;*
2) *The space points are coplanar;*
3) *The space points and the two camera optical centers lie on a twisted cubic.*

The above classification of the three cases are complete. In [10], Chum et al analyzed degrees of the H-degeneracy on three cases: i) two views are related by a homography; ii) all image point pairs other than one pair are related by a

homography; iii) all image point pairs other than two pairs are related by a homography. Then based on the degrees, they developed a DEGENSAC algorithm to compute **F** unaffected by a dominant plane by detecting H-degeneracy.

The relationship and differences between our work and Chum et al's [10] are as follows. The cases in the theorem of Section 3.1 related to the H-degeneracy are: ($a1$) The two camera optical centers and all the space points lie on a twisted cubic. ($a2$) The two camera optical centers and all other than one the space points lie on a twisted cubic. ($a3$) The two camera optical centers and all other than two the space points lie on a twisted cubic.

According to Lemma 3, the two views in (a1) are related by a homography, in ($a2$) the image point pairs except for one pair are related by a homography, and the image point pairs except for two pairs are related by a homography in ($a3$). Although these geometric relations between the two views in the three cases are the same as Chum et al's, the degeneracy degrees are different. Here in our work, the degeneracy is more critical. For case ($a1$), since the coefficient matrix has rank 5, the linear space of **F** has dimension 4 while in [10] for two views related with a homography the dimension is 3. For case ($a2$), the corresponding dimension is 3 while that in [10] is 2. For case ($a3$), the corresponding dimension is 2 while that in [10] could be 1 if linear 8-point algorithm is applied. It follows that the twisted cubic cases could cause more critical degeneracy than the plane cases, though they have the same geometric H-relations between the two views.

The cases in the theorem of Section 3.1 not involved in [10] are: ($b1$) All the space points and one of the camera optical centers lie on a twisted cubic. The other camera optical center is not on this twisted cubic. ($b2$) All other than one of the space points and one of the camera optical centers lie on a twisted cubic. The other camera optical center is not on this twisted cubic. ($b3$) All the space points but the two camera optical centers lie on a twisted cubic.

The three cases do not fall into the work of [10]. In the three cases at least one of the optical centers does not lie on the twisted cubic and the space points are also not coplanar. Thus according to Lemma 3 all or most of the image point pairs in each case ($b1$), ($b2$), ($b3$) do not agree to a homography relation.

Therefore, our work not only develops the work in [10] but also makes some new contribution in theory. The aim of [10] is to stably estimate **F** unaffected by a dominant plane. We also will explore a detection method on the degeneracy caused from twisted cubic and then apply the RANSAC on degenerate data in [13] to robustly compute fundamental matrix. Detection on the degeneracy deserves studies also because usually computations of matrix rank or its singular values are very sensitive to noise and presetting a threshold to discriminate the degeneracy from the non-degeneracy is not easy, as pointed out in [21].

4 Experiment

We performed both simulations and experiments on real data. The results verify the established theorem. One group of the experiments is reported below.

4.1 Simulations

The parametric equation of a space twisted cubic is:

$$\mathbf{M} \approx \begin{pmatrix} 2 & 5 & -3 & 2.5 \\ 1 & -1 & 12 & 1 \\ 6 & -15 & -2 & 3 \\ -7 & 5 & 3 & 2 \end{pmatrix} \begin{pmatrix} \theta^3 \\ \theta^2 \\ \theta \\ 1 \end{pmatrix} \tag{11}$$

Ten points $\mathbf{M_i}$ on this twisted cubic are taken, of which the parameters are respectively -1.1, -0.35, -0.75, -0.22, -0.6, 0.1, -0.1, 0.2, 1.9, -2.

At first, we consider the case of that both the two optical centers and the space points lie on the same twisted cubic. Let the two points of the twisted cubic with parameters 1.25, 1.5 be the two optical centers $\mathbf{O}$, $\mathbf{O'}$. The space distribution is shown as Fig.1. Then, the corresponding camera projection matrices consistent with the optical centers are set as follows:$\mathbf{P} = \begin{pmatrix} 1000 & 0 & 512 & 43198 \\ 0 & 900 & 384 & 95484 \\ 0 & 0 & 1 & -103 \end{pmatrix}$, $\mathbf{P'} = \begin{pmatrix} -529 & 648.1 & 287.4 & -4321.3 \\ 338.6 & -295.4 & 748.7 & -1810.1 \\ -0.7 & -0.1 & 0.7 & -3.9 \end{pmatrix}$. Projected by $\mathbf{P}$, $\mathbf{P'}$, we generated two simulated images of the ten space points and established the equations on the fundamental matrix. Under the noise level of zero, the rank of the coefficient matrix of these equations could be computed out and the result is as five.

We also tested the case when one of the optical centers does not lie on the twisted cubic any more. Let $\mathbf{Q_2} = \begin{pmatrix} 3 & 7.3 & 2 & 1 \end{pmatrix}^T$ which is not on this twisted cubic, the corresponding camera projection matrix is set as: $\mathbf{P_2} = \begin{pmatrix} 1000 & 0 & 512 & -4024 \\ 0 & 900 & 384 & -7338 \\ 0 & 0 & 1 & -2 \end{pmatrix}$. By this camera projection matrix, another new image is generated. From this image and that of $\mathbf{P'}$, we established equations on the fundamental matrix and then computed the rank of the coefficient matrix under noise level of zero. The result is six that is consistent with the proposed theorem.

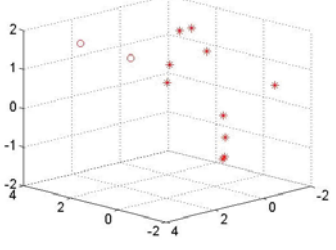


Fig. 1. Space points and two optical centers lie on a twisted cubic, where * denotes the space points, and o denotes the camera optical centers

Finally, we give the experimental result of the case when the two optical centers do not lie on the twisted cubic (11). Another optical center is set as $\mathbf{Q}'_{\mathbf{2}} = \begin{pmatrix} 0.67 & -1.49 & -2.8 & 1 \end{pmatrix}^T$ which is also far away from the twisted cubic (11). From the two images generated, we computed rank of the coefficient matrix of the equations on the fundamental matrix and the result is seven.

If there are one or two of the space points that do not lie on the twisted cubic determined by other space points and the optical centers, the same results are obtained. All the experimental results validate the theorem in Section 3.1. However, we find that the direct computation on the matrix rank or the rank computation by the singular values is only correct in the absence of noise. When we add noise to the image, the rank of the coefficient matrix becomes to 8 and the computation becomes very unstable. Therefore, in order to robustly estimate the fundamental matrix, it is necessary to develop a method of detection on the degenerate configuration. We will explore a detection method on the degeneracy caused from twisted cubic and apply the RANSAC on degenerate data in [19,13] to robustly compute the fundamental matrix.

4.2 Experiments on Real Data

We tested the degeneracy of six points from real data. The experiments of more points on real data need to be performed after the detection on degenerate data and the corresponding RANSAC are proposed.

We took the images of six space points at different viewpoints. Four of them with a size of 640×480 pixels are shown in Fig. 2, where the dot points denote the used image points.

In order to know whether the six space points and the corresponding optical center lie on a twisted cubic or not, we measured the space coordinates of the six points and then by the criterion function proposed in [13] detected the situation. The values of the criterion function on the four images in Fig. 2 are respectively 1.0655, 1.0504, 2.2934, and 2.5091. Then, by the method in [13], we know that the six points and the two corresponding optical centers of Fig. 2 (a)(b) are on a same twisted cubic, while the six points and the two corresponding optical centers of (c)(d) are not. We also computed the singular values of the coefficient matrix $\mathbf{G}$ in (3). The result from the two images in Fig. 2 (a)(b) is: 623427.73, 156095.86, 41657.74, 6772.02, 79.53,9.81. And the result from the two images in Fig. 2 (c)(d) is:508796.41, 138904.18, 33040.13, 9883.42, 112.31, 37.68. We see that the condition number of coefficient matrix $\mathbf{G}$ from (a) (b)in Fig. 2 is larger than that from (c)(d). However, usually it is difficult to detect the degeneracy by using the condition number because the singular values are very sensitive to noise and presetting a threshold to discriminate the degeneracy from the non-degeneracy is not easy, as pointed out in [21]. We found sometimes the condition number of the degeneracy is yet smaller than that of the non-degeneracy. This is why we would like to pursue a detection method for the degeneracy from two image data in the future.

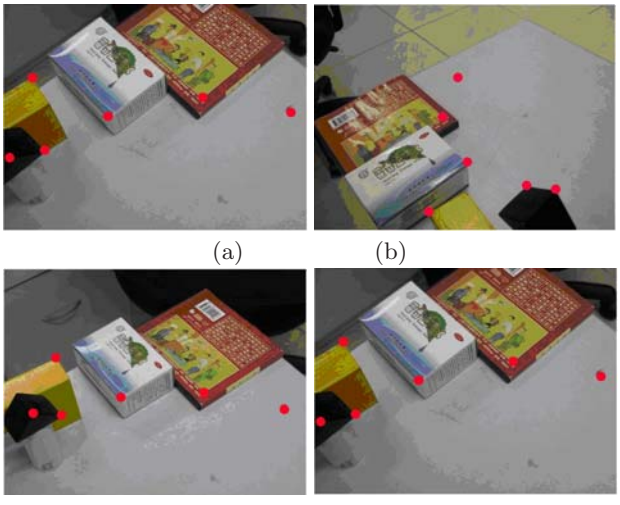


Fig. 2. Images of six points, where the space points and the two camera optical centers (a)(b): are on a same twisted cubic; (c)(d): are not on a same twisted cubic

5 Conclusion

This paper provides all the possible degenerate configurations caused by twisted cubic and the corresponding degeneracy degrees for estimating fundamental matrix. Relationships with the ruled quadric degeneracy and the homograghy degeneracy are also given. The result is helpful to improving the accuracy of the estimations. Indeed, for a robust RANSAC, initial samples with worse estimations should be removed or mended. These initial samples not only are those including mismatching pairs but also are those that are degenerate. The latter case usually is ignored by people but really affects stability of the computations. The reason of the ignorance may be that the degeneracy has not been studied thoroughly. We give some research on the degeneracy in this work and further robust detection on the twisted cubic configurations will be developed.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under grant No. 60633070, 60773039.

References

1. Bartoli, A., Sturm, P.: Non-linear estimation of the fundamental matrix with minimal parameters. Pattern Analysis and Machine Intelligence 26, 426–432 (2004)
2. Bober, M., Georgis, N., Kittler, J.: On accurate and robust estimation of fundamental matrix. Computer Vision and Image Understanding 72, 39–53 (1998)

3. Frahm, J., Pollefeys, M.: Ransac for (quasi-)degenerate data (qdegsac). Computer Vision and Pattern Recognition 1, 453–456 (2006)
4. Hartley, R.: In defense of the eight-point algorithm. Pattern Analysis and Machine Intelligence 19, 580–593 (1997)
5. Luong, Q.T., Faugeras, O.: The fundamental matrix: Theory, algorithms, and stability analysis. International Journal of Computer Vision 17, 43–76 (1996)
6. Torr, P.H.S., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. International Journal of Computer Vision 23, 271–300 (1997)
7. Torr, P.H.S., Zisserman, A., Maybank, S.J.: Robust detection of degenerate configurations while estimating the fundamental matrix. Computer Vision and Image Understanding 71, 312–333 (1998)
8. Zhang, Z.: Determining the epipolar geometry and its uncertainty: A review. International Journal of Computer Vision 27, 161–195 (1998)
9. Huang, T.S., Ahuja, J.N.: Motion and structure from two perspective views: algorithms, error analysis, and error estimation. Pattern Analysis and Machine Intelligence 11, 451–476 (1989)
10. Chum, O., Werner, T., Matas, J.: Two-view geometry estimation unaffected by a dominant plane. Computer vision and Pattern recognition, 772–779 (2005)
11. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambridge (2000)
12. Buchanan, T.: The twisted cubic and camera calibration. Computer Vision, Graphics and Image 42, 130–132 (1988)
13. Wu, Y., Li, Y., Hu, Z.: Detecting and handling unreliable points for camera parameter estimation. International Journal of Computer Vision 79, 209–223 (2008)
14. Maybank, S.: Theory of reconstruction from image motion. Springer, Heidelberg (1992)
15. Luong, Q.T., Faugeras, O.: A stability analysis of the fundamental matrix. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 577–588. Springer, Heidelberg (1994)
16. Hartley, R.: Ambiguous configurations for 3-view projective reconstruction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 922–935. Springer, Heidelberg (2000)
17. Hartley, R., Kahl, F.: Critical configurations for projective reconstruction from multiple views. International Journal of Computer Vision 71, 5–47 (2006)
18. Maybank, S., Shashua, A.: Ambiguity in reconstruction from images of six points. In: International Conference on Computer Vision, pp. 703–708 (1992)
19. 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)
20. Semple, J.G., Kneebone, G.T.: Algebraic projective geometry. Oxford University, Oxford (1952)
21. Kahl, F., Henrion, D.: Globally optimal estimates for geometric reconstruction problems. International Journal of Computer Vision 74, 3–15 (2007)

Two-View Geometry and Reconstruction under Quasi-perspective Projection

Guanghui Wang and Q.M. Jonathan Wu

Department of Electrical and Computer Engineering, University of Windsor
401 Sunset, Windsor, ON, Canada N9B 3P4
ghwangca@gmail.com, jwu@uwindsor.ca

Abstract. Two-view geometry under quasi-perspective camera model and some new results are reported in the paper. Firstly, we prove that quasi fundamental matrix can be simplified to a special form with six degrees of freedom and it is invariant to any non-singular projective transformation. Secondly, the plane induced homography under quasi-perspective model can be simplified to a special form defined by six degrees of freedom. Quasi homography may be recovered from only two pairs of correspondences with known fundamental matrix. Extensive tests on synthetic and real images are performed to validate the results.

1 Introduction

Reconstructing three-dimensional information from stereo views of a scene is a fundamental problem in computer vision. Many approaches have been proposed during the last two decades for different applications. The most typical algorithm is stereo vision technique from two images [5]. In case of image sequences, we usually adopt factorization based algorithm [8] to recover the structure and motion parameters.

All structure from motion algorithms are based on certain assumption of camera model. The most popular one is pinhole camera model, which is often referred to as perspective projection. This is an ideal and accurate model for general imaging process. However, perspective projection is a nonlinear transformation and is complicated due to the the unknown perspective scalar [7]. To simplify the computation, researchers proposed orthographic, weak perspective, and paraperspective projection model, which can be generalized as affine camera [6] [9]. Affine camera is a linear approximation and is valid when the distance of camera to object is much greater than the size of the object itself. More recently, Wang *et al.* [12] proposed a quasi-perspective projection model to fill the gap between simplicity of affine camera and accuracy of perspective projection. The model assumes small camera movement.

Fundamental matrix estimation is a central problem in stereo vision as it encapsulates the underlying epipolar geometry. Classical linear estimation technique for fundamental matrix is sensitive to noise. Hartley [4] analyzed the problem and proposed a normalized eight-point algorithm to improve the accuracy. Zhang and Kanade [14] gave a good review on fundamental matrix estimation and uncertainty analysis. Random sample consensus (RANSAC) paradigm [3] was originated for robust parameter estimation in present of outliers that the least-squares techniques may be severely affected.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 78–87, 2010.

Torr *et al.* [10] proposed to adopt RANSAC to estimate fundamental matrix. Dellaert *et al.* [1] also proposed a robust method to reject outliers and reconstruct 3D scene geometry.

The quasi-perspective projection model [12] [13] was originally proposed for factorization based structure recovery from image sequence. In this paper, we will carry out a further investigation of two-view geometry under the model. Some results are similar to those under perspective and affine camera model. There seems no such report to the best of our knowledge.

2 Camera Projection Geometry

Different camera models are proposed to formulate the geometry of imaging process. The most ideal one is perspective projection model. Under this model, a 3D point $\mathbf{X}_i$ is projected onto an image point $\mathbf{x}_i$ according to equation

$$\lambda_i \mathbf{x}_i = \mathbf{P}\mathbf{X}_i = \mathbf{K}[\mathbf{R}, \mathbf{T}]\mathbf{X}_i \tag{1}$$

where $\mathbf{P}$ is called projection matrix; $\mathbf{x}_i = [u_i, v_i, 1]^T$ and $\mathbf{X}_i = [x_i, y_i, z_i, 1]^T$ are expressed in homogeneous form; $\mathbf{R}$ and $\mathbf{T}$ are the corresponding rotation matrix and translation vector of the camera with respect to world coordinate system; $\mathbf{K}$ is camera calibration matrix; λ_i is a non-zero scale factor, commonly called projective depth.

When the distance of an object from a camera is much greater than the depth variation of the object, we may assume affine camera model. Under affine assumption, the last row of the projection matrix is of the form $\mathbf{P}_3^T \simeq [0, 0, 0, 1]$, where '$\simeq$' denotes equality up to scale. Then the projection process (1) can be simplified by removing the scale factor λ_i.

$$\bar{\mathbf{x}}_i = \mathbf{A}\bar{\mathbf{X}}_i + \bar{\mathbf{T}} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{2\times 3}$ is composed by the upper-left 2×3 submatrix of $\mathbf{P}$; $\bar{\mathbf{x}}_i = [u_i, v_i]^T$ and $\bar{\mathbf{X}}_i = [x_i, y_i, z_i]^T$ are the non-homogeneous form of $\mathbf{x}_i$ and $\mathbf{X}_i$ respectively; $\bar{\mathbf{T}}$ is the corresponding translation vector.

Assuming the camera is far away from the object and undergoes small rotations and movement, Wang *et al.* [12] proposed a quasi-perspective projection model and the imaging process is approximated as

$$\mathbf{x}_i = \mathbf{P}_q \mathbf{X}_{qi} \tag{3}$$

where $\mathbf{X}_{qi}$ is scale weighted space point in homogeneous form; $\mathbf{P}_q \in \mathbb{R}^{3\times 4}$ is called quasi-perspective projection matrix, whose last row is of the form $\mathbf{P}_3^T \simeq [0, 0, *, *]$, where $'*'$ stands for nonzero entry. Clearly, the matrix $\mathbf{P}_q$ has only 10 nonzero entries and 9 degrees of freedom (DOF) since it is defined up to scale. While the general perspective projection matrix $\mathbf{P}$ has 11 DOFs.

Under quasi-perspective projection, it is easy to verify that ideal points in X and Y directions of world system are mapped to ideal points in an image. Thus the parallelism in X and Y are invariant, but the parallel relation along Z axis is not preserved under the projection model (3).

3 Two-View Geometry and Reconstruction

3.1 Fundamental Matrix

Epipolar geometry is the intrinsic projective geometry between a pair of stereo images. Suppose $\mathbf{x}$ and $\mathbf{x}'$ are corresponding points in a stereo image pair, then the intrinsic geometry between the images can be encapsulated as follows.

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0 \tag{4}$$

where $\mathbf{F} \in \mathbb{R}^{3\times3}$ is called fundamental matrix. If the camera is calibrated, the two view geometry can also be expressed by essential matrix which is defined as $\mathbf{E}_q = \mathbf{K}'^T\mathbf{F}_q\mathbf{K}$. Both the fundamental matrix and essential matrix are rank 2 homogeneous matrices, thus they have only 7 DOFs.

Proposition 1. *The fundamental matrix and essential matrix under quasi-perspective projection can be simplified to the form of* $\begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix}$*, which is defined by 6 DOFs.*

We will now derive the quasi fundamental matrix $\mathbf{F}_q$. Suppose $\mathbf{R}$ is the relative rotation of the second camera with respect to the first. Let us decompose the rotation as three angles α, β, γ along the three axes X, Y, Z respectively. Then we have

$$\mathbf{R}=\mathbf{R}(\gamma)\mathbf{R}(\beta)\mathbf{R}(\alpha) = \begin{bmatrix} \mathcal{C}\gamma & -\mathcal{S}\gamma & 0 \\ \mathcal{S}\gamma & \mathcal{C}\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{C}\beta & 0 & \mathcal{S}\beta \\ 0 & 1 & 0 \\ -\mathcal{S}\beta & 0 & \mathcal{C}\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathcal{C}\alpha & -\mathcal{S}\alpha \\ 0 & \mathcal{S}\alpha & \mathcal{C}\alpha \end{bmatrix} \tag{5}$$

where '$\mathcal{S}$' stands for sine function, and '$\mathcal{C}$' stands for cosine function. Given rotation $\mathbf{R}$ and translation $\mathbf{t} = [t_x, t_y, t_z]^T$ between two views, the essential matrix can be computed from

$$\mathbf{E}_q = [\mathbf{t}]_\times\mathbf{R} = \begin{bmatrix} t_z\mathcal{S}\gamma\mathcal{C}\beta+t_y\mathcal{S}\beta & t_z(\mathcal{C}\gamma\mathcal{C}\alpha+\mathcal{S}\gamma\mathcal{S}\beta\mathcal{S}\alpha)-t_y\mathcal{C}\beta\mathcal{S}\alpha & * \\ -t_z\mathcal{C}\gamma\mathcal{C}\beta-t_x\mathcal{S}\beta & t_z(\mathcal{S}\gamma\mathcal{C}\alpha-\mathcal{C}\gamma\mathcal{S}\beta\mathcal{S}\alpha)+t_x\mathcal{C}\beta\mathcal{S}\alpha & * \\ * & * & * \end{bmatrix} \tag{6}$$

Under quasi-perspective assumption, the camera undergoes small movement and rotations, then we have $\{\mathcal{S}\alpha, \mathcal{S}\beta, \mathcal{S}\gamma\} \to 0$, $\{\mathcal{C}\alpha, \mathcal{C}\beta, \mathcal{C}\gamma\} \to 1$, which results to $\{e_{11}, e_{22}\} \to 0$. Therefore the essential matrix is simplified to

$$\mathbf{E}_q = \begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{21} & 0 & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix} \tag{7}$$

Suppose the camera parameters are fixed as $\mathbf{K} = \mathbf{K}' = \begin{bmatrix} f_1 & 0 & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}$. Then the fundamental matrix can be obtained from

$$\mathbf{F}_q = \mathbf{K}'^{-T}\mathbf{E}\mathbf{K}^{-1} = \begin{bmatrix} e_{11}/f_1^2 & e_{12}/(f_1f_2) & * \\ e_{21}/(f_1f_2) & e_{22}/f_2^2 & * \\ * & * & * \end{bmatrix} = \begin{bmatrix} 0 & f_{12} & f_{13} \\ f_{21} & 0 & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix} \tag{8}$$

Proposition 2. *Given two camera matrices* $\mathbf{P}_q$ *and* $\mathbf{P}'_q$*, the fundamental matrix can be recovered from* $\mathbf{F}_q = [\mathbf{e}']_\times\mathbf{P}'_q\mathbf{P}_q^+$*, where* $\mathbf{P}_q^+$ *denotes the pseudo-inverse of* $\mathbf{P}_q$*. The fundamental matrix is invariant to any non-singular projective transformation* $\mathbf{H} \in \mathbb{R}^{4\times4}$*. i.e.* $\mathbf{F}_q$ *remains the same if we set* $\mathbf{P}_q \leftarrow \mathbf{P}_q\mathbf{H}$*,* $\mathbf{P}'_q \leftarrow \mathbf{P}'_q\mathbf{H}$*.*

The proof is omitted here. It can be obtained similarly as in the case of perspective projection [5][14].

Under affine assumption, the optical center of an affine camera locates at infinity, it follows that all epipolar lines are parallel and both epipoles are at infinity. Thus the fundamental matrix is simplified to the form of

$$\mathbf{F}_a = \begin{bmatrix} 0 & 0 & * \\ 0 & 0 & * \\ * & * & * \end{bmatrix} \tag{9}$$

We can see that $\mathbf{F}_a$ is already of rank 2 with 4 DOFs.

3.2 Plane Induced Homography

When the space points are coplanar, we may assume the plane equation as $Z = 0$ without loss of generality, then the quasi-perspective projection (3) is simplified to

$$\mathbf{x}_i \simeq \mathbf{H}_\pi \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & h_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{10}$$

where $\mathbf{H}_\pi \in \mathbb{R}^{3\times 3}$ is called homography or perspectivity. There are 6 DOFs in $\mathbf{H}_\pi$ and it can be recovered from 3 non-collinear space points with known positions. For coplanar space points, their images in two views are related with a planar homography, we call it plane induced homography.

Proposition 3. *Under quasi-perspective projection, the plane induced homography can be simplified to the form of* $\mathbf{H}_q = \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{bmatrix}$ *with 6 DOFs.*

Proof. Suppose $\mathbf{x}$ and $\mathbf{x}'$ are images of coplanar space point $\mathbf{X}$ in the two views, $\mathbf{H}_\pi$ and $\mathbf{H}'_\pi$ are the perspectivities of the two views. Then we have

$$\mathbf{x} \simeq \mathbf{H}_\pi \mathbf{X},\ \mathbf{x}' \simeq \mathbf{H}'_\pi \mathbf{X} \tag{11}$$

By eliminating $\mathbf{X}$ from (11), we have $\mathbf{x}' \simeq \mathbf{H}'_\pi \mathbf{H}_\pi^{-1} \mathbf{x} = \mathbf{H}_q \mathbf{x}$, where $\mathbf{H}_q$ is the plane induced homography and it can be written as

$$\mathbf{H}_q = \mathbf{H}'_\pi \mathbf{H}_\pi^{-1} = \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ 0 & 0 & h'_{33} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & h_{33} \end{bmatrix}^{-1} = \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{bmatrix} \tag{12}$$

The homography $\mathbf{H}_q$ is a full rank matrix with 6 DOFs, and at least 3 non-collinear corresponding points can give an unique solution. □

It is easy to verify that the homography under affine camera model has the same form as (12). While general homography under perspective model has 8 DOFs and at least 4 points are required for computation.

Proposition 4. *Given fundamental matrix* $\mathbf{F}_q$*, then the plane induced homography* $\mathbf{H}_q$ *may be recovered from two pairs of correspondences* $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i, i = 1, 2$.

The result is obvious, since an additional correspondence of the epipoles $\mathbf{e} \leftrightarrow \mathbf{e}'$ can be obtained from the fundamental matrix as $\mathbf{F}_q \mathbf{e} = 0, \mathbf{F}_q^T \mathbf{e}' = 0$. Thus we have 3 pairs of correspondences and the homography $\mathbf{H}_q$ can be uniquely determined if the two image points are not collinear with the epipole.

Table 1. The trial number for different models to ensure probability $p = 99\%$ under different minimal subsets and outlier-to-inlier ratios

Model		Minimal subset	Outlier-to-inlier ratio 10%	20%	40%	60%	80%	100%
Fundamental	Persp	8/7	8/7	18/15	66/47	196/122	506/280	1177/588
	Quasi	6/5	6/5	12/9	33/23	75/46	155/85	293/146
	Affine	4	5	7	16	28	47	72
Homography	Persp	4	5	7	16	28	47	72
	Quasi	3	4	6	11	17	25	35
	Affine	3	4	6	11	17	25	35

3.3 RANSAC Algorithm with Outliers

In above analysis, we assume all correspondences are inliers without mismatches. The result may be severely disturbed in present of outliers. In this case, we usually adopt the RANdom SAmple Consensus (RANSAC) algorithm [3] to eliminate outliers and obtain a robust estimation. RANSAC algorithm is an iterative method to estimate parameters of a mathematical model and is computationally intensive. We will give a comparison on trial number for different projection models.

Suppose the outlier-to-inlier ratio is $k = \frac{N_{outlier}}{N_{inlier}}$, the number of the minimum subset to estimate the model is n, and we want to ensure that at least one of the random samples is free from outliers with a probability of p. Then the trial number N must satisfy $1 - p = \left(1 - (\frac{1}{k+1})^n\right)^N$, which leads to

$$N = \frac{ln(1-p)}{ln\left(1 - (\frac{1}{k+1})^n\right)} \tag{13}$$

Under a given probability p, the number of trials depends on the proportion k of outliers over inliers and the number of subset n. In practice, we usually select a conservative probability $p = 99\%$. Table 1 shows the required trial number under different conditions. We can see from the table that the required trial number increases sharply with the increase of subset n and outlier ratio k. The algorithm under quasi-perspective is computationally less intensive than that under perspective projection, especial for large proportion of outliers. For fundamental estimation of general perspective projection, we may adopt normalized 8-point linear algorithm [4] or 7-point nonlinear algorithm with rank-2 constraint [14]. Similarly, we have normalized 6-point linear algorithm and 5-point nonlinear algorithm for quasi-perspective fundamental matrix. We can adopt simple linear algorithm when the ratio k is small. However, it is wise to adopt nonlinear algorithm for larger outlier ratios.

3.4 3D Structure Reconstruction

Quasi-perspective projection is a special case of perspective projection, thus most theories on 3D reconstruction under perspective model may be applied directly to quasi-perspective. In case of calibrated cameras, we can recover the camera matrices directly

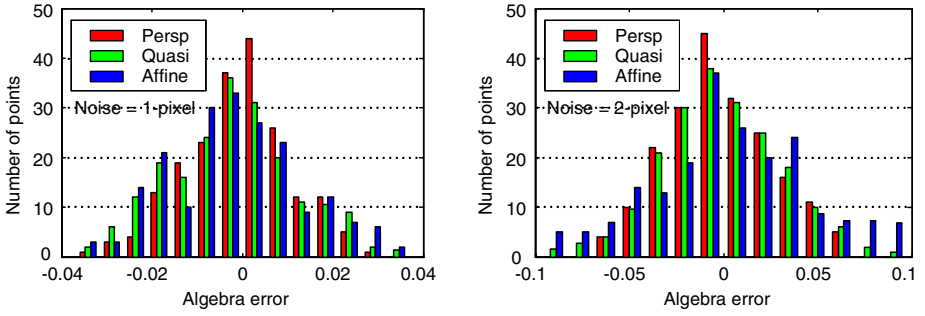


Fig. 1. The algebra error distribution of fundamental matrix under different camera models. The results are obtained with 1- and 2-pixel Gaussian noise.

from singular value decomposition (SVD) of the essential matrix [5]. For uncalibrated cameras, we can adopt stratified reconstruction [2] to recover the 3D structure. Here we will give some properties of quasi-perspective reconstruction.

Result 5. *Under quasi-perspective assumption, a pair of canonical cameras can be defined as*

$$\mathbf{P}_q = [\,\mathbf{I}\,|\,\mathbf{0}\,],\ \mathbf{P}'_q = [\,\mathbf{M}_q\,|\,\mathbf{t}\,] \tag{14}$$

where $\mathbf{M}_q$ *is a* 3×3 *matrix with its last row in form of* $[\,0, 0, *\,]$.

Result 6. *Suppose* $\left(\mathbf{P}_{q1}, \mathbf{P}'_{q1}, \{\mathbf{X}_{1i}\}\right)$ *and* $\left(\mathbf{P}_{q2}, \mathbf{P}'_{q2}, \{\mathbf{X}_{2i}\}\right)$ *are two quasi-perspective reconstructions of a set of correspondences* $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ *between two images. Then the two reconstructions are defined up to a quasi-perspective transformation as* $\mathbf{P}_{q2} = \mathbf{P}_{q1}\mathbf{H}_q$, $\mathbf{P}'_{q2} = \mathbf{P}'_{q1}\mathbf{H}_q$, *and* $\mathbf{X}_{2i} = \mathbf{H}_q^{-1}\mathbf{X}_{1i}$, *where the transformation* $\mathbf{H}_q$ *is a* 4×4 *non-singular matrix in form of*

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{A}_{2\times 2} & \mathbf{B}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{C}_{2\times 2} \end{bmatrix} \tag{15}$$

4 Evaluation on Synthetic Data

During simulation, we randomly generated 200 points within a cube of $20 \times 20 \times 20$ in space, and simulated two images from these points by perspective projection. The image size is set at 800×800. The camera parameters are set as follows. The focal lengths are set randomly between 1000 and 1100. The three rotation angles are set randomly between $\pm 5^\circ$. The X and Y positions of the cameras are set randomly between ± 15, while the Z positions are set randomly between 210 to 220. The synthetic imaging condition is close to affine and quasi-perspective assumption.

Evaluation on fundamental matrix: We recovered $\mathbf{F}_q$ by normalized 6-point algorithm, and calculated the algebra error of the fundamental matrix as $\varepsilon_{1i} = \mathbf{x}_i'^T \mathbf{F}_q \mathbf{x}_i$. The distribution of the errors across the 200 correspondences is outlined in Fig. 1. Gaussian image noise was added to each image point during the test. As a comparison, we

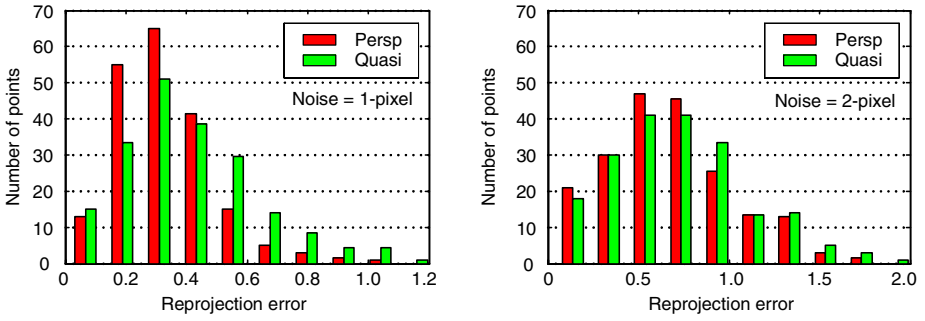


Fig. 2. Histogram distribution of reprojection errors by plane induced homography of two images. The results are obtained with 1- and 2-pixel Gaussian noise

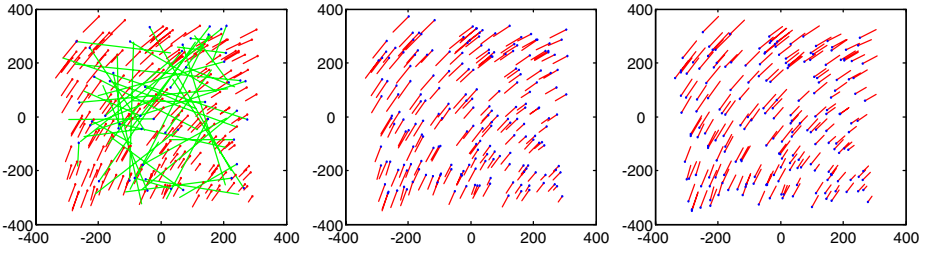


Fig. 3. Results of RANSAC algorithm. (a) one image matches with outliers; (b)&(c) final detected correspondences in two images after RANSAC.

also recovered the general fundamental matrix $\mathbf{F}$ by normalized 8-point algorithm and the affine fundamental matrix $\mathbf{F}_a$ by 4-point linear algorithm. We can see that the error of quasi-perspective lies in between those under perspective projection and affine.

Evaluation on homography: We set all space points on the plane $Z = 10$ and re-generated two images with the same camera parameters. Then we recovered the plane induced homography $\mathbf{H}_q$ and $\mathbf{H}$ under quasi-perspective and perspective projection respectively, and evaluated the reprojection error as

$$\varepsilon_{2i} = \frac{1}{2}\Big(d(\mathbf{x}_i, \mathbf{H}_q^{-1}\mathbf{x}_i')^2 + d(\mathbf{x}_i', \mathbf{H}_q\mathbf{x}_i)^2\Big) \tag{16}$$

where $d(*,*)$ denotes the Euclidean distance of two feature points. The histogram distributions of the errors under different noise level are shown in Fig. 2. We can see than the error given by $\mathbf{H}_q$ is higher than that by $\mathbf{H}$. The homography under affine model is the same as $\mathbf{H}_q$.

Evaluation on RANSAC: We randomly added 50 mismatches to the initial generated correspondences and adopted RANSAC paradigm to estimate $\mathbf{F}_q$ and eliminate outliers. The initial matches with disparities and outliers are shown in the first plot of

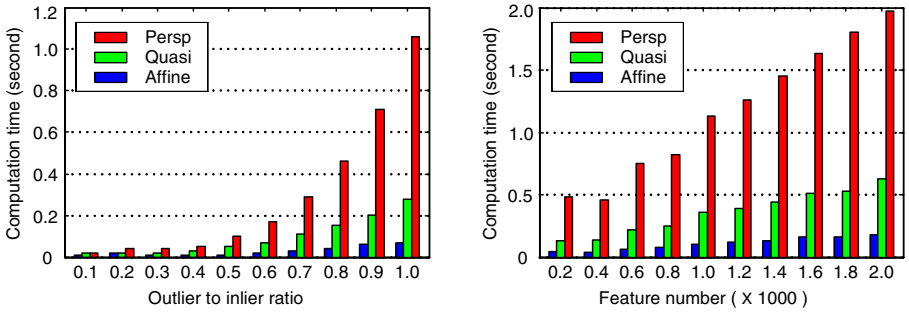


Fig. 4. The average computation time under different camera models with respect to (a) different outlier-to-inlier ratios and (b) feature point numbers

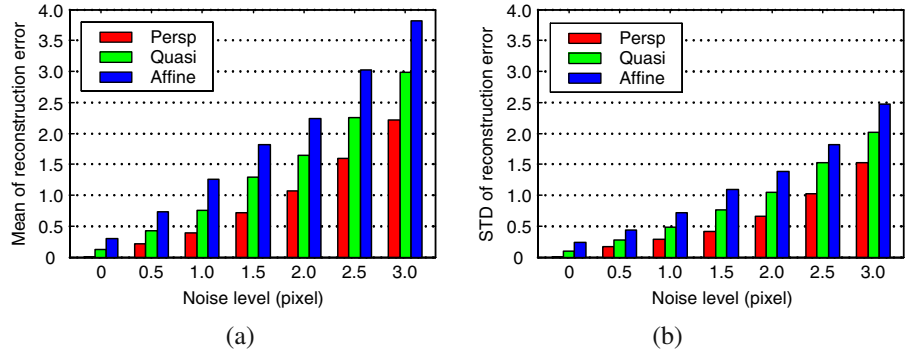
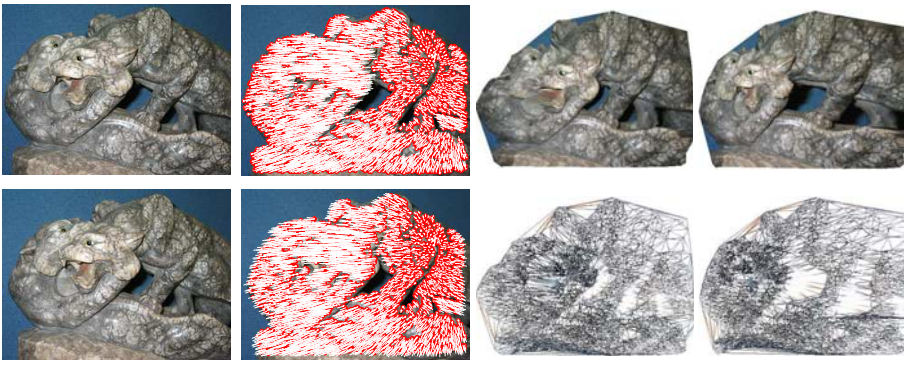


Fig. 5. Evaluation on 3D reconstruction. The mean (a) and standard deviation (b) of reconstruction errors by different models.

Fig. 3. All mismatches were rejected by the algorithm. We then calculated the average computation time in estimating the fundamental matrix under different model. Only linear algorithm was adopted and the minimal subsets for $\mathbf{F}$, $\mathbf{F}_q$, $\mathbf{F}_a$ are set as 8, 6, and 4 respectively. The program was implemented with Matlab R14 on Dell Inspiron 600m laptop of Pentium(R) 1.8GHz CPU. In first case, we selected 200 correspondences and varied the outlier-to-inlier ratio from 0.1 to 1.0. In second case, we set the outlier ratio at 0.8 and varied the feature number from 200 to 2000. The result is shown in Fig. 4. We can see that the algorithm under quasi-perspective model is faster than that under perspective projection, especially for larger data sets and outlier ratios.

Evaluation on reconstruction: We reconstructed the 200 points under different camera models and registered the result with the ground truth. The reconstruction error was calculated as pointwise distance between the recovered structure and its ground truth. In order to obtain a statistical meaningful result, we varied the image noise from 0 to 3 pixels at a step of 0.5, and took 100 independent tests at each noise level. The mean and



(a) (b)

Fig. 6. Reconstruction result of stone dragon. (a) Two images and the tracked features with relative disparities; (b) the reconstructed VRML model of the object and the corresponding wireframe shown from different viewpoints.

standard deviation of the errors are shown in Fig. 5. The reconstruction under quasi-perspective is more accurate than that with affine assumption.

5 Evaluation on Real Images

We tested and compared different models on many real images, and we will report one result in the paper due to limited space. The test is on two images of a stone dragon that were captured by Canon Powershot G3 camera with a resolution of 1024×768. The camera parameters were calibrated offline. The correspondences were established by the matching system based on SIFT and epipolar constraint [11], and totally 4621 reliable features were corrected matched as shown in Fig.6. We recovered the fundamental matrix under quasi-perspective projection and then reconstructed the 3D structure of the scene via factorization of the essential matrix [5]. Fig.6 shows the reconstructed model with texture mapping and the corresponding wireframe viewed from different viewpoints. The structure of the dragon is correctly recovered and looks realistic. After reconstruction, we reprojected the 3D structure to the two images and calculated the reprojection errors. The mean errors by perspective projection, quasi-perspective, and affine model are 0.72, 0.80, and 0.86 respectively. The quasi-perspective result outperforms that under affine camera model.

6 Conclusion

In this paper, we have investigated the two-view geometry of quasi-perspective projection model and presented some special properties of the fundamental matrix, plane induced homography, and 3D reconstruction under the model. Both theocratical and experimental analysis show that quasi-perspective model is a good tradeoff between the

simplicity of affine and the accuracy of perspective projection. One possible limitation of quasi-perspective projection is the requirement of small camera rotations and movement. However, the constraint is usually satisfied in practice, since we tend to constrain camera movement during image taking so as to guarantee large overlapping between images and facilitate feature matching process.

Acknowledgment

The work is supported in part by Natural Sciences and Engineering Research Council of Canada, and the National Natural Science Foundation of China under Grant No. 60575015.

References

1. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure from motion without correspondence. In: Proc. CVPR, pp. 2557–2564 (2000)
2. Faugeras, O.: Stratification of 3-D vision: projective, affine, and metric representations. Journal of the Optical Society of America A 12, 465–484 (1995)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
4. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Trans. Pattern Anal. Mach. Intell. 19(6), 580–593 (1997)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
6. Mundy, J.L., Zisserman, A.: Geometric Invariance in Computer Vision. The MIT Press, Cambridge (1992)
7. Oliensis, J., Hartley, R.: Iterative extensions of the sturm/triggs algorithm: convergence and nonconvergence. IEEE Trans. Pattern Anal. Mach. Intell. 29(12), 2217–2233 (2007)
8. Poelman, C., Kanade, T.: A paraperspective factorization method for shape and motion recovery. IEEE Trans. Pattern Anal. Mach. Intell. 19(3), 206–218 (1997)
9. Shapiro, L.S., Zisserman, A., Brady, M.: 3D motion recovery via affine epipolar geometry. International Journal of Computer Vision 16(2), 147–182 (1995)
10. Torr, P.H.S., Zisserman, A., Maybank, S.J.: Robust detection of degenerate configurations while estimating the fundamental matrix. Computer Vision and Image Understanding 71(3), 312–333 (1998)
11. Wang, G.: A hybrid system for feature matching based on SIFT and epipolar constraint. Tech. Rep. Department of ECE, University of Windsor (2006)
12. Wang, G., Wu, J.: Quasi-perspective projection with applications to 3D factorization from uncalibrated image sequences. In: Proc. CVPR, pp. 1–8 (2008)
13. Wang, G., Wu, J.: 3D Quasi-perspective projection model: theory and application to structure and motion factorization from uncalibrated image sequences. International Journal of Computer Vision (2009) (in press), doi:10.1007/s11263-009-0267-4
14. Zhang, Z., Kanade, T.: Determining the epipolar geometry and its uncertainty: A review. International Journal of Computer Vision 27(2), 161–195 (1998)

Similarity Scores Based on Background Samples

Lior Wolf[1,⋆], Tal Hassner[2], and Yaniv Taigman[1,3]

[1] The School of Computer Science, Tel-Aviv University
[2] Computer Science Division, The Open University of Israel, Israel
[3] face.com

Abstract. Evaluating the similarity of images and their descriptors by employing discriminative learners has proven itself to be an effective face recognition paradigm. In this paper we show how "background samples", that is, examples which do not belong to any of the classes being learned, may provide a significant performance boost to such face recognition systems. In particular, we make the following contributions. First, we define and evaluate the "Two-Shot Similarity" (TSS) score as an extension to the recently proposed "One-Shot Similarity" (OSS) measure. Both these measures utilize background samples to facilitate better recognition rates. Second, we examine the ranking of images most similar to a query image and employ these as a descriptor for that image. Finally, we provide results underscoring the importance of proper face alignment in automatic face recognition systems. These contributions in concert allow us to obtain a success rate of 86.83% on the Labeled Faces in the Wild (LFW) benchmark, outperforming current state-of-the-art results.

1 Introduction

In a learning framework, we define *background samples* as samples that do not belong to the classes being learned. Collecting such samples is often easy as they do not require labeling. For example, in a face identification scenario, these samples could be a face set of individuals not among those which the system is being trained to recognize. Besides being easy to collect, we believe such examples may provide valuable information about which images may be considered "the same" and which may not. In this paper we present similarity measures designed to exploit such background samples. These measures are then shown to outperform state-of-the-art techniques on standard image similarity tests.

Why would background samples be useful for defining similarity functions? The sample vectors are embedded in a vector space in which various metrics can be employed. In order to know which metric is most suitable for the similarity task at hand, the underlying structure of the manifold on which the samples reside needs to be analyzed. Supervised learning can sometimes be used, but may require extra labeling information. On the other hand, background samples without additional information directly answer questions such as "is this sample

⋆ Lior Wolf is supported by the Israel Science Foundation (grants No. 1440/06, 1214/06), the Colton Foundation, and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 88–97, 2010.

closer to that one than to a typical example from the background set?" (One-Shot); "are these two examples well separated from the background sample set?" (Two-shot); and "do these two samples have similar sets of neighboring samples in the background set?" (ranking similarity).

As a benchmark for testing our methods, we use the Labeled Faces in the Wild (LFW) database [1]. It offers a unique collection of annotated faces captured from news articles on the web. It can be used to estimate face recognition performance on faces detected automatically in web images, and may serve as a reasonable benchmark for photo album applications. The dataset is published with a specific benchmark, which focuses on the face recognition task of *pair matching*. In this task, given two face images, the goal is to decide whether the two pictures are of the same individual. This is a binary classification problem, with two possible outcomes: "same" or "not-same".

The best results currently reported on the LFW benchmark were obtained by [2] using the One-Shot Similarity measure. Our tests here on the same benchmark indicate that exploiting background samples yields improved performance.

The rest of the paper is organized as follows. Section 2 reviews related work. In Section 3 we describe the novel Two-Shot similarity measure and its efficient computation. Using image ranking as an additional image descriptor is proposed in Section 4. Section 5 discusses the importance of accurate face alignment for recognition. We present our results in Section 6 and finally conclude in Section 7.

2 Related Work

The literature on similarity functions is extensive. Some similarity measures proposed in the past have been hand crafted (e.g., [3,4]). Alternatively, a growing number of authors have proposed tailoring similarity measures to available training data by applying learning techniques (e.g., [5,6,7,8,9]). In all these methods testing is performed using models (or similarity measures) learned beforehand.

Recently [10,11], the One-Shot Similarity (OSS) score was introduced as an alternative approach which utilizes background samples. The OSS draws its motivation from the growing number of so called "One-Shot Learning" techniques; that is, methods which learn from one or few training examples (see for example [12,13]). Unlike previous methods for computing similarities, the OSS score of two signals is computed by training a discriminative model *exclusive* to the two signals being compared, by using a set of background samples. It was consequently shown to be instrumental in obtaining state-of-the-art results on the Labeled Faces in the Wild (LFW) image pair-matching challenge [1].

Employing background samples differs from semi-supervised learning [14] and from transductive learning [15] since in both cases the unlabeled samples belong to the set of training classes. It differs from flavors of transfer learning that use unlabeled samples [16], since they use separate supervised learning tasks in order to benefit from the unlabeled set.

Although learning with background samples can be seen as belonging to the group of techniques called "learning with side-information", it differs from

existing methods in the literature known to us. In particular, some of the previous contributions, e.g., [17,18,19], require having training samples with the same identity. Other side-information contributions, e.g., [20] assume that the variability in the side information differs from that in the relevant data.

3 The Two-Shot Similarity Score

We begin our description of the TSS measure by reviewing the OSS [10,11]. Given two vectors **I** and **J** their OSS score is computed by considering a training set of background sample vectors **A**. This set contains examples of items not belonging in the same class as neither **I** nor **J**, but are otherwise unlabeled. A measure of the similarity of **I** and **J** is then obtained as follows. First, a discriminative model is learned with **I** as a single positive example, and **A** as a set of negative examples. This model is then used to classify the vector, **J**, and obtain a confidence score. The nature of this score depends on the classifier used. Using linear SVM, for example, this score may be the signed distance of **J** from the hyperplane separating **I** and **A**. A second such score is then obtained by repeating the same process with the roles of **I** and **J** switched. The final OSS score is the average of these two scores.

The Two-Shot similarity score is obtained in a single step by modifying the process described above. Again, we consider the same auxiliary set of negative examples **A**. This time, however, we train a single discriminative model using both **I** and **J** as positive examples, and the set **A** as a set of negative examples. The Two-Shot score is then defined as a measure of how well this model discriminates the two sets. Again, the particular definition of this score depends on the underlying classifier used. Using the SVM classifier, for example, this can simply be the width of the margin between the two sets. In the following sections we provide detailed analysis of this new similarity score.

3.1 Background-Sample Based Similarities with LDA

The OSS and TSS scores are actually meta-similarities which can be fitted to work with almost any discriminative learning algorithm. In our experiments, we focused on the Fisher Discriminant Analysis (FDA or LDA) [21,22] as the underlying classifier. Similarities based on LDA can be efficiently computed by exploiting the fact that the set **A** of negative samples is used repeatedly, and that the positive class, which contains just one or two elements, contributes either nothing or a rank-one matrix to the within class covariance matrix.

We focus on binary LDA, which is relevant to this work. Let $p_i \in \mathbb{R}^d, i = 1, 2, ..., m_1$ be a set of positive training examples, and let $n_i \in \mathbb{R}^d, i = 1, 2, ..., m_2$ be a set of negative training examples. Let μ be the average of all points and μ_p (resp. μ_n) be the average of the positive (negative) training set. Two matrices are then considered [23], S_B measuring the covariance of the class centers, and S_W, which is the sum of the covariance matrices of each class. The LDA algorithm computes a projection v which maximizes the quotient:

$$v = \arg\max_{v} \frac{v^\top S_B v}{v^\top S_W v} \tag{1}$$

In the two class case, v is easily determined as:

$$v = \frac{S_W^+(\mu_p - \mu_n)}{\|S_W^+(\mu_p - \mu_n)\|} \tag{2}$$

Note that we use the pseudo-inverse S_W^+ instead of the inverse S_W^{-1} in order to deal with cases where the within-class covariance matrix is not full rank. This is equivalent to requiring in Eq. 1 that v be spanned by the training vectors.

Once v has been computed, the classification of a new sample $x \in \mathbb{R}^d$ is given by the sign of $v^\top x - v_0$, where v_0 is the bias term (see below).

LDA-based One-Shot Similarity. The LDA-based OSS score and its computation was recently analyzed in [11]. By exploiting the fact that the positive set contains a single sample and the negative set is fixed, it was shown that the LDA-based OSS between samples **I** and **J** given the auxiliary set **A** becomes:

$$\frac{(I-\mu_A)^\top S_W^+ (J - \frac{I+\mu_A}{2})}{\|S_W^+(I-\mu_A)\|} + \frac{(J-\mu_A)^\top S_W^+ (I - \frac{J+\mu_A}{2})}{\|S_W^+(J-\mu_A)\|} \tag{3}$$

The overall complexity for the OSS per pair is thus $O(d^2)$ once the (pseudo) inverse S_W has been computed. In addition, if similarities are computed for the same point repeatedly, one can factor the positive definite $S_W^+ = HH^\top$ and pre-multiply this point by the factor H.

LDA-based Two-Shot Similarity. In the two-shot case, **I** and **J** serve as the positive class, while the set **A** of background samples is used repeatedly as the negative class. In contrast to the One-Shot case, the within class covariance matrix S_W changes from one similarity computation to another.

In order to be robust to the size of the background set and for simplicity, we balance the positive and the negative classes and define the within-class convenience matrix as $S_W = \frac{1}{2}S_A + \frac{1}{2}S_{IJ}$, where $S_A = \frac{1}{|A|}\sum_{x\in A}(x-\mu_A)(x-\mu_A)^\top$, and $S_{IJ} = \frac{1}{2}((I - \frac{(I+J)}{2})(I - \frac{(I+J)}{2})^\top + (J - \frac{(I+J)}{2})(J - \frac{(I+J)}{2})^\top) = \frac{1}{4}(I-J)(I-J)^\top$

Since S_{IJ} is a rank-one matrix, the inverse of S_W can be computed by updating the inverse of S_A with accordance to the Sherman-Morrison formula as:

$$\frac{1}{2}S_W^{-1} = S_A^{-1} - \frac{S_A^{-1}(I-J)(I-J)^\top S_A^{-1}}{4 + (I-J)^\top S_A^{-1}(I-J)} \tag{4}$$

If S_W is not full rank, a similar formula can be applied to update the pseudoinverse, based on rank-one updates [24] of the Cholesky factor or SVD of S_A. The details are omitted. Note that the matrix S_W^{-1} need not be computed explicitly. Let $\nu = (I+J)/2 - \mu_A$. From equation 2, v can be computed up to scale as:

$$S_A^{-1}\nu - \frac{S_A^{-1}(I-J)(I-J)^\top (S_A^{-1}\nu)}{4 + (I-J)^\top S_A^{-1}(I-J)}$$

The TSS itself measures the separability of the two classes, i.e., the distance between the centers of the two classes in the direction of v. Thus, once the covariance matrix of the background samples is inverted, computing the TSS requires $O(d^2)$ operations. If points I_i are used repeatedly, $S_A^{-1} I_i$ can be precomputed, and future TSS computations become $O(d)$.

4 Ranking Based Background Similarity

The idea of representing an image by a set of similarities to other images or to prelearned classifiers is well known [25]. Bart and Ullman [26] have proposed to use it for learning a novel class from one example. We have tried using a vector of similarities to the background samples as a face descriptor. Specifically, we generated for image I and for image J vectors of similarities by comparing I or J to each image in A. The resulting vectors produce much worse classification results than the original similarity between I and J.

Instead, we consider a retrieval system in which images I or J are used to retrieve similar images from the set A, and examine the order in which the images are retrieved. In other words, image I (or J) produces an order on the elements of A from the most similar to the least similar.

To compare two such orders, we can employ the non-parametric technique of computing the correlation between the rank vectors. Each image (I or J) is represented by a vector which contains the ranking of each image in the set A from 1 (most similar image) to $|A|$ (least similar image). The correlation between the two rank vectors measures the similarity between the two permutations.

In our experiments, we have found that it is better to focus on the most similar images. We propose the following statistical test. For each of the two samples I and J we compute the rank vectors r_I and r_J as before. Let π_I (π_J) be the order of images in A according to their similarity to I (J). We then compute the similarity s as the sum of the ranking by one image to the first 100 images in the order of the second image: $s(I, J) = -\sum_{k=1}^{100} r_I(\pi_J(k)) + r_J(\pi_I(k))$. (higher values mean more similar examples). We are yet to conduct a full test for the value of the parameter; currently 100 seems to produce good results.

5 Face Alignment

In order to produce an aligned version of the LFW images, we automatically processed them using a commercial face alignment system provided by face.com. The alignment system is based on localization of fiducial points. An affine transformation that brings those feature points to fixed locations is applied to the image. In order to train the feature detectors face.com have collected a set of labeled face images with manually marked points. These images do not intersect the LFW set in images or in identity.

Our experiments, reported in Section 6, show that this alignment method significantly improves the performance of all tested methods. To illustrate the importance of this improved alignment, we tested the performance of the system

Fig. 1. Two images aligned using the Funneling technique of [28] (on the left) and the face.com alignment system. Misalignments on the left hand pair are visible when comparing the positions of the mouth and the eyes to the markers. These misalignments are all but removed in the right hand pair.

designed by Nowak and Jurie [27] on our own aligned version of the LFW image set. Originally, the LFW images were aligned using the "Funneling" technique of [28]. On this funneled set, Nowak and Jurie obtained a recognition rate of 0.7393. The same method obtains a recognition score of 0.7912 on our aligned set. Note that this performance boost was gained even tough the method of [27] has build-in mechanisms to deal with misalignments.

Figure 1 presents an example of one image pair aligned using both the original Funneling technique of [28] and our own alignment method. The improved alignment of both the eyes and the mouth is evident by comparing their positions to the markers. It is important to note that while the Funneling technique requires no additional training (it is an unsupervised technique), feature-point based alignment techniques, including the method employed by the authors of [29], rely on the existence of a training set of images with marked fiducial points.

6 Experiments

We test the effect of the various contributions on the 10 folds of view 2 of the LFW dataset [1]. Similarly to previous contributions, we employ "image-restricted training". This benchmark consists of 6,000 pairs, half marked "same" and half not, and is divided into 10 equally sized sets. The benchmark test is repeated 10 times, each time using one set for testing and nine others for training. The goal is to predict which of the test pairs match using only the training data.

We used one of the nine training splits for the background set A and the other eight for classifier training. The background split contains 1,200 images. The subjects in these images do not appear in the test set, as the LFW benchmark is constructed to have subjects in the different splits mutually exclusive [1].

6.1 The Contribution of Alignment

Our first set of experiments repeats the experiments of [10] while introducing the automatic alignment. Note that we did not make any attempt to verify the alignment. If the alignment fails for any reason, we still use the resulting image.

The results are described in Table 2. We use the same descriptors of [10] with the addition of a SIFT descriptor: the LBP descriptor [30], two variants called Three-patch and Four-patch LBP (TPLBP and FPLBP) [10], the C1 image descriptor [31], and SIFT [32]. The parameters of all descriptors were copied from [10]. To compute the SIFT descriptor, we subdivide the image into a grid of 7x7, and compute a 128D SIFT descriptor for each one of the 49 patches.

Table 1. Mean (± standard error) scores on the LFW, Image-Restricted Training benchmark ("view 2") using Euclidean similarities

	Original images		Funneled		Alignment	
Image Descriptor	Euclidian	SQRT	Euclidian	SQRT	Euclidian	SQRT
LBP	0.6649	0.6616	0.6767	0.6782	0.6824	0.6790
Gabor (C1)	0.6665	0.6654	0.6293	0.6287	0.6849	0.6841
TPLBP	0.6713	0.6678	0.6875	0.6890	0.6926	0.6897
FPLBP	0.6627	0.6572	0.6865	0.6820	0.6818	0.6746
Above combined	0.7107 ± 0.0045		0.7062 ± 0.0046		0.7450 ± 0.0068	
SIFT	0.6617	0.6672	0.6795	0.6870	0.6912	0.6986
All combined	0.7223 ± 0.0092		0.7193 ± 0.0049		0.7521 ± 0.0055	

All descriptors are then concatenated to a single vector. Compared to the LBP variants, the SIFT descriptor is less sensitive to misalignment, however, it is easily misled by sharp edges caused by glasses or illumination.

We use either the descriptor vectors or their square roots (i.e., the Hellinger distance). In the latter case, instead of using the descriptor vector $g(I)$ we use $\sqrt{g(I)}$. The 10 descriptor/mode scores in the table are obtained by training SVM on $4,800$ (8 sets) 1D vectors containing the similarity scores. The "Combined" classification is based on learning and classifying the 8D/10D vectors which are the concatenations of the eight/ten 1D vectors (including or excluding SIFT). The results are reported in Table 2. The contributions of adding SIFT and of performing a proper alignment are clearly seen.

6.2 The Contribution of One-Shot

Next, we examine the performance on the one-shot measure in Table 2. The descriptors used are the same as above. Here again we use either the original descriptor vectors, or their square roots. The "Combined" classification is based on learning and classifying the 8D/10D vectors which are the concatenations of the eight/ten 1D One-Shot similarites. Results are reported without SIFT (to allow comparison to [10]) and with SIFT. The "Hybrid" results contain all direct (Euclidean) similarities above and the One-Shot similarities. Note the gap in performance compared to the funneled no-sift hybrid previously reported.

6.3 The Contribution of Two-Shot

The two-shot similarity adds another layer of information. By itself, it is not very discriminative. For the aligned images, all 10 (5 descriptors and using or not using square root) two-shot similarities provide a combined score of 0.6593 ± 0.0076, which is lower than the corresponding figure of 0.8207 for the One-Shot Similarities and the 0.7521 for the baseline similarities.

However, in combination with the baseline similarities and the One-Shot Similarities, the Two-Shot Similarities boost performance considerably. Adding those similarities to the mix increases the performance in the aligned images from 0.8398 ± 0.0035 to 0.8513 ± 0.0037.

Table 2. Mean (± standard error) scores on the LFW, Image-Restricted Training benchmark ("view 2") using OSS

	Original images		Funneled		Alignment	
Image Descriptor	OSS	OSS SQRT	OSS	OSS SQRT	OSS	OSS SQRT
LBP	0.7292	0.7390	0.7343	0.7463	0.7663	0.7820
Gabor (C1)	0.7066	0.7097	0.7112	0.7157	0.7396	0.7437
TPLBP	0.7099	0.7164	0.7163	0.7226	0.7453	0.7514
FPLBP	0.7092	0.7112	0.7175	0.7145	0.7466	0.7436
Above OSS Comb.	0.7582± 0.0067		0.7653 ± 0.0054		0.8002 ± 0.0018	
Above Hybrid	0.7752 ± 0.0063		0.7847 ± 0.0051		0.8255 ± 0.0031	
SIFT	0.7126	0.7199	0.7202	0.7257	0.7576	0.7597
All OSS Combined	0.7673 ± 0.0039		0.7779 ± 0.0072		0.8207 ± 0.0041	
All Hybrid	0.7782 ± 0.0036		0.7895 ± 0.0053		0.8398 ± 0.0035	

6.4 The Contribution of the Ranking Descriptor

The ranking based similarities obtained by the proposed score, which considers the ranking by one example of the first 100 images closest to the other example. It is slightly more effective than Two-Shot Similarity above, and the score obtained by combining all 10 rank similarities using SVM is 0.6918 ± 0.0062. As mentioned in Sec. 4, using other forms of representation by similarity are not better.

Similar to the the Two-Shot Similarity above, the contribution of the ranking descriptor by adding it to the other descriptors. A hybrid descriptor which contains 10 original distances, 10 One-Shot distances, 10 Two-Shot distances, and 10 ranking based distances produces a result of 0.8557 ± 0.0048, which is much higher than the current record of 0.7935 ± 0.0055 [33].

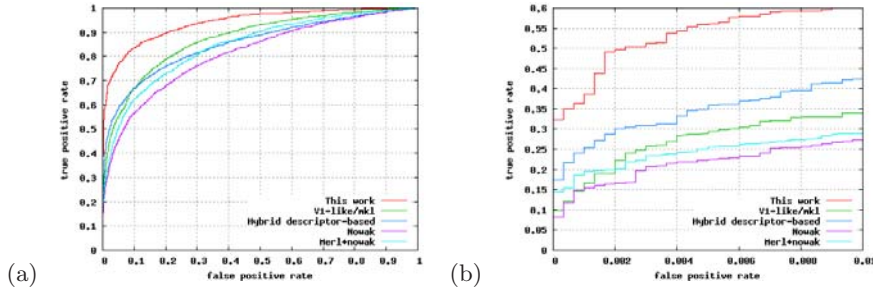


Fig. 2. ROC curves for View 2 of the LFW data set. Each point on the curve represents the average over the 10 folds of (false positive rate, true positive rate) for a fixed threshold. (a) Full ROC curve. (b) A zoom-in onto the low false positive region. The proposed method is compared to scores reported in http://vis-www.cs.umass.edu/lfw/results.html. These methods include the combined nowak+Merl system [29], the Nowak method [27], the hybrid method of [10], and the recent V1-like/mkl method of [33].

6.5 Combining Background Similarities beyond LDA

The One-Shot and Two-Shot similarities are frameworks that can be applied with LDA as above or with other classifiers. Applying it with SVM instead of LDA gives very similar results. However, a considerable boost in performance is obtained when adding SVM based OSS and TSS to those of LDA. Adding those 20 additional dimensions results in a performance of 0.8297 ± 0.0037 for the funneled images and 0.8683 ± 0.0034 for the aligned images.

The ROC curves of the final combined result, as well as the results of previous work is presented in Figure 2. As can be seen, the present result is considerably better than previous method. This is especially so in the low-false-positive region, which is the crucial region for most applications.

7 Conclusions

We follow the Detection-Alignment-Recognition pipeline devised in [1] for the study of face recognition in unconstrained environments. For alignment, we demonstrate the significance of proper localization by improving upon results obtained on already aligned images. For representation, we augment the set of descriptors by adding SIFT. For similarity we study three frameworks for employing background samples, shifting focus from one-shot to two examples to many examples. This form of side information has not gained considerable attention previously, and we demonstrate its effectiveness. The obtained leap in performance is impressive given the law of diminishing returns and the amount of work invested by various groups on the LFW benchmark.

Acknowledgments

We thank Michal Irani and Greg Shakhnarovich for discussion that led to this work.

References

1. Huang, G., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. UMASS, TR 07-49 (2007)
2. Taigman, Y., Wolf, L., Hassner, T.: Multiple one-shots for utilizing class label information. In: BMVC (2009)
3. Belongie, S., Malik, J., Puzicha, J.: Shape context: A new descriptor for shape matching and object recognition. In: NIPS (2001)
4. Zhang, H., Berg, A., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)
5. Bilenko, M., Basu, S., Mooney, R.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML (2004)
6. Cristianini, N., Kandola, J., Elisseeff, A., Shawe-Taylor, J.: On kernel-target alignment. In: NIPS (2002)
7. Shental, N., Hertz, T., Weinshall, D., Pavel, M.: Adjustment learning and relevant component analysis. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 776–790. Springer, Heidelberg (2002)

8. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2006)
9. Xing, E., Ng, A.Y., Jordan, M., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS (2003)
10. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Faces in Real-Life Images Workshop in ECCV (2008)
11. Wolf, L., Hassner, T., Taigman, Y.: The one-shot similarity kernel. In: ICCV (2009)
12. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. PAMI 28(4), 594–611 (2006)
13. Fink, M.: Object classification from a single example utilizing class relevance pseudo-metrics. In: NIPS (2004)
14. Chapelle, O., Scholkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
15. Joachims, T.: Transductive learning via spectral graph partitioning. In: International Conference on Machine Learning (ICML), pp. 290–297 (2003)
16. Quattoni, A., Collins, M., Darrell, T.: Transfer learning for image classification with sparse prototype representations. In: CVPR (June 2008)
17. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: NIPS, Cambridge, MA (2003)
18. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: ICML (2003)
19. Liu, W., Hoi, S., Liu, J.: Output regularized metric learning with side information. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 358–371. Springer, Heidelberg (2008)
20. Chechik, G., Tishby, N.: Extracting relevant structures with side information. In: NIPS, pp. 857–864 (2002)
21. Fisher, R.: The use of multiple measurements in taxonomic problems. Annals Eugenics 7, 179–188 (1936)
22. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Springer, Heidelberg (2001)
23. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley, Chichester (2001)
24. Brand, M.: Fast low-rank modifications of the thin singular value decomposition. Linear Algebra and its Applications 415(1), 20–30 (2006)
25. Edelman, S.: Representation and recognition in vision. MIT Press, Cambridge (1999)
26. Bart, E., Ullman, S.: Single-example learning of novel classes using representation by similarity. In: British Machine Vision Conference (2005)
27. Nowak, E., Jurie, F.: Learning visual similarity measures for comparing never seen objects. In: CVPR (June 2007)
28. Huang, G., Jain, V., Learned-Miller, E.: Unsupervised joint alignment of complex images. In: IEEE International Conference on Computer Vision (2007)
29. Huang, G., Jones, M., Learned-Miller, E.: Lfw results using a combined nowak plus merl recognizer. In: Faces in Real-Life Images Workshop in ECCV (2008)
30. Ojala, T., Pietikainen, M., Harwood, D.: A comparative-study of texture measures with classification based on feature distributions. Pattern Recognition 29(1) (1996)
31. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience 2(11), 1019–1025 (1999)
32. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
33. Pinto, N., DiCarlo, J., Cox, D.: How far can you get with a modern face recognition test set using only simple features? In: CVPR (2009)

Human Action Recognition Using Spatio-temporal Classification

Chin-Hsien Fang, Ju-Chin Chen, Chien-Chung Tseng, and Jenn-Jier James Lien

Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan 70101, R.O.C.
{func1115,joan,ed,jjlien*}@csie.ncku.edu.tw

Abstract. In this paper a framework "Temporal-Vector Trajectory Learning" (TVTL) for human action recognition is proposed. In this framework, the major concept is that we would like to add the temporal information into the action recognition process. Base on this purpose, there are three kinds of temporal information, LTM, DTM, and TTM, being proposed. With the three kinds of proposed temporal information, the k-NN classifier based on the Mahanalobis distance metric do have better results than just using spatial information. The experimental results demonstrate that the method can recognize the actions well. Especially with our TTM and DTM framework, they do have great accuracy rates. Even with noisy data, the framework still have good performance.

1 Introduction

Human action recognition has been an active issue over the last decades in computer vision community, which has created a wide range of applications, such as video surveillance, human-computer interaction, analysis of sports events, etc. Several situations would cause the action recognition process to be challenging, including non-stationary backgrounds in videos, ambiguity of human body shapes between different actions, and intra-class variations of appearance, physical characteristics, motion style and motion temple of different human subjects.

One of the most important questions in human action recognition is how to extract features from the video sequences, and in [12] [25], the authors summarized the methods had been proposed, including: the computation of optical flow [8], space-time gradients [19] [29], feature tracking models [3] [5] [22] [27], sparse spatio-temporal interest points [7] [13] [14] [15]. However, recognition based on space-time gradient features or feature tracking models have the limitations in case of low quality videos or large variability in the articulation of human body and lighting conditions, respectively. On the other hand, the recognition based on the usage of the sparse representations of interest points would be limited due to the discard of global structural information [27].

Alternatively, the feature of human silhouettes are becoming popular in recent studies for human action recognition [4] [10] [12] [23] [24] [25], which is easier to obtain and the silhouettes can still contain the detailed body shape information. Moreover, a sequence of human silhouettes generate space-time shapes which not only

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 98–109, 2010.

encode spatial information of the body shape but also temporal motion information of global body and local body parts [25]. A human silhouette of each frame (image) can be represented by a vector in high-dimensional space, and thus the human silhouettes collected from human action movements are expected intrinsically lie in a low-dimensional space embedded in the high-dimensional space [12]. Manifold learning methods, e.g. isometric feature map (Isomap) [20], locally linear embedding (LLE) [18], Laplacian eigenmap [2], can discover the intrinsic geometrical structure of data and thus the human action motion can be analyzed in a compact low-dimensional space. Accordingly, Elgammal and Lee [9] applied LLE to learn the view-based representations for walking manifolds in order to recover intrinsic body configurations. Wang [25] found a low-dimensional feature representation for human silhouettes via locality preserving projections (LPP) [11] which is a linear approximation to LE [2].

Recently the supervised manifold learning methods are proposed that they take the class label information into consideration, including marginal Fisher analysis (MFA) [28], supervised-LPP [25], locality sensitive discriminant analysis (LSDA) [6], etc. By regarding the class label information, the local spatial discriminant structure can be discovered and thus the images with different action classes can be separated. However, not only spatial information, i.e. body shape of silhouette, the temporal motion information inhered from the video sequences, i.e. dynamic shape variation of human silhouette sequences, can provide important information for recognition. Various studies incorporate the temporal information from different aspects. In [12], Jia et. al. propose a spatio-temporal subspace learning method (LSTDE) in which temporal subspaces associated with data points of consecutive frames are constructed. The proposed subspace are constrained by not only maximizing the discriminant structure according to class label but also maximizing the principal angles between those temporal subspaces of different classes. On the other hand, in [25], temporal evolution of an action motion is viewed as a sequence of projection points with temporal orders, and state-space models, i.e. HMM are applied to capture the structural and dynamical nature of human movements.

Motivated by the above studies, we propose a silhouette-based human action recognition system that not only considers the spatial information in the spatial-motion subspace, but also include the motion trajectory in the temporal motion space. In the system, the features of human silhouettes, which contain the detailed body shape information, are extracted from motion videos. First, the subspace method is applied to obtain a spatial-motion subspace that the human body shapes from the collected human silhouettes can be analyzed. However, in order to cope with the ambiguity of human body shapes between different action types which caused overlap in the subspace, the temporal-vectors are proposed to characterize the trajectory motion information in the subspace. Moreover, the Mahalanobis distance metric is based on the goal that those k-nearest temporal-vectors belongs to the same action class should gather together while the ones with different action class can be separated by a margin. Benefit from the temporal-vector trajectory learning, our system can be performed on frame-based accuracy measurement.

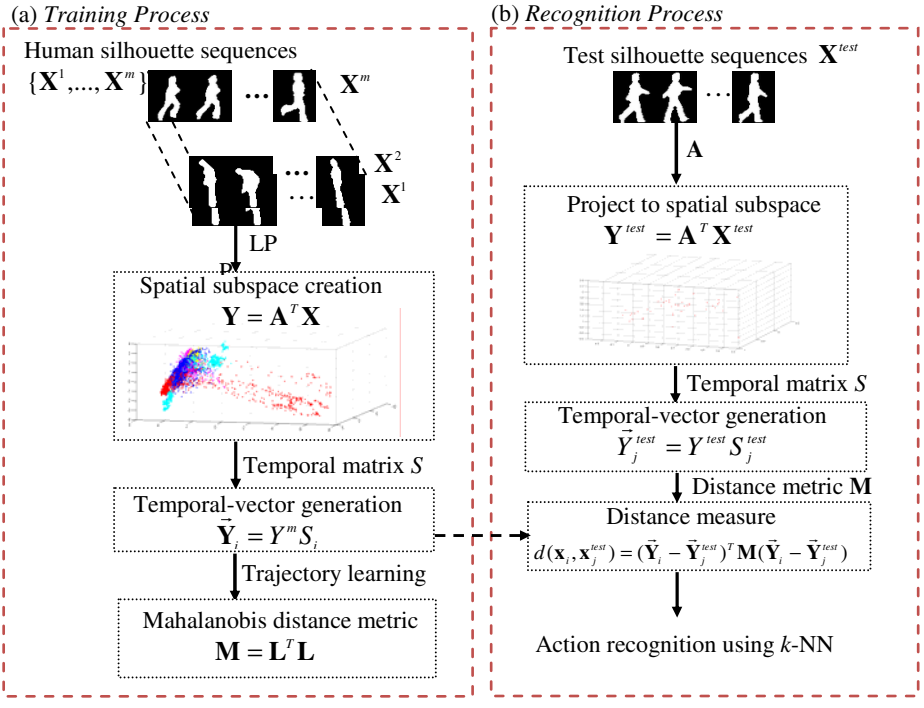


Fig. 1. Flowchart of the proposed human action recognition based on temporal-vector trajectory learning

2 Learning for Spatio-temporal Classification

Fig 1 shows the proposed action recognition system based on temporal-vector trajectory learning. Assume there are m video sequences, each of which has n_i training frames. The training process commences by extracting human silhouettes $\{\mathbf{X}^i\}_{i=1}^m$ as feature representation for each human action video. Human silhouettes can be obtained via the background subtraction method. In order to reduce the intra-class variations of different subject's sizes, the normalize process is applied to centralize and resize the human silhouettes so that the resized human silhouettes ($w \times h$) can contain the body shape information as much as possible. Each silhouette $\mathbf{x}_i$ can be represented by a D-dimensional vector $\mathbf{x}_i \in R^{D=w \times h}$ and thus we have the whole training set with all resized silhouettes $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$, where $N = n_1 + n_2 + ... + n_m$.

As shown in Fig. 1, the proposed system mainly consists of two steps to analyze the spatial and temporal motion of human silhouettes, respectively. In Section 2.1, we first find a spatial motion subspace in which we can capture the local spatial motion information of high-dimensional silhouettes. Then, in order to incorporate the temporal information from the video sequences to enhance the recognition performance, in Section 2.2, three kinds of temporal-vectors are proposed to extract various kinds of

temporal information and moreover the Mahalanobis distance metric is learned to make those data which with similar local spatial information but different trajectory to be well separated.

2.1 Spatial Subspace Creation Using Locality Preserving Projection

Human action video sequences represented by the collection of human silhouettes can be viewed as the data points on nonlinear manifolds. The first step, we would like to obtain a low-dimensional space to discover the intrinsically nonlinear structure of spatial-motion information where the local spatial information can be preserved, i.e. the points that are close(similar) in high-dimensional space would be also close in the low-dimensional motion subspace. Here we choose the well-known method Locality Preserving Projection (LPP) [11] because it can provide an optimal linear transformation as the approximation to nonlinear spectral embedding techniques (i.e. Laplacian Eigenmap [2]). Thus, via the linear transformation method, LPP can provide us the transformation matrix for new data or test data.

In LPP, in order to construct the adjacency undirected graph for **X**, we first apply *k*-nearest neighbors methods to determine neighbors for each data and the adjacency matrix $\mathbf{W} \in R^{N \times N}$ is obtained in which stores the pairwise relation of data points. Here, we can simply assign 0 or 1 to the adjacency matrix's component. If $\mathbf{x}_i$ is the neighbor of $\mathbf{x}_j$ or $\mathbf{x}_j$ is the neighbor of $\mathbf{x}_i$, then we assign 1 to W_{ij}, otherwise we assign 0. Note that the heat kernel function can be also applied to assign values to the adjacency matrix components.

According to the locality-preserving criterion, we assume we have the transformation vector **a** that it has to minimize the objective function (1). For mapping X to a line, that is we would like to project X to a 1-D space, then we get $\mathbf{y}^{\mathbf{T}} = \mathbf{a}^{\mathbf{T}}\mathbf{X}$, where $\mathbf{y} = (y_1, y_2 .....y_N)^T$.

$$\arg\min_{\mathbf{a}} \sum_{ij} (y_i - y_j)^2 W_{ij} = \arg\min_{\mathbf{a}} \sum_{ij} (\mathbf{a}^{\mathbf{T}}\mathbf{x}_i - \mathbf{a}^{\mathbf{T}}\mathbf{x}_j)^2 W_{ij} . \tag{1}$$

Thus, we attempt to make sure that in low-dimensional embedding y_i and y_j should be close if $\mathbf{x_i}$ and $\mathbf{x_j}$ are close in the original high-dimensional space. Via algebraic steps [28], the above objective function can be reformulated as:

$$\frac{1}{2}\sum_{ij} (\mathbf{a}^{\mathbf{T}}\mathbf{x}_i - \mathbf{a}^{\mathbf{T}}\mathbf{x}_j)^2 W_{ij} = \mathbf{a}^{\mathbf{T}}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathbf{T}}\mathbf{a} . \tag{2}$$

L is the Laplacian matrix and **L=D-W,** where **D** is a diagonal matrix and $D_{ii} = \sum_j W_{ij}$. The value of the element D_{ii} is an importance measurement for each data point $\mathbf{x}_i$. That is the more neighbors $\mathbf{x}_i$ has, the more importance it gets, i.e. the larger value D_{ii} has. According to [11], the constraint is further imposed

$$\mathbf{a}^{\mathbf{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathbf{T}}\mathbf{a} = 1 . \tag{3}$$

According to Eq. (2) and (3), the optimization problem becomes

$$\arg\min_{\mathbf{a}} \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\mathbf{a} \quad \text{s.t. } \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{a} = 1. \tag{4}$$

Eq. (4) can be solved as the generalized eigenvalue problem. Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d]$ be the solution set of Eq. (4). And the transformation matrix $\mathbf{A} \in R^{D \times d}$ corresponds to the eigenvectors with the d smallest eigenvalues. Thus, the low-dimensional embedding $\{\mathbf{y}_i\}_{i=1}^{N} \in R^d$ of the original data can be obtained by

$$\mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i \tag{5}$$

Hence, the matrix $\mathbf{Y} = \mathbf{A}^{\mathrm{T}}\mathbf{X}$ is a $d \times N$ matrix, which contains the low-dimensional embedding of each data $\mathbf{x}_i$ in its columns, i.e. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]$.

2.2 Learning for Classification in Temporal Subspace

After obtaining the spatial-motion subspace, all training silhouettes can be represented as the low-dimensional embedding. However, ambiguity of human body shapes between different action types will cause overlap in the motion subspace, as shown in Fig. 1. In order to separate these data, we take temporal information into consideration, which is inherent in the video sequences.

Inspired from the kernel methods that transform the input data which are not linearly separable to higher or infinite-dimensional feature space via kernel function $\phi(\bullet)$, we design three kinds of temporal matrices i.e. S_i^1, S_i^2, and S_i^3 for each embedding $\mathbf{y}_i$ in order to incorporate different temporal information, respectively. Each S_i^p (p=1~3) is defined as a $N \times (2t+1)$ matrix. And t is a parameter that it decide how long the temporal information we would like to take into account, i.e. if $t = 2$ it means that base on data $\mathbf{y}_i$ we want to use those data that is in the same sequence and is the top four closet data temporally(from t = -2 ~ t = 2) to be the additional information. Assume $\mathbf{y}_i$ is the low-dimensional embedding of the frame $\mathbf{x}_i$, which corresponds to the r^{th} frame of sequence m, here we denote it as Y_r^m for convenience. For each $\mathbf{y}_i$, let $F_i = \{f_1, f_2, ..., f_{2t+1}\} = \{r-t, ..., r, ..., r+t\}$ denotes the set of frame number of corresponding frame number of each $\mathbf{y}_i$.

The first temporal segment can be defined as

$$(S_i)_{pq}^1 = \begin{cases} 1, & if \ p = f_q, \quad f_q \in F_i, \quad q = 1, ..., 2t+1 \\ 0, & otherwise \end{cases} \tag{6}$$

Multiply S_i^1 with Y^m, i.e. $Y_i' = Y^m S_i$, where Y^m contains the embeddings of all silhouettes in video sequence m, and thus Y_i' contains not only the original data but it includes his temporal neighbor as part of his information too.

$$Y_i' = [Y_{r-t}^m, ..., Y_r^m, ..., Y_{r+t}^m] \tag{7}$$

where Y_i' is a $1 \times [d \times (2t+1)]$ matrix. Through the temporal matrix S_i^1, the spatial embeddings of associated temporal frames can be obtained.

Different from S^1, in order to incorporate the motion difference as information, the second matrix is defined as

$$(S_i)^2_{pq} = \begin{cases} 1, & if \quad p = f_{t+1}, \quad q = 1,...,2t+1 \\ -1, & if \quad p = f_q, \quad f_q \in F_i, \quad q = 1,...,2t+1 \ \wedge q \neq t+1 \\ 0, & otherwise \end{cases} \tag{8}$$

Thus, we can obtain the motion difference as

$$Y_i' = [Y_r^m - Y_{r-t}^m,...,Y_r^m - Y_{r-1}^m,\ Y_r^m\ ,Y_r^m - Y_{r+1}^m,...,Y_r^m - Y_{r+t}^m] \tag{9}$$

Note that the spatial information of Y_r^m has to be included such that the motion difference is relative to this point Y_r^m. That is except the temporal information, still we have the original spatial information inside. Different from Eq. (9), the third matrix is designed by adding the motion trajectory as the temporal information, i.e.

$$(S_i)^3_{pq} = \begin{cases} 1, & if \quad p = f_q + 1 \quad q < t+1 \quad \vee \quad p = f_q \quad q > t+1 \quad , f_q \in F_i \\ -1, & if \quad p = f_q - 1 \quad q > t+1 \quad \vee \quad p = f_q \quad q < t+1 \quad , f_q \in F_i \\ 0, & otherwise \end{cases} \tag{10}$$

Similarly, all relatives distance from Y_r^m can be obtained

$$Y_i' = [Y_{r-(t-1)}^m - Y_{r-(t)}^m,...,Y_r^m - Y_{r-1}^m,\ Y_r^m\ ,Y_{r+1}^m - Y_r^m,...,Y_{r+t}^m - Y_{r+(t-1)}^m] \tag{11}$$

As Eq. (9), the motion trajectory temporal information still have the original spatial information inside. From Eq. (7), (9), and (11), different temporal information Y_i' can be selected by using the corresponding temporal matrix. Finally, by concatenating all elements in Y_i', the vector $\vec{\mathbf{Y}}_i \in R^{1*[(2t+1)\times d]}$ that contains temporal information can be obtained for the corresponding $\mathbf{y}_i$ in spatial-motion subspace.

Although the temporal information has been included in $\{\vec{\mathbf{Y}}_i\}_{i=1}^N$, it is inappropriate to compare the similarity between two temporal-vectors with Euclidean distance metric. It is because each temporal-vector $\{\vec{\mathbf{Y}}_i\}_{i=1}^N$ mixtures the information of different scales, i.e. the spatial information $\mathbf{y}_i$ itself and the vector differences with its relative frames as temporal information. Hence, we have to obtain a distance metric to measure the pairwise similarity between temporal vectors which has different information inside.

Unlike Euclidean distance, the Mahalanobis distance metric can provide us a proper way for similarity measurement because it takes the correlations of the data and the scale-invariant problem into account, thus it isn't dependent on the scale of data. In addition, motivated by [26], the Mahalanobis distance metric can be designed to achieve the goal that separate the temporal vectors in confused area and with different class apart i.e. the data belongs to different action class but have similar body shape information should be separated.

Given the temporal vectors $\{\vec{\mathbf{Y}}_i\}_{i=1}^N$ with the corresponding class label $l_i \in \{1,2,...,c\}$. The distance between $\vec{\mathbf{Y}}_i$ and $\vec{\mathbf{Y}}_j$ can be computed based on the Mahalanobis distance metric. The $\mathbf{M} \in R^{[(2t+1)\times d]\times[(2t+1)\times d]}$ as:

$$D(\vec{\mathbf{Y}}_i', \vec{\mathbf{Y}}_j') = (\vec{\mathbf{Y}}_i' - \vec{\mathbf{Y}}_j')^T \mathbf{M} (\vec{\mathbf{Y}}_i' - \vec{\mathbf{Y}}_j') = \left\| \mathbf{L}(\vec{\mathbf{Y}}_i' - \vec{\mathbf{Y}}_j') \right\|^2. \tag{12}$$

The distance metric $\mathbf{M}$ can be express in terms of the square matrix $\mathbf{M} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$, where $\mathbf{L}$ is represents a linear transformation. According the goal that we want to separate the temporal-vectors of different class, the Mahalanobis distance metric can be designed as in [26], i.e. the objective function can be defined as Eq. (13).

$$\begin{aligned} &\text{minimize} \quad \sum_{ij}\eta_{ij}(\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_j')^{\mathrm{T}}\mathbf{M}(\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_j') + \beta\sum_{ij}\eta_{ij}(1-p_{il})\delta_{ijl} \\ &\text{subject to} \quad \text{(i)} \ (\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_k')^{\mathrm{T}}\mathbf{M}(\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_k') - (\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_j')^{\mathrm{T}}\mathbf{M}(\vec{\mathbf{Y}}_i'-\vec{\mathbf{Y}}_j') \geq 1-\delta_{ijk} \\ &\qquad\qquad \text{(ii)} \ \delta_{ijk} \geq 0 \\ &\qquad\qquad \text{(iii)} \ \mathbf{M} \ \text{has to be positive semi - definite} \end{aligned} \tag{13}$$

where $\eta_{ij} \in \{0,1\}$ indicates whether $\vec{\mathbf{Y}}_{\mathbf{j}}$ is a target neighbor of $\vec{\mathbf{Y}}_{\mathbf{i}}$ with the same class, $p_{ik} \in \{0,1\}$ indicates whether $\vec{\mathbf{Y}}_i$ and $\vec{\mathbf{Y}}_{\mathbf{k}}$ have the same class label. δ_{ijk} is the slack variables. Note that the first term of the object function minimizes the distance of input $\vec{\mathbf{Y}}_i$ and its target neighbors with same class label. While the second term penalizes small distances between each input and all other inputs with different class label ($\vec{\mathbf{Y}}_i$ to $\vec{\mathbf{Y}}_{\mathbf{k}}$). The scalar β can tune the importance between two terms.

3 Recognition Process

Now after the training process as shown in Section 2, we obtain two transformations matrices to analyze the spatial- and temporal-information, i.e. the motion subspace transformation $\mathbf{A}$ and the distance matrix $\mathbf{M}$ for trajectory classification, respectively. The recognition process commences by obtaining the human silhouettes for the input test sequences as used for the training sequences. Then, via these two transformations matrices, the input test sequences can be analyzed and recognized.

Given silhouettes of the test sequence with *n* frames $\mathbf{X}^{test} = [\mathbf{x}_1^{test}, \mathbf{x}_2^{test}, ...\mathbf{x}_n^{test}] \in R^D$. Note that we assumed the silhouettes have been centralized and resized as the same in training process. Firstly, by projecting each silhouette $\{\mathbf{x}_i^{test}\}_{i=1}^n$ to the motion subspace, e.g. $\mathbf{y}_i^{test} = \mathbf{A}^T\mathbf{x}_i^{test}$, the embeddings, $\{\mathbf{y}_i^{test}\}_{i=1}^n \in R^d$, contained spatial information is obtained. Following that, in order to improve the recognition accuracy, different temporal information can be further extracted by using the matrix defined in Eq. (6), (8), or (10) and thus we can obtain the corresponding temporal-vectors $\{\vec{\mathbf{Y}}_i^{test}\}_{i=1}^n \in R^{(2t+1)\times d}$ for each $\mathbf{y}_i$. Note that the vector length of $\vec{\mathbf{Y}}_i^{test}$ should be identical as in the training process for unanimity.

According to temporal information, *k*-nearest neighborhood classifier is applied to assign a class label to every test sequence frame, Here, the *k*=6 is set as in [2]. For each test frame, the top six closet neighbors among training data are chosen and the label of test frame $l(\mathbf{x}_i^{test})$ is assigned by winner-takes-all rule. Note that the distance between each training $\mathbf{x}_i$ and test frame $\mathbf{x}_j^{test}$ is defined as

$$d(\mathbf{x}_i, \mathbf{x}_j^{test}) = (\vec{\mathbf{Y}}_i - \vec{\mathbf{Y}}_j^{test})^T\mathbf{M}(\vec{\mathbf{Y}}_i - \vec{\mathbf{Y}}_j^{test}). \tag{14}$$

where **M** is the Mahalanobis distance metric learned in Sec 2.2 for trajectory classification. Finally, the label of the sequence is determined by the majority of assigned labels of test frames.

4 Experimental Results

To evaluate the proposed system for action recognition, we here choose the dataset from [10] as our experimental data. The performance is evaluated in the cases of the usage of only spatial and the further incorporation with the three proposed temporal information, respectively. Moreover, the system is also applied to test the performance of noisy data.

4.1 Human Action Databases

We use the action dataset from [10]. And we do the preprocessing base on [12]. It consists of 10 action types performed by 9 persons. The 10 actions are bending (bend), jumping jack (jack), jumping-forward-on-two-legs (jump), jumping-in-place-on-two-legs (pjump), galloping-sideways (side), running (run), walking (walk), waving-one-hand (wave1), and wave-two-hads (wave2) [10]. Here, we use 90 sequences in our experiment. Note that we directly use the foreground masks from [10] for subsequence processing and the shadow removing is not in consideration. Then, we centralize, crop and resize the silhouettes to the 64×48 pixels. In order to compute overall unbiased estimate results, nine-fold cross validation are applied for recognition. In other words, each time we exclude all sequences from one certain person (e.g. 10 testing sequences) and then the remaining 80 sequences are used for training process. The results are averaged over the nine runs for the reported accuracy.

4.2 Recognition Results and Comparison

In order to investigate the role of spatio- and temporal information in our action recognition framework, the experiments are conducted by using various feature types. **Type I:** The features are extracted in the spatio-motion subspace, i.e. the testing data are only projected to the spatio-motion subspace and classified based on Euclidean distance metric, and the temporal information is discarded. We designate it as s*patial motion of Euclidean distance* (SE). **Type II:** The Mahalanobis metric are learned from [26] for classification in the motion subspace but the temporal information is still not considered, and it is designated as *spatial moti*on *of Mahalanobis distance* [26]. **Type III:** The temporal information described in Eq. (7) is included and the corresponding Mahalanobis metric is applied for classification. It is designated as *locations' temporal motion of Mahalanobis distance* (LTM). **Type IV** and **Type V** are *difference' temporal motion of Mahalanobis distance* (DTM) and *trajectory temporal motion of Mahalanobis distance* (TTM), respectively, as defined in Eq. (9) and (11). Note that the recognition accuracy reported here is in terms of the percentage of the correctly recognized frames among all test frames.

Table 1. Frame-based recognition accuracies by using different kinds of motion subspace, LPP, supervised LPP, and LSDA with t = 2

t=2	SE	SM	LTM	DTM	TTM
Unsupervised LPP	78.67	79.45	**84.91**	**89.45**	**88.56**
Supervised LPP	74.24	74.11	76.86	79.69	80.54
LSDA	77.44	79.86	84.41	86.99	87.45

Table 2. Frame-based recognition accuracies by using different kinds of motion subspace, LPP, supervised LPP, and LSDA with t = 3

t=3	SE	SM	LTM	DTM	TTM
Unsupervised LPP	78.67	80.00	**79.64**	**90.21**	**88.76**
Supervised LPP	74.24	74.31	79.77	82.23	81.10
LSDA	77.44	79.66	73.02	89.62	87.79

In addition, we apply different manifold-learning methods for obtaining the spatio-motion subspace, including unsupervised LPP, supervised LPP [25] and LSDA [6] for further comparison. The recognition rates of using various combinations of motion subspace and feature types are summarized in Table 1. Here the parameter $k = 6$ is chosen for the number of nearest neighbors in the graph and the dimensionality are 31, 9, 34 for LPP, SLPP, and LSDA respectively, which are the same as used in [12].

Try to observe the experimental results, the proposed concept with adding temporal information do have admired improvement on the recognition results, especially for the feature type DTM and TTM. Note that the results of SE by LPP and LSDA are not the same as reported in [12], it may because of the difference in the normalization process. Thus, the improvement rates of recognition accuracy, as listed in Table 1, are applied to make comparison with the results in [12], and the results are better than LPP (improved by 2.8 %) and LSDA (improved by 2.8 %) in [12]. Moreover, the improvement rates are various for different spatio-motion subspaces. As observed, the supervised ones (SLPP and LSDA), which include the class label information, do not provide better results than LPP. It is inferred that if we include the class label information at obtaining the motion subspace; it may ruin the data motion structure because the class label will make the data points with same label close and different labels apart. Thus, the second step won't get any benefit from the structure. Hence, in our framework, the motion subspace would not take the class label for necessity.

In the above experiment, temporal parameter $t = 2$ is set, i.e. 5-frame segment is chosen to be the temporal data. For further investigation, the influence of the length in the temporal segment, we perform experiments with different parameter, here we set t=2, 3, i.e. 5-frame segment and 7-frame segment are used. Table 1 and Table 2 show the average recognition results for different feature types. We can see that with longer segment won't make the result better definitely. We observe the experimental results which are bold in Table 1 and 2, we can see that with longer segment don't seem to bring benefit to every method. For DTM and TTM, they do have the improvement, but unfortunately for LTM, it gets the worse result. We think it may

because the longer segment you choose, you may lose more attention to those more important data when you calculate the Mahalanobis distance metric.

4.3 Performance Evaluation in Noisy Data

To go deeper in the framework, we construct additional experiments to see if the framework has flexibility. Thus we add salt and pepper noise with different variances to the data, and see the influence of adding noise. Here simply using Matlab function to produce noisy data with different variances 0.1 0.15 and 0.2, the noisy data are just like in Fig. 2. We test each noisy data set individually, and we choose LPP with DTM and LPP with TTM as the two uniform frameworks. The results are show in Table 3.

(a) (b) (c)

Fig. 2. The noisy data image with different variances. (a) v = 0.1 (b) v = 0.15 (c) v = 0.2.

In Table 3, try to observe that for v = 0.1 and v = 0.15, the noise problem seems not to have too much influence on the results. But as v = 0.2, it's easily seen that the noise problem become obvious. Just take a look at the results when v = 0.2, the accuracy values decrease for 2% to 3%. Although the noise do decrease the performance, still the accuracy rates are admired. Thus it can be admired that the proposed framework in this paper does have good improvement here.

Table 3. Classification accuracies with v = 0.1, 0.15, and 0.2 in our two framework LPP+DTM and LPP+TTM

t=2	v=0	v=0.1	v=0.15	v=0.2
LPP+DTM	89.45	89.01	89.52	87.23
LPP+TTM	88.56	88.11	88.58	85.28

5 Conclusions

In this paper, we propose a novel framework TVTL for human action recognition. The TVTL tries to find a proper way to measure the similarity by taking temporal information into consideration. In the frame-by-frame experimental results, it got good performance in section 4. So here in this paper, the system do find a good way to add the temporal concept into the action recognition process. Thus it's confirmed that the TVTL framework do have positive improvement for action recognition. Especially our DDM and TTM frameworks, they do have impressive progress in the experiment. Moreover, even with the noise's disturbance, the system still survive and have good performance as well.

References

1. Ali, S., Basharat, A., Shah, M.: Chaotic invariants for human action recognition. In: ICCV, pp. 1–8 (2007)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems 14, 585–591 (2002)
3. Bissacco, A., Chiuso, A., Ma, Y., Soatto, S.: Recognition of human gaits. CVPR 2, 52–57 (2001)
4. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. PAMI 23(3), 257–267 (2001)
5. Bregler, C.: Learning and recognizing human dynamics in video sequences. In: CVPR, pp. 568–574 (1997)
6. Cai, D., He, X., Zhou, K., Han, J., Bao, H.: Locality sensitive discriminant analysis. In: IJCAI, pp. 708–713 (2007)
7. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: ICCV Work-shop: VSPETS, pp. 65–72 (2005)
8. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, vol. 2, pp. 726–733 (2003)
9. Elgammal, A., Lee, C.S.: Inferring 3D body pose from silhouettes using activity manifold learning. In: CVPR, vol. 2, pp. 681–688 (2004)
10. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Action as space-time shapes. PAMI 29(12), 2247–2253 (2007)
11. He, X., Niyogi, P.: Locality preserving projections. Advances in Neural Information Processing Systems 16, 153–160 (2003)
12. Jia, L.K., Yeung, D.Y.: Human Action Recognition Using Local Spatio-Temporal Discriminant Embedding. In: CVPR, pp. 1–8 (2008)
13. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: ICCV, pp. 166–173 (2005)
14. Laptev, I.: On space-time interest points. IJCV 64(2-3), 107–123 (2005)
15. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR, pp. 1–8 (2008)
16. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and Viterbi path searching. In: CVPR, pp. 1–8 (2007)
17. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learn-ing of human action categories using spatial-temporal words. In: BMVC (2006)
18. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 22, 290(5500), 2323–2326 (2000)
19. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR, vol. 3, pp. 32–36 (2004)
20. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 22, 290(5500), 2319–2323 (2000)
21. Tran, D., Sorokin, A.: Human Activity Recognition with Metric Learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
22. Wang, L., Ning, H.Z., Tan, T.N., Hu, W.M.: Fusion of static and dynamic body biometrics for gait recognition. In: ICCV, pp. 1449–1454 (2003)
23. Wang, L., Suter, D.: Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model. In: CVPR, pp. 1–8 (2007)

24. Wang, L., Suter, D.: Learning and matching of dynamic shape manifolds for human action recognition. IEEE Trans. on IP 16(6), 1646–1661 (2007)
25. Wang, L., Suter, D.: Visual Learning and Recognition of Sequential Data Manifolds with Applications to Human Movement Analysis. CVIU 110(2), 153–172 (2008)
26. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research 10, 209–244 (2009)
27. Yacoob, Y., Black, M.J.: Parameterized modeling and recognition of activities. CVIU 73(2), 232–247 (1999)
28. Yan, S., Xu, D., Zhang, B., Zhang, H.J.: Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. IEEE Trans. on PAMI 29(1), 40–51 (2007)
29. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. CVPR 2, 123–130 (2001)

Face Alignment Using Boosting and Evolutionary Search

Hua Zhang[1], Duanduan Liu[2], Mannes Poel[3], and Anton Nijholt[3]

[1] College of Software Engineering, Southeast University, Nanjing 210096, China
reynzhang@sina.com
[2] Lab of Science and Technology, Southeast University, Nanjing 210096, China
liuduanduan@seu.edu.cn
[3] Human Media Interaction, University of Twente, P.O. Box 217
7500 AE Enschede, The Netherlands
{anijholt,mpoel}@cs.utwente.nl

Abstract. In this paper, we present a face alignment approach using granular features, boosting, and an evolutionary search algorithm. Active Appearance Models (AAM) integrate a shape-texture-combined morphable face model into an efficient fitting strategy, then Boosting Appearance Models (BAM) consider the face alignment problem as a process of maximizing the response from a boosting classifier. Enlightened by AAM and BAM, we present a framework which implements improved boosting classifiers based on more discriminative features and exhaustive search strategies. In this paper, we utilize granular features to replace the conventional rectangular Haar-like features, to improve discriminability, computational efficiency, and a larger search space. At the same time, we adopt the evolutionary search process to solve the deficiency of searching in the large feature space. Finally, we test our approach on a series of challenging data sets, to show the accuracy and efficiency on versatile face images.

Keywords: face alignment, boosting appearance models, granular features, evolutionary search.

1 Introduction

Face alignment is usually regarded as minimizing the distance between a template and a given face image. Among the various technologies of face alignment, Active Shape Models (ASM) [1] and Active Appearance Models (AAM) [2] have gradually taken the stage center. ASM utilized the local texture information in search of a better template, and AAM constructed appearance models according to shape parameters and global texture constraints. After ASM and AAM, Zhou *et al.* [3] introduced Bayesian Tangent Shape Model (BTSM) with an EM-based method to implement the MAP estimation, Liang *et al.* [4] utilized a Constrained Markov Network for accurate face alignment, and Boosting Appearance Models (BAM) [5] presented a discriminative method with boosting algorithm and rectangular Haar-like features, which resulted in outstanding accuracy and robustness. Enlightened by BAM, the speed of face alignment can be improved by more discriminative features and boosting classifiers bring in the benefit of

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 110–119, 2010.

computational efficiency. Huang *et al.* [6] introduced granular features to form a larger feature space. At the same time, evolutionary search process [7] made great improvements on exploring the better granular features in a large feature space. In this paper, we improve BAM by introducing granular features and evolutionary search. Firstly, we generate a large number of positive samples by wrapping the image from the average shape, and then we harvest negative samples by randomly perturbing parameters from the current shape. Secondly, we generate a series of granular features from the feature space. After the evolutionary search, we can find a set of granular features to construct a strong classifier. Finally, the face alignment process is regarded as finding the warped image, which has a higher response than the final threshold of a strong classifier.

This paper is organized in the following way: Section 2 introduces Boosting Appearance Models, Section 3 expatiates the process of exploring the better weak classifier of boosting algorithm, and the fitting process of alignment is presented in Section 4. Finally, Section 5 compares our method with other methods by experiments.

2 Boosting Appearance Models

Active Appearance Models (AAM) [2] are composed of a shape model, a texture model, and a fitting method. Boosting Appearance Models (BAM) [5] propose a more discriminative method via rectangular Haar-like features and boosting. Inspired by AAM and BAM, we propose a framework based on granular features, a Bayesian stump weak learner, and evolutionary search for features.

2.1 Shape and Texture Models in Active Appearance Models [2]

Inspired by Active Appearance Models (AAM) [2], the morphable face model is generated from a set of facial images. From a giving face database, we manually label a series of 2D annotations $\{x_i, y_i\}, i = 1, 2, \ldots, n$, which include important facial components such as eyes, nose, and mouth. For each face image, we constitute a shape $s = [x_1, y_1, \ldots, x_n, y_n]^T$ from these annotations. After applying Principle Component Analysis (PCA) [2], a morphable shape model is constructed as

$$s = \overline{s} + U_s P, \tag{1}$$

where $\overline{s}$ is the mean shape, $P = [p_1, .., p_n]$ are the first n principal component vectors, and U_s is the coefficients of s with respect to these first n principal components. In virtue of shape, texture information of the images is warped into the mean shape $\overline{s}$ via piecewise affine transformation $T(x, y; U_s)$. If we want to warp an image $\mathbf{I}$, a set of pints $I_j \in \mathbf{I}, j = 1, \ldots, n$ in the coordinates $\{x_i, y_i | i \in 1, \ldots, n\}$ are mapped to new positions $\{x'_i, y'_i\}$ by defined warping function

$$T(x, y; P) = [1, x, y]\mathbf{A}(P), \tag{2}$$

where $\mathbf{A}(P)$ is a transformation matrix between average shape $\overline{s}$ and current shape s [2]. When shape parameters P are given, the $\mathbf{A}(P)$ matrix needs to be computed for

each triangle. It is a method to normalize all warp images as the same size. Then the eigen-texture information is presented by

$$t = \overline{t} + U_t Q. \tag{3}$$

Finally the PCA based shape and texture model are combined to form the appearance model.

Conventionally, the fitting process of AAM is in search of the minimum between current warped texture and the model texture. Hence, the matching process is

$$\delta(P) = \|t_s - t_m\|^2, \tag{4}$$

where P is the shape parameters of the shape model, $t_s = I(T(x, y; P))$(I means cropping the texture from the transformed image $T(x, y; P)$ is the warped texture of the current shape, and t_m is the current model texture given by Equation 3. By gradient ascent methods, this minimization can be solved.

2.2 Appearance Modeling

Similar to AAM and BAM, our appearance model is derived from the warped image $I(T(x, y; P))$. If we consider face alignment as a two-category classification problem, the shape instance $S(P)$ is the manually landmark of a face image I, then P becomes the positive shape parameters. At the same time, we perturb P to generate the negative shape parameters. If we can define a function $h(\bullet)$, which outputs positive score when the given sample is positive, or outputs negative score when the given sample is negative, then we can collect a set of $h(\bullet)$ to add the responses from $h(\bullet)$. When the added response is over a given threshold, the current parameters are just the landmark parameters. Adaboost is a simple and robust method to learn an accurate classifier from a set of weak classifiers [8]. After a feature $\theta_i = \theta(I(T(x_i, y_i; P_i)))$ is extracted from the wrapped image as a weak classifier, we can construct a combined strong classifier by these features. Therefore, we define a combination of many weak classifiers and local features as the appearance model

$$H(I(T(x, y; P)), \Theta) = \sum_{m=1}^{M} h_m(I(T(x, y; P)), \theta_i), \theta_i \in \Theta, \tag{5}$$

where $h(I(T(x, y; P)), \theta_i)$ is a function on using feature θ_i to operate image patch $I(T(x, y; P))$. Namely, $H(\bullet)$ and $h_m(\bullet)$ are strong and weak classifiers respectively.

2.3 Real Adaboost Learning for Strong Classifier

Boosting [8] algorithm is a method of integration of various "weak" classifiers into a powerful "board". In this paper, we choose Real AdaBoost [9] algorithm, which returns the response of weak classifiers as real numbers (Table 1). Given a set of faces with annotated landmarks, we generate training data for boosting learning. From each shape, we warp image $I(W(x, y; P))$ as the positive samples. Then we randomly perturb P to

get the negative samples. Each sample is normalized to the same size to construct the training set (Figure 2(a)). The final strong classifier contains a series of weak classifiers, which preserves a granular feature and a threshold. After the responses of the weak classifier are accumulated, we can get the response trace shown in Figure 2(b).

Table 1. Real Adaboost for learning strong classifier

- Input and initialize
  Training data $\{x_i; i = 1, \ldots, K\}$, and their labels $\{y_i; i = 1, \ldots, K\}$.
  Initialize weights $\omega_i = \frac{1}{K}, i = 1, \ldots, K$.
- For $m = 1, \ldots, M$, do
  (1)Fit the class probability estimate $h_m(x) = \arg\min \sum h(x) = \sum_{i=1}^{k} \omega_i (y_i - h(x_i))^2$.
  (2)Choose this weak classifier $h_m^* = \frac{1}{2} \log \frac{h_m(x)}{1-h_m(x)} \in R$.
  (3)Update the weight $\omega_i = \frac{\omega_i \exp[-y_i h_m(x_i)]}{Z_i}$, where Z_t is a normalization factor.
- Output
  The strong classifier $sign[H(x)] = sign[\sum_m h_m(x)]$.

3 Learning Sparse Granular Features for Weak Classifier

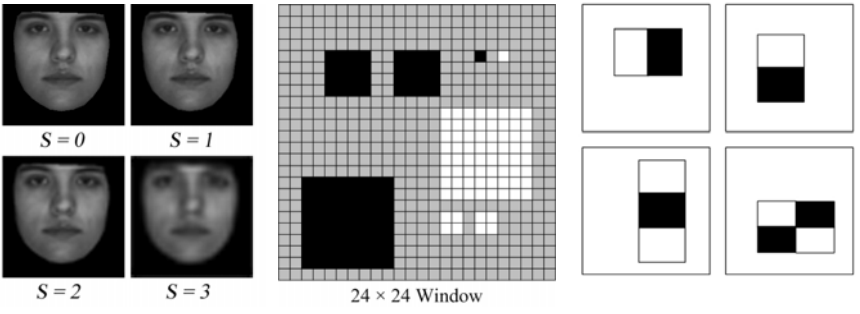
Since face alignment is time-constrained, BAM constructs the weak classifiers based on rectangular Haar-like features [10], which lead to great success because of integral image. However, Haar-like features encounter defects in irregular patterns. In order to overcome this difficulty, Huang *et al.* [6] presented a granular space to generate a series of granular features, which adopts a heuristic search algorithm to search for discriminative sparse features. In the process of search for better features, Treptow and Zell [11], Abramson *et al.* [7] utilized an evolutionary method to find better features. We combine these ideas, and introduce an evolutionary search algorithm to pursue discriminative granular features.

3.1 Granular Features

A granular space is established by a pyramid of bitmaps $\{I_0, I_1, I_2, I_3\}$, and each layer of the pyramid is denoted from a smooth filtering in a way of averaging $2^s \times 2^s$ patches of the input image (Figure 1(a)). In space, a sparse feature is represented by a linear combination of several granules, as

$$\theta = \sum_i \alpha_i I(p(x_i, y_i, s_i)), \alpha \in \{-1, +1\}, s_i \in \{0, 1, 2, 3\}, \tag{6}$$

where $I(\bullet)$ indicates the pixel data of a granule. Through three parameters: x-offset x_i, y-offset y_i, and scale s_i, a granule $p(x_i, y_i, s_i)$ means a square at the coordinate (x_i, y_i) with the size of $2^{s_i} \times 2^{s_i}$. From a 24×24 reference window, we can totally extract $\sum_{s=0,1,2,3} (24 - 2^s + 1)^2 = 1835$ different granules. Compared to conventional rectangular Haar-like features [10], sparse granular features are more scalable and robust [6].



(a) (b) (c)

Fig. 1. (a)Pyramid of granular space.(b)Example of granular sparse features which black or white color indicates the coefficient α_i in Section 3.1. (c)Examples of rectangular Haar-like features [10] for initialization in Section 3.3.

3.2 Bayesian Stump Look Up Table Weak Classifier

When we consider the problem of two-category classification, the probability of Bayesian error is defined as

$$P(error|x) = \begin{cases} P(\omega_1|x) & \text{if we decide } \omega_2 \\ P(\omega_2|x) & \text{if we decide } \omega_1, \end{cases} \tag{7}$$

and the expected error is $P(error) = \int_{-\infty}^{\infty} P(error, x)dx = \int_{-\infty}^{\infty} P(error|x)p(x)dx$. If we follow the Bayes' decision rule, we decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$ and otherwise decide ω_2, then Equation 7 becomes $P(error|x) = \min[P(\omega_1)|x, P(\omega_2|x)]$, and the total Bayesian error is

$$B_{error} = P(error) = \int_{-\infty}^{\infty} \min[P(\omega_1)|x, P(\omega_2|x)]dx. \tag{8}$$

Xiao *et al.* [12] proposed a method called Bayesian Stump to find $P(\omega_c, x), c \in \{1, 2\}$ by using histogram to estimate the probability distribution. We divide all features' output value $\{\mu(\theta_i)\}$ into k sections $\delta_k = (r_{k-1}, r_k]$, and the histogram of $P(\omega_c, x)$ is

$$P(k, \omega_c) = \int_{\mu(\theta_i)\in\delta_k} P(\mu(\theta_i), \omega_c)d\mu(\theta_i), c \in \{1, 2\}. \tag{9}$$

Following the method in [12], we can easily build a k-bins Bayesian Stump. Moreover, we can extend it to a Look Up Table(LUT) weak classifier for RealBoost algorithms by using log-likelihood output to replace the binary output in every interval. In summary, we can define the weak classifier as

$$h(x, \theta) = \frac{1}{2} \ln \sum_{k=1}^{K} \{\frac{W_{-1}^k + \varepsilon}{W_{+1}^k + \varepsilon}\} B_k(\theta(x)), B_k(u) = \begin{cases} 1, u \in \delta_k \\ 0, \text{otherwise} \end{cases}. \tag{10}$$

W_{+1}^{j} and W_{-1}^{j} are the total weights of positive and negative samples falling into the jth bin, $\theta(x) = \theta(I(T(x, y; p)))$ represent the feature under the current wrapped face patch, and ε is a small constant to avoid that denominator is zero.

3.3 Evolutionary Search for Sparse Feature Selection

Although sparse granular features bring abundance to construct a versatile classifier, the gigantic number of possible features consume enormous computational resources. To address this issue, an evolutionary search process is introduced to efficiently constitute a compact granular feature set. Howard *et al.* [13] implement Genetic Programming (GP) to detect ships in satellite images. Treptow and Zell [11] combine an evolutionary algorithm with the Adaboost framework to detect human faces. Abramson *et al.* [7] use a hybrid method of Hill Climbing and Evolutionary Search to detect cars. In our method, firstly we generate a large number of traditional Haar-like features in the granular space (Figure 1(c)). Through calculating the *Fitness* as Function 11, we choose l features with the highest score to construct the initial feature set Θ_i. After many rounds of evolutionary search loop, we can harvest a large set Θ_i with diversified granular features. The best feature is drawn out as the current weak classifier from the set.

Fitness evaluation of a sparse granular feature reflects the discriminability of the feature and dominates the search process. In order to improve the performance of LUT weak classifier (Section 3.2), we should find the feature with lower Beyesian error. Meanwhile, since the feature with less granulae give rise to less computational cost and simpler structural complexities, we prefer finding the features with few granule and low Bayesian error. The discriminability of a sparse granular feature is defined as $D(\theta_i) = 1 - B_{error}(\theta_i)$, where $B_{error}(\theta_i)$ is the upper bound of Bayesian error. The sparse feature is more discriminative when $D(\theta_i)$ gets higher. By considering the complexity of features, we present the *Fitness* function as

$$\mathit{Fitness}(\theta_i) = D(\theta_i) - \beta^c, \tag{11}$$

where c is the granule number of the sparse feature, β is the empirical parameter for the penalty for more granules. Generally speaking, we can preserve hundreds of granular features in each loop.

4 Face Fitting with Boosting Classifier

According to (10), the final classifier can be written as

$$H(\Theta, x, y; p) = \sum_{m=1}^{M} \frac{1}{2} \ln \sum_{k=1}^{K} \{\frac{W_{-1}^{k} + \varepsilon}{W_{+1}^{k} + \varepsilon}\} B_k(\theta_m(I(W(x, y; P))), \theta_m \in \Theta. \tag{12}$$

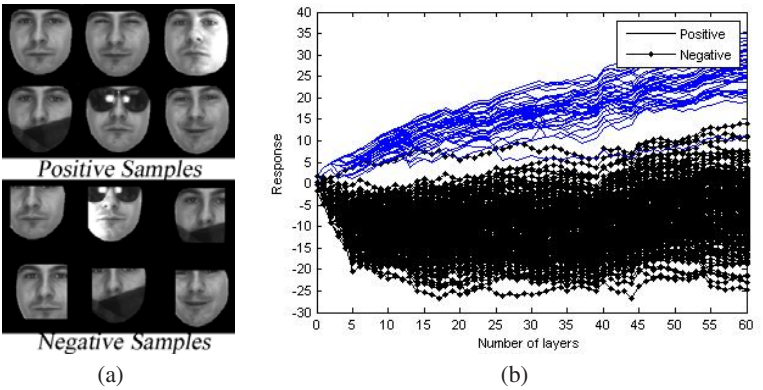
And the derivative by P is

$$\frac{dH}{dP} = \frac{1}{2} \sum_{m=1}^{M} \frac{1}{M} \sum_{k=1}^{K} \{\frac{W_{-1}^{k} + \varepsilon}{W_{+1}^{k} + \varepsilon}\} \nabla B_k \nabla I \nabla \theta_m \frac{\partial W}{\partial P}, \tag{13}$$

$$M = \sum_{k=1}^{K} \{\frac{W_{-1}^{k} + \varepsilon}{W_{+1}^{k} + \varepsilon}\} B_k(\theta_m(I(W(x, y; P))). \tag{14}$$

Table 2. Weak classifier learning based on evolutionary search for sparse granular feature

- Input
  Training set $\{x_i\}$,corresponding weight set $\{\omega_i\}$, and Haar-like features set $\Theta = \{\theta_1, \ldots, \theta_n\}$in the granular space.
- Initialize
  Choose better features with higher *Fitness*, add them to initial feature set $\Theta_i = \{\theta_i | \forall\theta_i \in \Theta, \forall\theta_i \in \{\Theta - \Theta_i\}, \mathit{Fitness}(\theta_i) \geq \mathit{Fitness}(\theta_j)\}$, constrict the set as $\|\Theta_i\| = l$.
- Evolutionary search loop
  (1)To every feature θ_i in the set Θ_i, we implement variance in the four ways.
  ◇**Add**. If this granular feature contains less than eight granula, we add a new granule. All possible granulae in the granular spaces are separately added into this feature to generate new features.
  ◇**Delete**. Delete a granule in the current granular features.
  ◇**Move**. To each granule, we randomly move the coordinate between -5 and 5 pixels.
  ◇**Resize**. To each granule, we randomly adjust the scale s to change its size.
  (2)After these variations, we can harvest varied granular features θ_i to form a set Θ_g. Then we randomly choose m features from Θ_g, and combine them with initial feature set $\Theta_i = \Theta_i \cup \{\theta_i | \theta_i \in \Theta_g\}$, $\|\{\theta_i\}\| = m$.
- Weak classifier learning
  (1)To every feature $\theta_i \in \Theta_i$, we construct a weak classifier $h(x, \theta_i)$.
  (2)Find the weak classifier $h(x, \theta) = \arg\max_{h(x,\theta)}(\mathit{Fitness}(\theta_i))$, which has the highest *Fitness*.
- Output
  The weak classifier $h(x, \theta)$ and corresponding granular feature θ.



(a) (b)

Fig. 2. (a)Some positive samples and negative samples. (b)Response trace of different samples.

Face alignment factually is a process to find the best parameters P to get the best shape. After given a face image I, we firstly compute the warped $I(W(x, y; P))$ image via a piecewise affine transformation. The face alignment algorithm is presented in Table 3.

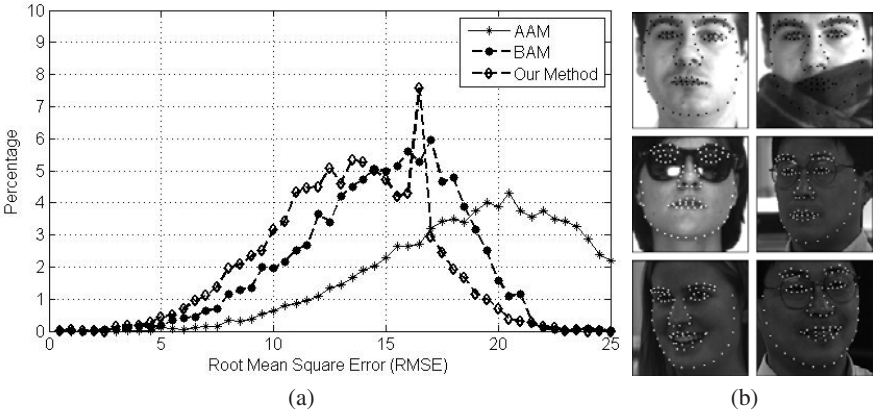
Table 3. Face alignment algorithm

- Input
  Input image I, initial shape parameters P, boosted strong classifier H, initial response$t$ and final rejection threshold T.
- While $t < T$
  (1)Warp I with piecewise affine transformation to generate $I(W(x, y; P))$.
  (2)Compute the current response t with each classifier in 10.
  (3)Compute the current ∇P by Equation 13.
  (4)Update $P = P + \nabla P$.
- Output
  The shape parameters P.

5 Experiments

In our experiments, we have collected about 2148 images from several databases, including the AR database [14], FERET database [15], PIE database [16]. We randomly select 1208 images for training and reserve the rest for testing. For each image in the training set, we manually label 87 points on facial components such as eyes, eyebrow, nose, mouth, *etc*. In order to train boosting classifiers, we generate 1208 positive samples. To every positive sample, we perturb parameters to generate 10 negative samples. After boosting training, we can get a classifier with eighty weak classifiers. In the process of model calibration, we choose images from the AR database. In AR database, the same person is shown in different images under various conditions. We choose 13 different conditions from the same person to calibrate the classifiers.

To test our method, we have implemented benchmark tests among Active Appearance Models (AAM), Boosting Appearance Models (BAM), and our method (Figure 3(a)). In Figure 3(a), Root Mean Square Error (RMSE) indicates the distribution of



(a) (b)

Fig. 3. (a)The RMSE results among AAM, BAM, and our method on test set. (b)Face alignment results by our method.

between test results and ground-truth label. Figure 3(b) shows some face alignment results by our method.

6 Conclusion

In this paper, we have introduced a novel framework of face alignment, which brings in granular features, an evolutionary search process, and boosting learning process to find better weak classifiers. Since granular features produce a lot of diversified and discriminative rectangles, the boosting process has better discriminative capabilities with less weak classifiers. At the same time, we implement an evolutionary search in order to deal with deficiency of gigantic feature space. Evolutionary search not only generates versatile granular features, but also guarantees the robustness of classifiers. With granular features and evolutionary search, we can construct a novel fitting process for real time face alignment. In the future, there are more improvements that can be implemented on our approach. Firstly, other features can be added into the training system to achieve better discriminative capabilities. Secondly, calibration methodology can be used to tune the final strong classifier. Therefore, new explorations still wait for further consideration.

Acknowledgement

This work is partially supported by the European IST Programme Project FP6-033812 (AMIDA). Many thanks to Lynn Packwood for the proof reading and many important suggestions.

References

1. Cootes, T.F., Cooper, D.H., Taylor, C.J., Graham, J.: Trainable method of parametric shape description. Image and Vision Computing 10, 289–294 (1992)
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active apperance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681–685 (2001)
3. Zhou, Y., Gu, L., Zhang, H.J.: Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian reference. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 109–116 (2003)
4. Lin, L., Fang, W., Ying-Qing, X., Xiaoou, T., Heung-Yeung, S.: Accurate face alignment using shape constrained Markov network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1313–1319 (2006)
5. Liu, X.: Generic face alignment using boosted appearance model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1–8 (2007)
6. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 671–686 (2007)
7. Abramson, Y., Moutarde, F., Steux, B., Stanciulescu, B.: Combining adaboost with a hill-climbing evolutionary feature-search for efficient training of performant visual object detectors. In: Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence (2006)

8. Freund, Y., Schapire, R.: A decision-theoretic generaliation of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
9. Fridman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28, 337–374 (2000)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
11. Treptow, A., Zell, A.: Combining adaboost learning and evolutionary search to select features for real-time object detection. In: Proceedings of the Congress on Evolutionary Computation, vol. 2, pp. 2107–2113 (2004)
12. Xiao, R., Zhu, H., Sun, H., Tang, X.: Dynamic cascade for face detection. In: Proceedings of IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007)
13. Howard, D., Roberts, S.C., Brankin, R.: Evolution of ship detectors for satellite sar imagery. In: Langdon, W.B., Fogarty, T.C., Nordin, P., Poli, R. (eds.) EuroGP 1999. LNCS, vol. 1598, pp. 135–148. Springer, Heidelberg (1999)
14. Martinez, A.R., Benavente, R.: The AR face database. CVC Technical Report, vol. 24 (1998)
15. Philips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1090–1104 (2000)
16. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1615–1618 (2003)

Tracking Eye Gaze under Coordinated Head Rotations with an Ordinary Camera

Haibo Wang[1,2], Chunhong Pan[1], and Christophe Chaillou[2]

[1] LIAMA, Institute of Automation, Chinese Academy of Sciences, China
{hbwang1427,chunhongp}@gmail.com
[2] LIFL & INRIA Futurs, University of Lille, France
{haibo.wang,christophe.chaillou}@lifl.fr

Abstract. Previous efforts in eye gaze tracking either did not consider head motion, or considered the 6 DOF head motions with multiple cameras or light sources. In this paper, we show that it is possible to track eye gaze under naturally head rotations(Yaw and Pitch) with only an ordinary webcam. We first carry out a study to examine the occurrence of eye-head coordination, and then show how to track such coordinated gaze by deriving a linear coordination equation and developing a tracking system based on a single webcam. Besides the theoretical aspect, we develop a vision-based tracking framework that can achieve an acceptable tracking accuracy in our experiments for estimating such eye-head coordinated gaze.

1 Introduction

Eye gaze tracking is one of the most active research topics in computer-human interaction. Since the last decade, an extensive number of non-intrusive eye gaze systems have been proposed [7] [1] [5] [2] [12]. Although good performances have been achieved[11], these earlier systems are limited in keeping head still while estimating eye gaze.

Recent efforts have attempted to incorporate head motion into eye gaze estimation. These attempts can be broadly clustered to be 3D gaze direction-based and 2D mapping-based[16]. The former method determines gaze point by recovering 3D gaze direction and simply intersecting with the scene. It allows free head motion, but can only estimate gaze direction rather than giving gaze targets[13] [8]. On the contrary, the latter method encodes a set of eye-related vectors as inputs for a calibrated mapping function to determine gaze targets. This method is widely used due to its high tracking accuracy. One typical example is the pupil center and corneal reflections technique, a.k.a. PCCR[5] [14]. However, the classical PCCR suffers from a calibration-decay problem caused by head motion[16]. In order to overcome it, multiple stereo cameras or IR lights must be added in the PCCR systems[5] [15]. In this way, they are able to track eye gaze under fully natural head motion(6 DOF). The main drawbacks are that the camera sets need to be carefully calibrated and IR lights are expensive and usually inconvenient to be mounted.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 120–129, 2010.

Our objective is to track eye gaze under natural head motion with only an ordinary camera. However, Guestrin *et al* [5] proved that using a single camera was impossible to track eye gaze under natural head motion. On the other hand, neurophysiological studies [6] [4] [3] manifest that the free head motions along gaze shift are usually the rotations in the same directions. They specify this phenomenon as eye-head coordination of gaze shift. We therefore turn to explore the possibility of tracking gaze with one camera, while only considering the coordinated head rotations. This is proven feasible by developing a new tracking approach in this paper. Within this approach, the price we have to pay is two-time calibrations, although acceptable in practice. In particular, our approach follows the 2D mapping-based scheme [16] to directly output the targeted points. In order to compensate the affect of head rotations, we first derive a linear relationship between gaze vector and head rotations, and then develop a monocular vision-based framework to track gaze, which needs to detect iris-corner vector, track head rotations and warp facial images to extract a *head-normalized* iris-corner vector.

The paper is organized as follows. Section 2 depicts a user study about eye-head coordination. Section 3 describes the derivation for tracking head-coordinated eye gaze and Section 4 presents the gaze tracking framework. Experimental results are shown in Section 5.

2 Eye-Head Coordination Study

We first describe a study that we conducted to investigate the frequency of eye-head coordination in gaze behaviors. Based on the theory proposed by E.G. Freedman *et al* [4] [3] that the total head movement amplitude and gaze amplitude are linearly correlated in a certain range, we examine such linearity as the quantitative evidence of the occurrence of eye-head coordination.

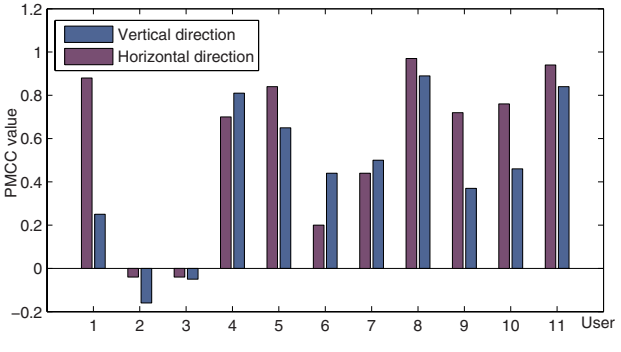


Fig. 1. The PMCC values between head and gaze amplitude among 11 participants

The study was conducted with 11 participants from our laboratory. Each participant is asked to stare at a visual sphere moving arbitrarily on a 19-in screen. Meanwhile, the trajectory of the sphere is recorded as gaze amplitude and the participant's head amplitude is tracked using the developed head pose tracker(See Section 4). The correlations between head rotation and gaze amplitude is measured in terms of the Pearson Product-Moment Correlation Coefficient(PMCC), which is shown in Figure 1. We can see that the PMMC is significantly different as different person. If PMMC>= 0.6 indicates a possible linear relationship, we find that there are seven possible linearities in both directions. We also assess the influence of the distance between participant and screen on eye-head coordination. It turns out that the farther participant stays away from the screen, the less likely head rotations are to occur. Therefore, we can conclude that the occurrence of eye-head coordination is very likely to occur in gaze behaviors, although it is circumstance-dependent and varied among different persons. This study validates our proposal that it is indispensable to incorporate head rotations in eye gaze tracking.

3 Tracking Head-Coordinated Eye Gaze

3.1 The Problem Formulation

Let us consider tracking eye gaze in front of a monitor screen. The gaze directions are measured with respect to the horizontal and vertical ones of the monitor plane. We adopt some notations from Zhu and Ji [16] to describe our problem. First of all, via an interactive calibration, we are able to obtain a set of head-stationary gaze vectors $\{\mathbf{v'_1}, \mathbf{v'_2}, ..., \mathbf{v'_n}\}$ and its associated set of fixation positions$\{S_1, S_2, ..., S_n\}$ on the screen. The data are then used to compute the coefficients in a mapping function F such that given any head-stationary gaze vector $\mathbf{v'}$, its fixed position S can be yielded: $S = F(\mathbf{v'})$. F is often a first or second order polynomial function. Then suppose at time t of tracking stage, we succeed in tracking head motion $\mathbf{b_h} = \{\beta_x, \beta_y\}$(only pitch and yaw rotations) and gaze vector $\mathbf{v_t}$. Since head motion $\mathbf{b_h}$ results in a change of the eye pose, the gaze vector $\mathbf{v_t}$ is no longer a correct input of the mapping function F. In order to reuse F, we need to compensate $\mathbf{v_t}$ for the head-caused bias to form an equivalently *head-compensated* gaze vector $\mathbf{v'_t}$: $\mathbf{v'_t} = \beth(\mathbf{v_t}, \mathbf{b_h})$, where $\beth$ stands for the *head compensation function*. This also means that $\mathbf{v'_t}$ are supposed to look at the same point S_t as $\mathbf{v_t}$ does, which yields:

$$S_t = F(\mathbf{v'_t}) = F(\beth(\mathbf{v_t}, \mathbf{b_h})) \tag{1}$$

The key of our approach is thus to derive an appropriate $\beth$ function so that the head rotation bias on iris-corner vector can be compensated. This can be done by projecting the iris-corner vector on image coordinate via a pinhole camera model.

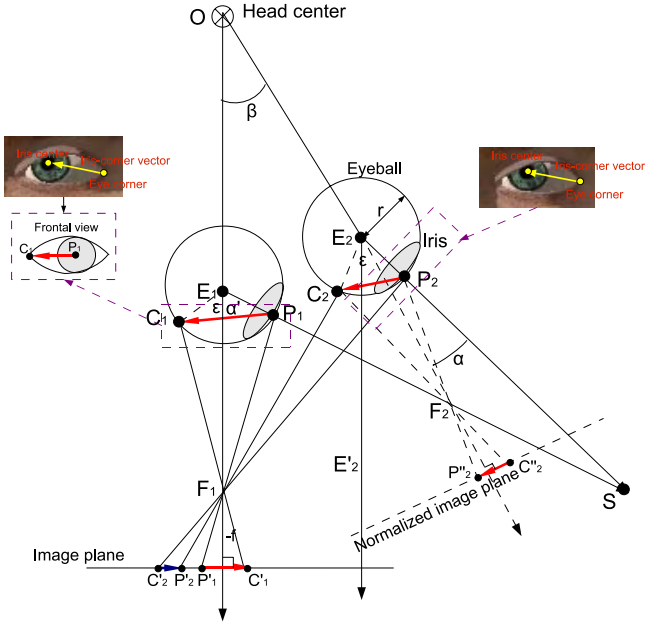


Fig. 2. Illustration of projecting eyes onto visual image plane via the pinhole camera model(Top view)

3.2 Deriving the Compensation Function ⊐

To derive ⊐, we analyze the kinematics of eye-head coordination using the pictorial notations in Figure 2. In the figure, eyeball is modeled as a sphere with a gray iris inside, and eye direction is denoted as the ray shooting from eyeball center to the target point. E_1 represents the eyeball center position when head stays frontal. Initially, in order to look at the target point S, eyeball rotates by a α' angle. When head shifts by a β angle, moving the eyeball center from E_1 to a new position E_2, the eyeball has to re-rotate to a new angle α to still look at S. The iris center is defined as the intersection point of gaze ray and iris outer surface, which is denoted by P_1 and P_2 at the two different locations. With some assumptions, we can derive a simple kinematical formula: $\alpha' \approx \alpha + \beta$, which means the total gaze amplitude is approximately the sum of eye and head amplitudes, which has also been empirically verified in neuroscience [4, 3].

The next step is to explore the relationship between iris-corner and the concurrent head rotations, based on the derived sum principle. We assume that the iris-corner vector is generated via a pinhole camera projection. As shown in Figure 2, we denote C_1 as an eye corner point, located in a face plane that is close to the eyeball surface. We assume that the face plane still remains stationary when eyeball rotates so that the corner point keeps a gaze-stationary point. As shown, when the head rotates by β angle, the corner point moves from C_1 to a

new position C_2. Based on the geometrical relationship shown in the figure, we are able to derive a linear formula written in the matrix form as:

$$\mathbf{b_h} = \mathbf{C}\triangle\mathbf{v} \tag{2}$$

where $\mathbf{b_h} = \begin{bmatrix}\beta_x \\ \beta_y\end{bmatrix}$, $\mathbf{C} = \begin{bmatrix}C_x & 0 \\ 0 & C_y\end{bmatrix}$ and $\triangle\mathbf{v} = \begin{bmatrix}x_{P_1'C_1'} - x_{P_2''C_2''} \\ y_{P_1'C_1'} - y_{P_2''C_2''}\end{bmatrix}$. We name $\mathbf{C}$ the coefficient matrix. In order to determine it, we need a re-calibration procedure, during which the user is asked to look at the calibrated points with natural eye-head coordination. Thus, we can acquire a set of pairs of $\mathbf{b_h}$ and $\triangle\mathbf{v}$. All the calibrated pairs form a linear system, which can be solved using least squares solution to estimate the matrix $\mathbf{C}$. Without loss of generality, if we assume the $\mathbf{C}$ is non-singular, the *head-compensated* gaze vector $\mathbf{v}'$ is simply calculated by

$$\mathbf{v}' = \mathbf{C^{-1}b_h} + \mathbf{v}'' \tag{3}$$

Note, that we introduce a *head-normalized* gaze vector $\mathbf{v}''$ that accounts for normalizing the head rotations with the observable iris-corner vector $\overrightarrow{P_2'C_2'}$. In the following section, we will give the details about how to extract $\mathbf{v}''$ and track 3D head rotations $\mathbf{b_h}$ from grabbed images.

4 Eye Gaze Tracking Framework

This section will present an eye gaze tracking framework under coordinated head rotations. The framework needs to detect iris-corner vector, track head rotations and then extract head-normalized gaze vector.

4.1 Extracting Iris-Corner Vector

We follow the technique in [15] to detect iris-corner vector as the eye gaze indicator. In our vector detection, a face detector is first employed to localize face region and then an eye detector is used to find the potential eye region on the localized face, both of which are based on the object detection cascade of boosted

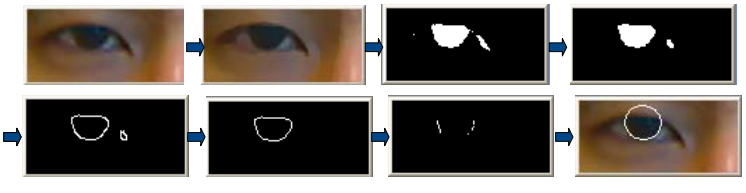


Fig. 3. Illustration of iris ellipse fitting. Follow the arrow direction: Original eye image → Meanshift Segmentation → Histogram-based binary segmentation → 'Open' morphology → Canny edge operation → Largest contour finding → Keeping near vertical edge points → Fitted ellipse.

classifier [10]. Subsequently, the iris and corner points are accurately found using the following method.

We combine the SIFT matching-based method and an ellipse fitting method to detect iris center. The SIFT was proposed by David Lowe [9] to detect feature points in images. We use the SIFT codes provided by Rob Hess[1] in our system. The reference SIFT features for iris center is manually selected during calibration. Setting the ratio of Euclidean distances between two nearest neighbor features $r = 0.65$ performs best in our tests. When the SIFT-based method fails, an ellipse fitting approach will take it over. This happens when the SIFT features between successive frames are over the matching threshold. The ellipse fitting approach, illustrated in Figure 3, is similar to the one proposed in [12]. On the other hand, the ellipse-fitted method may fail when the iris contour is unclear. In this case, the SIFT-based method will be re-evoked. Therefore, combining them enables us to acquire more robust iris detection results.

The SIFT-based method is also employed to detect the eye corner. Each single eye has two corners: inner corner and outer corner, but neither of them can be reliably detected over all the frames. Our strategy is to detect both of them and choose the one with smaller matching residual to form iris-corner vector. To gain robust performance, for both iris and corner points, their reference SIFT features are updated after each correct match. The SIFT-based detection is illustrated in Figure 4. Moreover, some distance and geometrical constraints, between the left and right eye, are added to eliminate the apparently wrong detections.

4.2 3D Head Pose Tracker

In order to estimate head rotations, we develop an online 3D head motion tracker that is person-independent. The head tracker is based on the 2D/3D registration of view-free appearances, in which we approximate head as an ellipsoidal. The appearances are generated from the video images with a piece-wise warping operation. By modeling the pixel intensity within appearance as a Multivariate Gaussian Distribution, the cost function to be minimized is defined as the *Mahalanobis* distance weighted by a confidence map. Then a standard first-order gradient descent approach is employed to optimize the cost function so as to recover head motion. Once the registration is accomplished, we adaptively update the appearance template using a recursive filter with a forgetting factor.

4.3 Extracting the Head-Normalized Gaze Vector

Extracting the *head-normalized* gaze vector $\overrightarrow{P_2''C_2''}$ (shown in Figure 2) consists of two steps: head normalization and iris-corner vector detection. The head normalization refers to normalizing the head rotations with the observable gaze vector $\overrightarrow{P_2'C_2'}$, which is implemented by an image warping operation. We define the warp

[1] http://web.engr.oregonstate.edu/~hess/

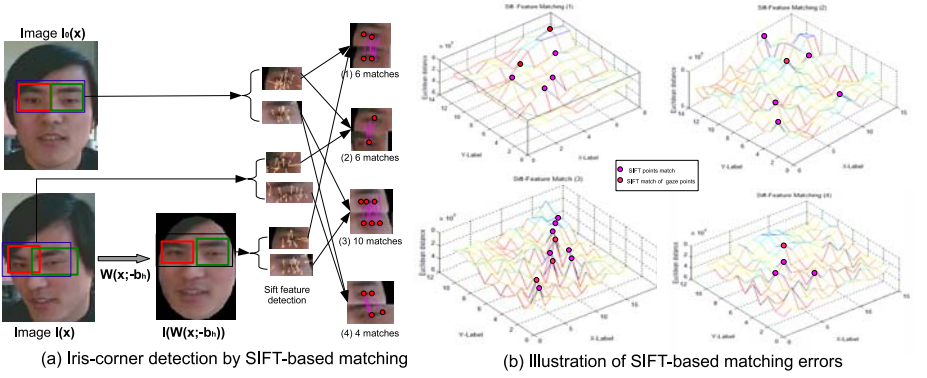


Fig. 4. The comparisons of SIFT-based gaze detection before and after head normalization. **(a)** The reference SIFT features are automatically detected on the eye region of the image $I_0(\mathbf{x})$. Two eye sub-images, extracted from input frame $I_{\mathbf{x}}$ and the normalized image $I(\mathbf{W}(\mathbf{x}; -\mathbf{b_h}))$, serve as the instances to detect SIFT feature points and match with the reference ones. The correct gaze point matchings are marked in red color. We can see that (1)(3) outperforms (2)(4) in the number of correct matchings. **(b)** Plots (1)-(4) show the matching residuals in (1)-(4) of sub-figure (a) respectively. X-Label denotes reference feature points and Y-Label means the query feature points. It is shown that the differences between marked peaks and noisy peaks in (1)(3) are more distinguished than in (2)(4). The average residuals in (1)(3) are 7.3×10^4 and 3.9×10^4, which are smaller than the ones of (2)(4), i.e. 1.0×10^5 and 6.0×10^4.

operator $\mathbf{W} : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \rightarrow \mathbb{R}^2$ as follows: a pixel point vector $\mathbf{x} = \{x, y\}^T$ on the face image plane is projected onto a 3D head surface(ellipsoidal in our case); then we transform the pose of head surface by $\triangle\mathbf{b_h}$ and back project the point from the 3D surface onto the image plane. For any given pixel $\mathbf{x}$, the warping function can yield its corresponding normalized point $\mathbf{W}(\mathbf{x}; -\mathbf{b_h})$. Denote $\mathbf{p'_2}$ as the point vector of P'_2 and $\mathbf{c'_2}$ as the point vector of C'_2. Their normalized points $\mathbf{p''_2}$ and $\mathbf{c''_2}$ can be given by: $\mathbf{p''_2} = \mathbf{W}(\mathbf{p'_2}; -\mathbf{b_h}); \mathbf{c''_2} = \mathbf{W}(\mathbf{c'_2}; -\mathbf{b_h})$.

In practice, we have two alternative options to extract $\overrightarrow{P''_2 C''_2}$. The first solution is to extract the points $\mathbf{p'_2}$ and $\mathbf{c'_2}$ on frame image $I(\mathbf{x})$ and then find the corresponding points $\mathbf{p''_2}$ and $\mathbf{c''_2}$. Contrarily, the second one is to first perform $\mathbf{W}$ to construct a *head-normalized* face image $I(\mathbf{W}(\mathbf{x}; -\mathbf{b_h}))$, and then run iris-corner extraction to locate $\mathbf{p''_2}$ and $\mathbf{c''_2}$ on $I(\mathbf{W}(\mathbf{x}; -\mathbf{b_h}))$. The essential difference between them is that the SIFT-based vector extraction will be applied before or after the head normalization. Although due to constructing $I(\mathbf{W}(\mathbf{x}; -\mathbf{b_h}))$, the second solution consumes more computational time, it's expected that applying the SIFT-based vector extraction on $I(\mathbf{W}(\mathbf{x}; -\mathbf{b_h}))$ will be superior to applying it on $I(\mathbf{x})$. Therefore, a comparative experiment is undertaken to evaluate the SIFT-based vector detection with and without head normalization. As shown in Figure 4, the results suggest that applying it after head normalization substantially outperforms the one before head normalization. We therefore adopt the second solution to extract $\overrightarrow{P''_2 C''_2}$.

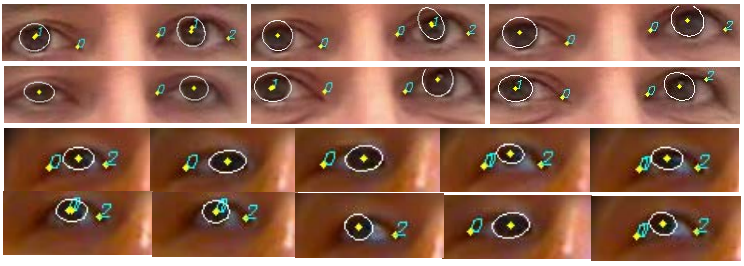


Fig. 5. Extracting the iris-corner vectors. The yellow point with green number is the SIFT-detected point while the one surrounded by an ellipse is the fitted ellipse center.

5 Experimental Results

Performance of the method extracting iris-corner vector was tested on real video clips containing eye gaze shift. The resolution of each video is 640 × 480 while the extracted eye region is about 85 × 40. Figure 5 shows some vector extraction results. We can see most of the iris-corner vectors are accurately extracted. In our tests, due to the usage of the combined detection methods, the vector detector is robust to facial variations and illumination changes to a certain degree. A full video demo is shown in http://video.google.com/videoplay?docid=6450216116737564885.

We have also applied the proposed framework to a gaze tracking system to test the accuracy of gaze estimation. The system was written in C/C++, using the renowned OpenCV and OpenGL library. In order to evaluate its accuracy, we test the system by collecting several image sequences with several testers. The tests are undertaken in front of a 19-in monitor with a video camera (640 × 480 pixels) mounted under it. There are two stages in the data collections. In the calibration stage, the testers are asked to stare at 12 evenly-distributed mark points in turn and yet maintain their heads stationary. In the tracking stage, the testers are allowed to freely rotate eye and head to stare the yellow-colored point that is randomly selected from the 12 calibrated points. The length of collected video for each tester is about 2 minutes.

We tested the accuracy of the gaze tracking with three actors. We first apply the system to extract calibrated iris-corner vectors, track head rotations and extract *head-normalized* gaze vectors. The maximum tracked head rotations in horizontal and vertical directions are 21.2° and 9.75° respectively. We then use these data to compute the matrix $\mathbf{C}$ and calculate the *head-compensated* vectors according to equation (2). The system accuracy is measured in the compensated errors, the pixel errors between the *head-compensated* vectors and calibrated vectors. For comparisons, we also measure the pixel errors between the observable vectors(such as $\overrightarrow{P_2'C_2'}$ in Figure 2) and calibrated vectors, called non-compensated errors. Figure 6 displays the statistics of the total accuracy measurements. As the four plots suggest, the compensated errors are substantially smaller than the non-compensated ones in horizontal direction while only slightly better in

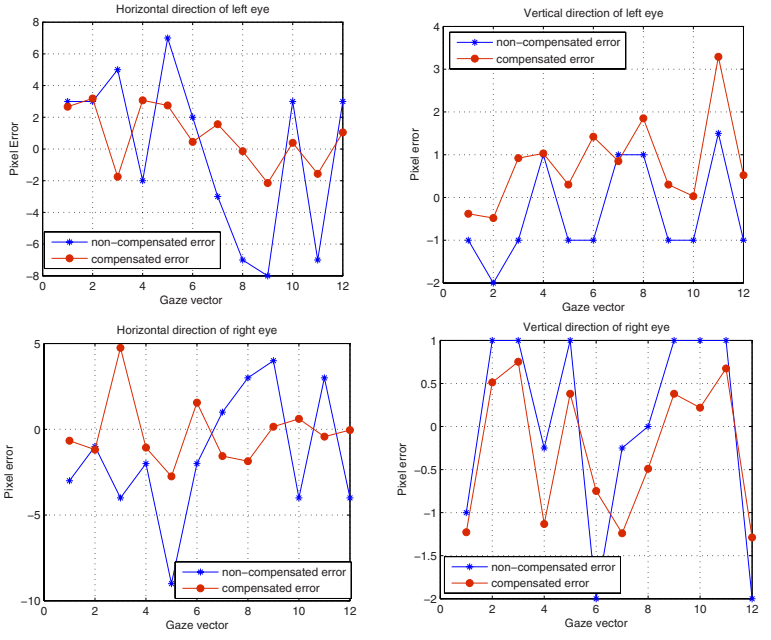


Fig. 6. The average pixel errors of eye gaze tracking under coordinated head rotations

vertical direction. Since the right part in equation (2) actually equalizes to the gaze tracking angle, we use this to compute the gaze tracking error. The average head-compensated error is about 5.06° and the non-compensated one is about 8.30°. We can see that compensating head rotation improves the gaze tracking accuracy by about 3.3°, which is a considerable value for a gaze tracking system. That is to say, ignoring the head rotations will significantly lower the accuracy of the gaze tracking system.

6 Conclusion

This paper addresses the problem of tracking eye gaze under coordinated head rotations. We first derive a linear relationship between gaze vector and head rotations and then develop a gaze tracking system, which enables us to track eye gaze under coordinated head rotations. Relying on a single webcam, we have to calibrate twice so that the linearity can be established properly. This procedure may be overcome by exploiting the deep correlations between eye and head motion. In addition, the accuracy of iris-corner detections must be improved, for example, by using sub-pixel accuracy [15].

Acknowledgement. This work was supported by National Natural Science Foundation of China under grant 60675012. We thank all the ALCOVE and the IGIT colleagues for their great help and support.

References

1. Baluja, S., Pomerleau, D.: Non-intrusive gaze tracking using artificial neural networks. In: Working Notes: AAAI Fall Symposium Series, Machine Learning in Computer Vision: What, Why and How? (1993)
2. Beymer, D., Flickner, M.: Eye gaze tracking using an active stereo head. In: CVPR 2003, June 2003, vol. 2, pp. 451–458 (2003)
3. Freedman, E.G.: Interactions between eye and head control signals can account for movement kinematics. Biol. Cybern. 84(6), 453–462 (2001)
4. Freedman, E.G., Sparks, D.L.: Coordination of the eyes and head: movement kinematics. Experimental Brain Research 131(1), 22–32 (2000)
5. Guestrin, E.D., Eizenman, M.: General theory of remote gaze estimation using the pupil center and corneal reflections. IEEE Trans. on B.E. 53(6) (2003)
6. Hanes, D.A., McCollum, G.: Variables contributing to the coordination of rapid eye/head gaze shifts. Biol. Cybern. 94(4), 300–324 (2006)
7. Heinzmann, J., Zelinsky, E.: 3-d facial pose and gaze point estimation using a robust real-time tracking paradigm. In: FGR 1998, pp. 142–147 (1998)
8. Kinoshita, K., Ma, Y., Lao, S., Kawaade, M.: A fast and robust 3d head pose and gaze estimation system. In: ICMI 2006, pp. 137–138 (2006)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
10. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. MRL Tech. Report, Intel Labs. (2002)
11. Matthews, I., Ishikawa, T., Baker, S., Kanade, T.: Passive driver gaze tracking with active appearance models. Technical Report CMU-RI-TR-04-08, Robotics Institute, Pittsburgh, PA (February 2004)
12. Wang, J.-G., Sung, E., Venkateswarlu, R.: Eye gaze estimation from a single image of one eye. In: ICCV 2003, October 2003, vol. 1, pp. 136–143 (2003)
13. Yamazoe, H., Utsumi, A., Yonezawa, T., Abe, S.: Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In: ETRA 2008, pp. 245–250 (2008)
14. Yoo, D.H., Lee, B.R., Chung, M.J.: Non-contact eye gaze tracking system by mapping of corneal reflections. In: FGR 2002, p. 101 (2002)
15. Zhu, J., Yang, J.: Subpixel eye gaze tracking. In: FGR 2002, May 2002, pp. 124–129 (2002)
16. Zhu, Z., Ji, Q.: Eye gaze tracking under natural head movements. In: CVPR 2005, pp. 918–923 (2005)

Orientation and Scale Invariant Kernel-Based Object Tracking with Probabilistic Emphasizing

Kwang Moo Yi, Soo Wan Kim, and Jin Young Choi

ASRI, PIRC, Dept. of Electrical Engineering and Computer Science,
Seoul National University, Seoul, Korea
{kmyi,swkim,jychoi}@neuro.snu.ac.kr

Abstract. Tracking object with complex movements and background clutter is a challenging problem. The widely used mean-shift algorithm shows unsatisfactory results in such situations. To solve this problem, we propose a new mean-shift based tracking algorithm. Our method is consisted of three parts. First, a new objective function for mean-shift is proposed to handle background clutter problems. Second, orientation estimation method is proposed to extend the dimension of trackable movements. Third, a method using a new scale descriptor is proposed to adapt to scale changes of the object. To demonstrate the effectiveness of our method, we tested with several image sequences. Our algorithm is shown to be robust to background clutter and is able to track complex movements very accurately even in shaky scenarios.

1 Introduction

Tracking of objects using the mean-shift algorithm is a popular method in the field of object tracking. The algorithm has its advantages in the fact that it is relatively easy to implement, it does not require heavy computation, and it shows robust results in practical object tracking tasks. However, the original mean-shift algorithm shows unsatisfactory results when the object shows complicate movements and there are objects similar to the target in the nearby background region. This is due to the three major problems of the original mean-shift algorithm. The first problem is the background clutter effect during mean-shift iterations, which may lead to tracking failures. The second problem is the lack of ability to track elaborate movements such as in-plane rotation. The third problem is its inability to adapt to scale changes (i.e. kernel bandwidth is fixed), which is critical to the tracking performance. These problems greatly affect object tracking results, but are not clearly solved.

The first problem was usually approached with the use of anisotropic kernels such as ellipsoids [1]. This method reduces the amount of background information in the object model but still contains background pixels inside the model which causes background clutter problems. A. Yilmaz proposed a method using level-set kernels which are in the exact shape of the object [2]. This level-set kernel does not have any restriction in shape and succeeded in exclusion of the background information inside the object model. Unfortunately, even when using

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 130–139, 2010.

these kernels, background information is still included inside the target candidate. R. Collins et.al. used a way of selecting discriminative features online to overcome the effect of the background information [3]. Their method succeeded in obtaining features which make the target object more discriminative to the background. But as they noted in their paper, the number of features to be selected is not certain. Moreover, the computation time increases proportional to the number of selected features. The second problem has not been covered much in the field of mean-shift object tracking since only translational movements can be estimated through the mean-shift vector. Rather, to track elaborate movements, other famous tracking algorithms such as the *"Particle Filter"* are used [4], [5], [6], [7]. However, for very complex movements, tracking using particle filter is hard to be done in real-time. Other silhouette tracking methods, such as tracking via direct minimization of contour energy function [8], are also capable of tracking elaborate movements, but require even more computation. The third problem was intuitively solved in [1] by the 10% method, but this method does not work well due to its nature of preferring the smaller kernel. R. Collins proposed a method using difference of Gaussian (DOG) mean-shift kernel in scale space [10] to solve this problem. However, this method is computationally expensive. C. Yang et.al. [11] succeeded in tracking objects with scale changes using the joint feature-spatial space concept [12], but their method adapts to scale changes without consideration of the regional changes of the template. Yi et.al. seemed to solve all three of these problems [14] but as they noted, estimation results are somewhat unstable.

In this paper, to overcome the three problems of mean-shift, we propose a new mean-shift based object tracking method. Our method is consisted of three parts. First, we propose an altered objective function for mean-shift, which makes the tracker robust to background clutter. Second, we propose an orientation estimation method to track objects with in-plane rotation. Third, we propose a method which utilizes a new scale descriptor to adapt to scale changes of the object. The test results show that the proposed algorithm is superior to the original mean-shift algorithm and is also comparable to another popular tracking algorithm, the particle filter.

The paper is organized as follows. Section 2 briefly describes the original mean-shift algorithm for reference. Next, the proposed method is explained in detail in section 3. Experimental results of our proposed algorithm are given in section 4 and finally, we will conclude our paper on section 5.

2 Mean Shift Tracking: Brief Review

In this section, we give a brief review of the original mean-shift algorithm [1]. The mean-shift method is a fast way of finding the local maxima of a sample distribution iteratively from a given starting position. In the field of object tracking, this sample is the color observed at a pixel $\mathbf{x}$. To this $\mathbf{x}$, the sample weight $w(\mathbf{x})$ is defined as

$$w(\mathbf{x}) = \sqrt{h_m(I(\mathbf{x}))/h_c(I(\mathbf{x}))}, \tag{1}$$

where $I(\mathbf{x})$ is the color of pixel $\mathbf{x}$, h_m and h_c are the color histograms generated from the model and candidate object regions, respectively. If we let the initial hypothesized position be $\hat{\mathbf{y}}_{old}$, the computed new position be $\hat{\mathbf{y}}_{new}$, the pixels inside the candidate region be $\mathbf{x}_i$, $\Delta\mathbf{y} = \hat{\mathbf{y}}_{new} - \hat{\mathbf{y}}_{old}$, and $K(.)$ be the radially symmetric kernel defining the tracking object region respectively, then using the sample weight (1), the mean shift vector is computed as

$$\Delta\mathbf{y} = \frac{\sum_i K(\mathbf{x}_i - \hat{\mathbf{y}}_{old})w(\mathbf{x}_i)(\mathbf{x}_i - \hat{\mathbf{y}}_{old})}{\sum_i K(\mathbf{x}_i - \hat{\mathbf{y}}_{old})w(\mathbf{x}_i)}. \tag{2}$$

This mean shift vector is an estimate of the gradient of the sample distribution. Using this mean shift vector, tracking of the object is performed iteratively.

$w(\mathbf{x}_i)$ in (1) is derived from the Taylor expansion of the Bhattacharyya coefficient used in [1]. Bhattacharyya coefficient is defined as $\rho(\mathbf{y}) \equiv \rho[p_c(\mathbf{y}), p_m] = \int \sqrt{p_{c_\mathbf{z}}(\mathbf{y}) p_{m_\mathbf{z}}} d\mathbf{z}$, where $p_c(\mathbf{y})$ denotes the probability distribution of the candidate when $\mathbf{y}$ is the center of the candidate, p_m denotes the probability distribution of the object model, and $\mathbf{z}$ denotes that it is of some feature. In the our case, we use color histograms as features, therefore the estimate for the Bhattacharyya coefficient can be defined as

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{p_c}(\mathbf{y}), \hat{p_m}] = \sum_\nu \sqrt{h_c(\nu, \mathbf{y}) h_m(\nu)}, \tag{3}$$

where $\hat{}$ denotes the estimator and $h_c(\nu, \mathbf{y})$ is the histogram value for color ν of the candidate when candidate is at position $\mathbf{y}$. Using Taylor expansion around some point $\mathbf{y}_0$ and kernel estimation, this equation can be approximated as follow [1]:

$$\begin{aligned} \rho[\hat{p_c}(\mathbf{y}), \hat{p_m}] \approx & \frac{1}{2} \sum_\nu \sqrt{h_c(\nu, \mathbf{y}_0) h_m(\nu)} \\ & + \frac{C_h}{2} \sum_i w(\mathbf{x}_i) K(\mathbf{y} - \mathbf{x}_i), \end{aligned} \tag{4}$$

where C_h denotes a normalizing constant.

3 The Proposed Method

3.1 Probabilistic Emphasizing

To solve the background clutter problem, we propose an altered objective function for mean-shift. Using this objective function emphasizes features that are more likely to be of the target object rather than the background. Generally, in mean-shift, color histograms are used as features to describe an object. Therefore, in our work, we also used the probability distribution of colors, i.e. color histograms, as features. We first start by obtaining the probability of some color ν being in the object model. If we denote the probability of being in the object model as $p(Obj)$, the probability of being in the model for ν can be denoted as

$p(Obj|\nu)$. This $p(Obj|\nu)$ can be interpreted as the probability of a pixel with color ν being in the object model, i.e. $1 - p(Obj|\nu)$ is the probability of that pixel being in the background. Then, using the Bayesian rule, $p(Obj|\nu)$ can be obtained by $p(Obj|\nu) = p(\nu|Obj)p(Obj)/p(\nu)$. Here, $p(\nu|Obj)$ can be estimated with the color histogram of the object model, $p(Obj)$ with the area ratio of the object region and the selected background region, and $p(\nu)$ with the color histogram of the background and object region. Therefore if we denote the color histogram of the background region as $h_{bg}(.)$, area of the candidate region as A_c, and the area of the background region as A_{BG}, respectively, Then, we can write the estimate of $p(Obj|\nu)$

$$\hat{p}(Obj|\nu) = \frac{A_c h_m(\nu)}{A_{BG} h_{BG}(\nu) + A_c h_m(\nu)}. \tag{5}$$

Next, we use some function of (5) as a penalty function [13] for finding the maxima of $\hat{\rho}(\mathbf{y})$ in (3). If we denote this penalty function as $\Phi(\nu)$, then, (3) can be modified as

$$\tilde{\rho}(\mathbf{y}) \equiv \tilde{\rho}[\hat{p_c}(\mathbf{y}), \hat{p_m}] = \sum_{\nu} \Phi(\nu) \sqrt{h_c(\nu, \mathbf{y}) h_m(\nu)}. \tag{6}$$

In our work, we used $\Phi(\nu) = \sqrt{\hat{p}(Obj|\nu)}$. Then, if we denote $\Phi(I(\mathbf{x}_i))w(\mathbf{x}_i)$ as $\tilde{w}(\mathbf{x}_i)$, the mean-shift equation (2) simply becomes

$$\Delta\tilde{\mathbf{y}} = \frac{\sum_i K(\mathbf{x}_i - \hat{\mathbf{y}}_{old})\tilde{w}(\mathbf{x}_i)(\mathbf{x}_i - \hat{\mathbf{y}}_{old})}{\sum_i K(\mathbf{x}_i - \hat{\mathbf{y}}_{old})\tilde{w}(\mathbf{x}_i)}. \tag{7}$$

This proposed objective function emphasizes the weights from pixels that are more likely to be in the object model than the background. Therefore the tracker tends to follow features that are more discriminant from the background, i.e. the tracker becomes more robust to background clutter problems.

3.2 Orientation Estimation

Our proposed method for orientation estimation uses color histograms constructed for each orientation divisions as in [14]. Within this sub-section, this *"orientation division"* terminology will be used often. Therefore we will first start by clearly defining this orientation division concept. If we denote $\mathbf{\Omega}$ as the object (or the candidate) region, $\mathbf{x}_c$ as the center of the object (or the candidate) region, and N_α as the number of orientation divisions, respectively, we can define the orientation divisions as

$$\alpha_i \triangleq \{\mathbf{x}_i | F_1(\arg(\mathbf{x} - \mathbf{x}_c)) \in [\eta_i, \eta_{i+1}), \mathbf{x} \in \mathbf{\Omega}\}, \tag{8}$$

where, $F_1(.)$ is a function to restrict the value of $\arg(\mathbf{x}_i - \mathbf{x}_c)$ to be in $[-\pi/2, \pi/2)$ and η_i is the boundary for each orientation division, respectively.

At the last steps of the mean-shift iteration, when tracking of the translation of the object is almost finished, most of the object is likely to be inside the

tracking window, and also since the time difference is very small between frames, the orientation of the object is likely to change little. This allows the assumption that color distribution of the the target candidate has not changed much from the object model in the last steps of the mean shift iteration. If we let p_m and p_c be the probability with respect to the object model and the target candidate respectively, under this assumption we can assume $\hat{p}_c(\alpha_i|\nu) \approx p_m(\alpha_i|\nu)$ [14], where $p_c(\alpha_i|\nu)$ and $p_m(\alpha_i|\nu)$ denotes the probability of color ν being in ith orientation division of the candidate and the model, and $\hat{}$ denotes the estimator, respectively. From this approximation, we can derive

$$\hat{p}_c(\alpha_j|\alpha_i) = \sum_{\nu} \hat{p}_c(\alpha_j|\nu) p_c(\nu|\alpha_i) \tag{9}$$

$$\approx \sum_{\nu} p_m(\alpha_j|\nu) p_c(\nu|\alpha_i). \tag{10}$$

This $\hat{p}_c(\alpha_j|\alpha_i)$ is the probability of each orientation division α_i being the orientation division α_j, i.e. this lets us know what each orientation division is likely to be. In (10), $p_m(\alpha_j|\nu)$, can be obtained by

$$p_m(\alpha_j|\nu) = \frac{p_m(\nu|\alpha_j) p_m(\alpha_j)}{\sum_j p_m(\nu|\alpha_j) p_m(\alpha_j)}. \tag{11}$$

The probability of color values for each α_j, $p_m(\nu|\alpha_j)$, can be calculated using color histograms constructed for each orientation division, and $p_m(\alpha_j)$ is just the area ratio of α_j and the object region with respect to the object model. If we denote the old estimated orientation of the tracker as $\hat{\theta}_{old}$, newly estimated orientation as $\hat{\theta}_{new}$, $\Delta\hat{\theta} = \hat{\theta}_{new} - \hat{\theta}_{old}$, and mean of orientation of α_i and α_j as θ_i and θ_j, respectively, using the results of (10) and (11), we can obtain $\Delta\hat{\theta}$ by

$$\Delta\hat{\theta} = \sum_i \left[\sum_j \hat{p}_c(\alpha_j|\alpha_i) F_2(\theta_j - \theta_i) \right] p_c(\alpha_i), \tag{12}$$

where, $p_c(\alpha_i)$ is the area ratio of α_i and the object region with respect to the target candidate, and function $F_2(.)$ is to enforce $\theta_j - \theta_i$ to be inside $[-\pi/2, \pi/2)$. Since from our definition of orientation divisions in (8), $|\theta_i| < \pi/2$ and $|\theta_j| < \pi/2$. Next, to make our orientation estimation result robust to background clutter, we use result of (5) to modify $p_m(.)$ and $p_c(.)$ in (10) and (11). Instead of using $p_m(\nu|\alpha_j)$ and $p_c(\nu|\alpha_i)$ we use $\tilde{p}_m(\nu|\alpha_j)$ and $\tilde{p}_c(\nu|\alpha_i)$ which are modified as

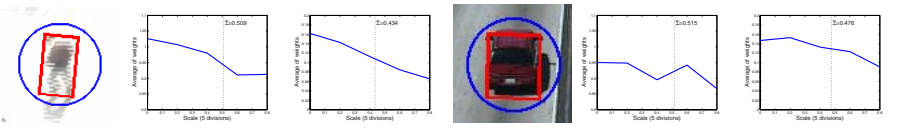
$$\tilde{p_k}(\nu|\alpha_j) \triangleq \frac{p_k(\nu|\alpha_j)\hat{p}(Obj|\nu)}{\sum_\nu p_k(\nu|\alpha_j)\hat{p}(Obj|\nu)}, k \in \{m, c\}. \tag{13}$$

Substituting (13) in (10), (11), and (12), we obtain the final equation

$$\Delta\tilde{\theta} = \sum_i \left[\sum_j \tilde{p}_c(\alpha_j|\alpha_i) F_2(\theta_j - \theta_i) \right] p_c(\alpha_i) \tag{14}$$

3.3 Scale Adaptation

To adapt to scale changes, we need to know what the current status of our tracker is, i.e. what the target candidate is looking at. If we could figure out which part of the distribution our target candidate is observing, then the whole problem of scale adaptation would be easily solved. Roughly speaking, our distribution of weights must be similar to the shape of the kernel we used for mean-shift: lager values if closer to the center and smaller values if further from center. This is due to the nature that when doing mean-shift, we adopt a kernel to estimate the probabilistic distribution for tracking, and therefore color histograms are constructed with larger values if closer to the center. We confirmed this by experimental data for some actual tracking situations (Figure 1).



(a) (b) (c) (d) (e) (f)

Fig. 1. Example of scale vs average of weights. Scale divided into 5 scale divisions. (a) and (d) are the original image, where inner red box denotes the target candidate region and the outer blue circle denotes the background region. (b) and (e) are the results using $w(\mathbf{x}_i)$, and (c) and (f) are the results using $\tilde{w}(\mathbf{x}_i)$.

To use this idea, we first start by defining the scale divisions of a kernel. if we denote the relative distance as $\sigma(\mathbf{x})$ (ranging from 0 to 1), following the same notation for the object (or the candidate) region from sub-section 3.2, we can define the scale division as $\varsigma_i \triangleq \{\mathbf{x}|\sigma(\mathbf{x}) \in [\zeta_i, \zeta_{i+1}), \mathbf{x} \in \Omega\}$, where ζ_i is given by $\zeta_i \triangleq (i-1)/N_\varsigma$ and N_ς is the number of scale divisions. Then, to observe which part of the original weight distribution respect to scale the target candidate is looking at, we can define the following descriptor for scale [14]:

$$\Sigma = \sum_j \left[\frac{w_{avg,j}}{\sum_i w_{avg,i}} \sigma_{avg,j} \right], \tag{15}$$

where $w_{avg,j} = \frac{1}{N_{\varsigma_j}} \sum_{\mathbf{x}_i \in \varsigma_j} w(\mathbf{x}_i)$, $\sigma_{avg,j} = \frac{1}{N_{\varsigma_j}} \sum_{\mathbf{x}_i \in \varsigma_j} \sigma(\mathbf{x}_i)$, and N_{ς_j} is the number of pixels inside ς_j. Since in consecutive frames the change in scale is little, this descriptor is sufficient for describing how the distribution of weights has changed. However, this Σ is may be inaccurate due to inclusion of the background information when obtaining $w(\mathbf{x}_i)$. To overcome this limitation, we use $\tilde{w}(\mathbf{x}_i)$ instead of using plane $w(\mathbf{x}_i)$. This give us the final equation for our newly proposed scale descriptor

$$\tilde{\Sigma} = \sum_j \left[\frac{\tilde{w}_{avg,j}}{\sum_i \tilde{w}_{avg,i}} \sigma_{avg,j} \right]. \tag{16}$$

Using this descriptor, we adapt scale descriptor of the current target $\tilde{\Sigma}_{candidate}$ to match the initial scale descriptor of the model $\tilde{\Sigma}_0$. By this adaptation, we can tracks object with scale without much increase in computation time.

3.4 Algorithm Summary

When tracking objects, mean-shift finds the most probable position of the target object through iteration. During this iteration, when the target candidate is moving Thus, our method estimates orientation and adapts to scale only when the target candidate is moving in small amounts, i.e. when $||\Delta\tilde{\mathbf{y}}||$ is smaller than some threshold ϵ'.

Given the object model $\mathbf{q}$ (the kernel, the color histogram of the model, the color histograms constructed for each orientation divisions, and $\tilde{\Sigma}_0$), the tracking algorithm can be summarized as follow:

Algorithm 1. Tracking

1: Create the target candidate model $\mathbf{p}$
2: Compute the $\Delta\tilde{\mathbf{y}}$ using $\mathbf{q}$ (7)
3: $\mathbf{y}_{new} \leftarrow \mathbf{y}_{old} + \Delta\tilde{\mathbf{y}}$
4: If $||\Delta\tilde{\mathbf{y}}|| > \epsilon'$ go to 1.
5: $\sigma_{new} \leftarrow \frac{\tilde{\Sigma}_{candidate}}{\tilde{\Sigma}_0}\sigma_{old}$ (16)
6: $\theta_{new} \leftarrow \theta_{old} + \Delta\tilde{\theta}$ (12)
7: Repeat steps 1 to 6 until $||\Delta\tilde{\mathbf{y}}|| < \epsilon''$,

where ϵ' is the threshold for orientation estimation and scale adaptation and ϵ'' is the threshold for convergence.

4 Experiments

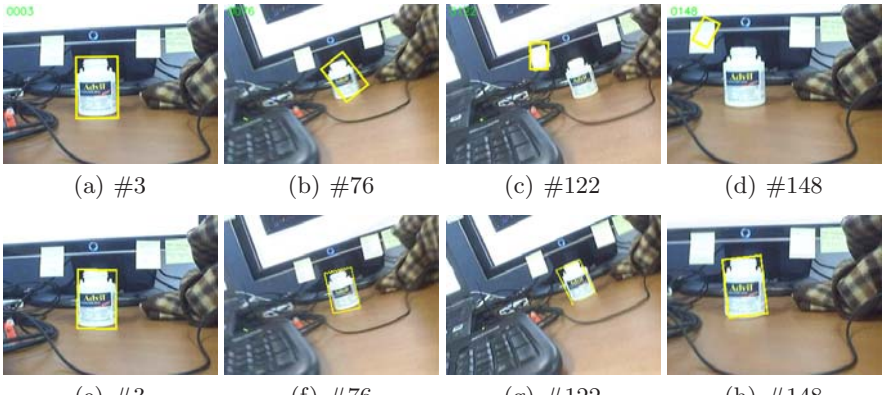
Our algorithm was implemented using C++ with $16 \times 16 \times 16$ RGB color histogram, 5 orientation divisions, 4 scale divisions, 2 for ϵ' and 0.9 for ϵ''. An anisotropic kernel in the shape of a rectangle was used for kernel density estimation in mean-shift. For the background region, we used the area inside a circle little bigger than our target candidate. In actual tracking scenarios, orientation changes are not large in consecutive frames, therefore, we clipped the orientation estimation result to -2.5˚and 2.5˚. This is to prevent erroneous estimation results for orientation estimation since our assumption in sub-section 3.2 holds only for small orientation changes. All experiments were held on a 2.0GHz PC and ran comfortably over 90fps.

Figure 2 is the tracking results for one selected frame of an image sequence of cars moving on a highway. Using this image sequence, we compared the proposed algorithm with the original mean-shift and mean-shift using 10% scale adaptation. The trackers were applied to the car on the top right. The original

(a) (b) (c)

Fig. 2. Result of the original mean shift tracker (a), original mean shift tracker using the 10% scale adaptation (b), and the proposed method (c)

mean shift algorithm without scale adaptation (a) resulted in tracking failure, since it could not adapt to scale change and another similar object entered the target candidate region. Mean-shift with the 10% method (b) failed to adapt to scale change and the tracker shrank to a small box. Our method (c) on the bottom shows some minor errors in following the orientation of the object due to the drastic change in scale and minor change in viewpoint, but succeeded in following the target object.



(a) #3 (b) #76 (c) #122 (d) #148

(e) #3 (f) #76 (g) #122 (h) #148

Fig. 3. Tracking results for the IVT tracker (without subspace update) with 80 particles (above) and the proposed method (below)

Figure 3 is the tracking result of the proposed method compared with a particle filter tracker on an shaky image sequence of a medicine bottle captured by a hand held webcam. In the image sequence, the medicine bottle shows abrupt translational, orientational, and random movements, i.e. sudden change in scale and orientation occur. Sub-figures (a), (b), (c), and (d) are the results of the IVT tracker [6] with 80 particles without subspace update and (e), (f), (g), and (h) are the results of the proposed method. Each sub-caption denotes the frame numbers in the image sequence. The IVT algorithm (a particle filter with eigen-space method combined) was used for comparison. The reason we used 80

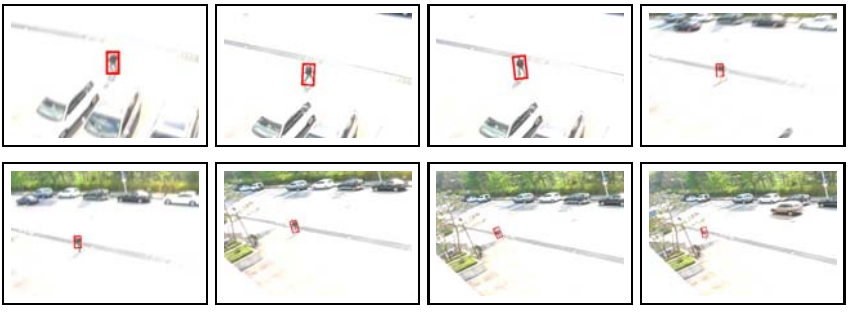


Fig. 4. Tracking of a person in a shaky scenario

particles is to achieve real-time performance (over 20fps) and compare it with our proposed method. Also, since with the subspace update, the IVT tracker was never able to follow no matter how many particles were used, we did not use the subspace update method. As shown in (b) and (f), frame 76, the IVT tracker fails to adapt to fast orientation and translation change, whereas the proposed method succeeds. In (c), frame 122, the IVT tracker fails to track and follows an object similar to the target. But in (g), since our method is robust to background clutter problems, we can see that our method succeeds in tracking the medicine bottle. The IVT tracker without subspace update was able to follow the medicine bottle using 600 particles, but takes 4 fps whereas our method is over 90 fps (both implemented using C++).

We also tested our proposed method on a shaky scene recorded with a handheld digital camcorder. The tracking results for selected frames are given in Figure 4. The recorded video image is very shaky, and therefore scenes of some frames are blurred out. The forth selected frame in Figure 4 is an example of this situation. In the forth selected frame, it is hard to recognize the legs of the tracked person even with human eyes. In the tracking results, there are some frames which our proposed method fails to adapt to scale change. These frames have abrupt changes in the position of the person due to the shake of the camcorder. However, our proposed method successfully re-adapts to scale change and ultimately, does not loose track of the scale change of the object. Orientation estimation results in (d) show similar behavior as the scale adaptation result in the fact that it shows some errors when abrupt motion occurs. But this is not a common case and we can see that our method successfully follows the orientation change of the person.

5 Conclusion

We proposed a new object tracking method to solve the problems of the original mean-shift algorithm. The method is consisted of three parts. To handle background clutter problems, we proposed a new objective function which emphasizes features that are more likely to be of the object model. We also proposed

an orientation estimation method to track object with orientation changes. Finally, to adapt to scale changes of the object, we proposed a scale adaptation method which utilizes a new scale descriptor. Experimental results show that the proposed method was able to track objects with scale and orientation changes even in shaky scenarios. In comparison with other tracking algorithms, the proposed method was shown to be superior to the traditional mean-shift and also comparable to the particle filter.

Acknowledgment

This work has been supported by the Korean Ministry of Knowledge Economy, Samsung Techwin, and BK 21 program.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 564–575 (2003)
2. Yilmaz, A.: Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In: IEEE Conf. on Computer Vision and Pattern Recognition (2007)
3. Collins, R.T., Liu, Y., Leordeanu, M.: Online Selection of Discriminative Tracking Features. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
4. Isard, M., Blake, A.: CONDENSATION - Conditional density propagation for visual tracking. International Journal on Computer vision 29(1), 5–28 (1998)
5. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conf. on Computer Vision, vol. 1, pp. 661–675 (2002)
6. Lim, J., Ross, D., Lin, R.S., Yang, M.H.: Incremental learning for visual tracking. In: Advances in Neural Information Processing Systems (2004)
7. Nummiaro, K., Koller-Meier, E., Gool, L.: An adaptive color-based particle filter. Image and Vision Computing 21(1), 99–110 (2003)
8. Yilmaz, A., Li, X., Shah, M.: Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(11), 1531–1536 (2004)
9. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. ACM Computing Surveys 38(4) (2006), http://dx.doi.org/10.1145/1177352.1177355
10. Collins, R.: Mean-shift blob tracking through scale space. In: IEEE Conf. on Computer Vision and Pattern Recognition (2003)
11. Yang, C., Duraiswami, R., Davis, L.: Efficient Mean-Shift Tracking via a New Similarity Measure. In: IEEE Conf. on Computer Vision and Pattern Recognition (2005)
12. Elgammal, A., Duraiswami, R., Davis, L.: Probabilistic tracking in joint feature-spatial spaces. In: IEEE Conf. on Computer Vision and Pattern Recognition (2003)
13. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
14. Yi, K.M., Ahn, H.S., Choi, J.Y.: Orientation and Scale Invariant Mean Shift Using Object Mask-Based Kernel. In: Int'l Conf. on Pattern Recognition (2008)

Combining Edge and Color Features for Tracking Partially Occluded Humans

Mandar Dixit and K.S. Venkatesh

Computer Vision Lab.,
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur
mandarddixit@gmail.com, venkats@iitk.ac.in

Abstract. We propose an efficient approach for tracking humans in presence of severe occlusions through a combination of edge and color features. We implement a part based tracking paradigm to localize, accurately, the head, torso and the legs of a human target in successive frames. The Non-parametric color probability density estimates of these parts of the target are used to track them independently using mean shift. A robust edge matching algorithm, then, validates and refines the mean shift estimate of each part. The part based implementation of mean shift along with the novel edge matching algorithm ensures a reliable tracking of humans in upright pose through severe scene as well as inter-object occlusions. We use the CAVIAR Data Set as well as our own IIT Kanpur test cases demonstrating varying levels of occlusion in daily life situations to evaluate our tracking method.

1 Introduction

Detection and Tracking of moving objects is central to many computer vision applications such as visual surveillance, activity recognition, and human computer interaction. The most commonly used methods for detection record the changes occurring in the scene. A statistical model of the scene background is learned and an intruding object is detected as a group of connected pixels not well represented by this model. [4] uses a single Gaussian, whereas, in [6], a mixture of Gaussians is used to represent the background and observe changes against it. Our system for tracking uses the foreground segmentation algorithm proposed in [7] and implemented by OpenCV [14] library functions. This method performs segmentation through Bayesian decisions on selected features representing static and moving scene elements.

Once detected, the object must be tracked in different frames using its signature. Features such as color, shape and texture may be used to establish correspondence between the occurrences of the same object in successive frames. Based on the nature of implementation, the tracking algorithms can be divided into feature based, contour based and region based categories. A Feature based tracker described in [8] uses corner points to track vehicles through traffic congestion. Point features, however, may not provide reliable means of tracking people through appearance changes. Contour based tracking demonstrated in [9] captures the shape information of the object. But these methods are generally slower than region based approaches. Among region

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 140–149, 2010.

based tracking systems, the Kernel based mean shift tracker by Comaniciu et al [1] is well known. Given the target density estimate, the mean shift algorithm converges at the nearest mode of the point sample distribution represented by the test image. Although mean shift tracking provides accurate localization of an isolated object over short intervals of time, its performance degrades in the event of occlusion or change in the object scale or appearance. The color histogram used by the tracker changes substantially when the person being tracked turns about the vertical axis or is partially occluded by static or dynamic scene elements. A proposed solution to the problem of occlusion is the idea of coordinated tracking of multiple parts of the same target. Fragment based tracking (Frag-Track) proposed by Adam et al. [3] uses a template of fragments to track an agent through scale changes and partial occlusions. Frag-Track performs well in severe occlusions but the method uses a rigid template to describe a semi-rigid human body.

In this paper, we propose an approach to tracking that is a combination of region based and feature based paradigms. We use an efficient algorithm of matching local edges in conjunction with the Kernel based mean shift algorithm to obtain an accurate localization and track of humans in difficult scenarios. In order to use a more spatially descriptive appearance model for mean shift, we initialize three independent mean shift trackers for the head, torso and legs of the individual to be tracked. The part based approach ensures a better confidence through various levels of scene occlusion than the overall (single) mean shift. Following the mean shift cycles, the edge matching step validates and refines the mean shift estimates. Efficient tracking of people is achieved through coordinated mean shift and robust edge matching even in cases of severe occlusion, which is the key contribution of our research.

The rest of the paper is organized as follows. Section 2 summarizes some related research efforts to solve similar problems. Section 3 outlines our approach of edge-color tracking. Results of tracking in various complex situations are demonstrated in Section 4. In Section 5 we discuss the limitations of our method and future scope.

2 Related Work

Extensive research is being carried out in order to develop a system for tracking a moving target in a complex dynamic environment. One of the most well-known algorithms for object tracking is the kernel based mean shift proposed by Comaniciu et al. in [1]. The main advantages of Mean Shift algorithm are its speed of operation and accuracy of localizing moving targets. One of the drawbacks of this technique, however, is the lack of adaptability to scale changes. This problem has been addressed in detail in [5] and a scale invariant mean shift tracking procedure has been proposed as a possible solution. The author uses mean shift in spatial as well as scale dimensions to obtain an accurate localization and scale of the target. Zivkovik et al. propose a modifed mean shift procedure in [11] to include both scale and orientation changes by defning five degrees of freedom for the kernel.

Although kernel based mean shift is an effective region based algorithm for tracking isolated objects, its localization degrades in presence of occlusions and clutter. It is, therefore, not accurate enough for seamlessly following a moving target in complicated environments. Instead of using a single model for the entire object as in mean

shift tracking [1], a more descriptive part based or fragment based representation of the target is being preferred for better results in crowded scenes with frequent occlusions. Elgammal and Davis [2] propose that a person can be represented as a set of color regions located along the vertical axis. A person in upright pose is modeled as a collection of parts namely head, torso and legs, each having a separate color density representation. Along with color information, the spatial distribution of these parts is also included in the appearance model. In [3], the authors have used a fixed rectangular grid of patches with an intensity histogram of each patch to capture the spatial details along with the photometric information. The algorithm proceeds by finding the best matches of each fragment from the grid in the local neighborhood. Based on the similarity measures of the patches, a voting scheme is implemented to find an accurate estimate of the template center. To handle partial occlusions, Shakunaga et al. [10] propose an interesting technique that uses spatial information in a particle filtering framework. Their model represents a human using three ellipses, one each for head, torso and legs. Trained model based algorithms have also been developed for detection and tracking of humans in presence of persistent occlusions. Zhao and Nevatia [12] use a part based human model to solve a multiple hypothesis association problem. Efficient optimization in a joint hypothesis space is achieved using Markov Chain Monte Carlo method. Wu and Nevatia [13] propose a hierarchical part based model for detection and tracking of partially occluded people through trajectory estimation. Edgelet features from different parts of human body are used to train the model. The overlapping scores of detected part edges with the overall target segmentation are used to attribute part responses to a human hypothesis.

The proposed approach in this paper is different from the above mentioned ones in various respects. Our method uses the part based approach to track an object in an upright pose using a combination of both mean shift and edge tracking. Achieving better results by efficiently using a combination of features is the novelty of our approach. We implement mean shift tracking for head, torso and legs of the target independently. Our part based model is more flexible than in [3] as it does not impose a rigidness constraint on a semi-rigid human body. After convergence of each mean shift part tracker, an efficient edge matching algorithm validates and refines the estimate. Model based detection methods [12, 13] require extensive on-line learning. Instead we use background subtraction method to detect the target and learn mean shift kernels for each part. The Canny edge detection algorithm is used to extract strong edges of the target. The edge tracker uses information regarding curvatures and relative locations of stable object edges to match them over successive frames.

3 Proposed Approach

We propose a part based approach to track humans in an upright pose using a combination of edge and color information. The detected human silhouette is segmented to obtain the head, torso and legs and independent mean shift trackers are initialized for each part. An iterative edge matching step follows each mean shift cycle. The parts with high edge confidence indicate an accurate estimate. The mean shift trackers that diverge from their targets are thus separated from the faithful ones. Combined confidence of edge matching and color based mean shift helps in tracking the target

through severe occlusions with impressive localization. The steps involved in the proposed method are explained in detail in the following sections.

3.1 Appearance Model

As mentioned earlier, the lack of spatial information in the appearance model of a mean shift tracker can be remedied to some extent by employing a part based technique. To implement a part based variant of the traditional mean shift tracker, we segment the incoming background subtracted silhouette for parts namely, head, torso and legs. We search for the pronounced valleys in the horizontal projection of the foreground silhouette which mark the junctions between the parts we seek to locate. When the sensor optical axis is almost horizontal, the valley corresponding to head and torso junction can be located at a height 0.6 to 0.8 times the height of a reasonably good foreground blob, whereas the valley corresponding to torso and leg junction can be located at a height 0.3 to 0.5 times that of the blob. The blob segmentation method is demonstrated on a test image in Fig. 1. The non-parametric color probability densities for individual parts are then learned for their individual mean shift trackers.

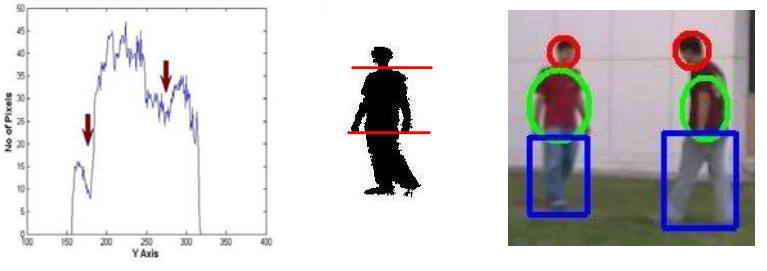


Fig. 1. Segmentation of blob along valleys in vertical projection

Although a part based appearance model ensures that at least one of the trackers follows the target accurately, the deviant behavior of a mean shift tracker in clutter or due to occlusion necessitates a method to verify the credibility of the individual part mean shift estimates. As a result, we supplement the color information of the appearance model with the local edge information which defines both shape and texture of a human target. Strong edges in the region of the foreground blob are obtained from the image using Canny edge detection algorithm implemented in OpenCV library functions. These learnt edges would then be matched with those extracted in following frames using their positions and curvature features. The edges obtained from a human image are relatively less stable when compared to those of rigid objects. Nevertheless, the change in their curvature and location is gradual enough to enable matching over a short interval of frames after which the template has to be reinitialized.

3.2 Mean Shift Part Tracking

As mentioned earlier, we use independent mean shift trackers to follow the head, torso and legs of a person. We use the Epanechnikov kernel to find the density

estimate of RGB color values of the segmented pixels. The probability of a color u as expressed by the kernel density function can be written as:

$$P_M(u) = C \sum k(\|x_i / h\|)\delta[b(x_i) - u], \quad u = 1, 2, \ldots, M \tag{1}$$

Where M denotes the number of histogram bins, x_i denotes the pixel location and k(.) denotes the profile function of the kernel. For details of mean shift tracking algorithm the readers are referred to [1].

The main drawback of mean shift tracking is the drift of the tracker due to occlusion and clutter. Fig. 2 shows the effect of scene occlusions on the mean shift algorithm. Such divergence of the mean shift tracker may cause complete track loss. Our methods prevent the degradation in performance due to occlusion through the use of a part based model. The trackers corresponding to un-occluded portions of the target maintain proper track throughout and thus, prevent incorrect localization as would happen in the case of traditional mean shift tracking.

Scale Handling

The Mean shift tracking algorithm does not account for scale changes of the target. The techniques of updating target scale proposed by [5, 11] are interesting but require heavy computations. We, instead, use a simple method to handle scale changes in our algorithm. The number of foreground pixels is a useful heuristic indicating change in the size of a target. Although, foreground blobs are not always reliable indicators of the shape and size of the object, we can identify scale changes since they cause blob size to alter gradually. We set thresholds on percentage changes in number of foreground pixels of the blob and reinitialize the mean shift kernel when the change stays within this threshold value. A blob change that exceeds the assigned threshold indicates either occlusion or improper segmentation, in the event of which, the mean shift kernel is kept unchanged. This simplified method provides satisfactory handling of the scale variation problem.

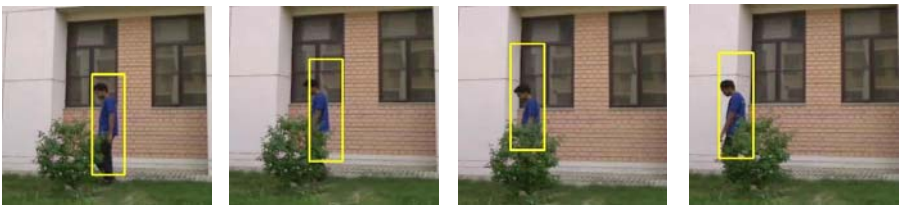


Fig. 2. The sequence of images demonstrates the divergence of mean shift tracker from the target in presence of scene occlusion

3.3 Edge Matching

We use a robust edge matching algorithm in conjunction with the part based mean shift to improve tracking performance in difficult and crowded situations. An edge can be identified by its location on the object and its curvature. We use both these

features to match the extracted edges with those in the learnt template. To capture the curvatures of an edge along its length, we model it using straight line segments of fixed lengths. The orientations of these straight line segments are recorded as an estimate of the edge curvatures. Smaller the length of these segments, greater is the accuracy of the estimate.

Although edges are a reliable feature for tracking, the edges of a moving human target change over time as against the edges of a rigid object. As a result, the edge template needs to be reinitialized when necessary. We use percentage change in the blob size as an indicator of changing scale and orientation (about the vertical axis) to reinitialize both the mean shift kernel and the edge template, by setting up an upper as well as lower threshold. The former is required to prevent spurious updates possibly caused by segmentation failures or occlusions.

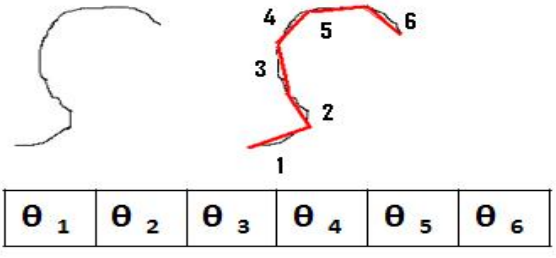


Fig. 3. A schematic showing edge approximation with straight line segments (in red). The vector $\{\theta_i\}$ indicates the orientations of these straight line segments (with respect to horizontal).

3.3.1 Algorithm for Edge Matching

An Edge e can be represented by a vector θ of orientations of the straight line segments approximating the edge, as shown in Fig. 3, and a representative point p located on it. Suppose we wish to calculate the matching score of an extracted edge e_m with a learnt edge e_t from the template. We denote their orientation vectors as θ_m and θ_t of lengths l_m and l_t respectively. Edges e_m and e_t may be same or different or one may be a part of the other. To verify a match between the two, we must locate the set of points common to the two edges. The straight line segments approximating the two edges that lie in this region of match have almost the same orientations. That is, one can ascertain a match between two edges by locating matching sections in the two orientation vectors. This can be depicted as sliding the orientation vector of one edge over another and matching the directions of overlapping sections. If the two edges match, the mean absolute orientation difference between the overlapping sections of their vectors would be negligible or zero. The problem of curvature matching between edges can, thus, be formulated as one of finding the minimum mean absolute orientation difference between their vectors and comparing it with a threshold.

$$\Theta(n) = \sum_k | \theta_t(k) - \theta_m(n+k) | / \Omega(n) \quad , n = -l_m + 1, -l_m + 2, \ldots , l_t - 1 \tag{2}$$

The term $\Omega(n)$ represents the overlap between two vectors θ_t and θ_m, corresponding to the value of index n, which indicates their relative positional displacements. The value of $\Omega(n)$ could be calculated as

$$\Omega(n) = \min(l_t, l_m) - n \tag{3}$$

When the match between the overlapping sections of the edges is perfect, the value of mean absolute orientation difference $\Theta(n)$ diminishes. This residual value is denoted as $\Theta_{m,t}$, the minimum difference between the curvatures of two edges e_t and e_m. The corresponding overlap $\Omega_{m,t}$ is the overlap of best curvature match between them.

$$\min_n \Theta(n) = \Theta(r) = \Theta_{m,t} \tag{4}$$

$$\Omega(r) = \Omega_{m,t} \tag{5}$$

Since different edges may have the same curvature, taking their location into consideration is of prime importance. After matching the edge curvatures, we find the midpoints of the overlapping parts of both edges. Suppose *p* and *q* denote the midpoints of overlapping sections of the edges. (Note: the coordinates of p indicate location of the point from the top-left corner of the mean shift window, whereas coordinates of q on the learnt edge indicate its location from the top-left corner of the template window. The proposed idea is that if a mean shift tracker maintains accurate track of an object, the relative locations of its edges on the target remain nearly the same over successive frames). The proximity of edges is indicated by the Euclidean distance between these two points. We modify the minimum difference metric of the edges to include this distance information and the value of edge overlap $\Omega_{m,t}$ to prevent false matches.

$$\gamma_{m,t} = \min_n \Theta(n) \times d(p, q) / \Omega_{m,t} \tag{6}$$

where $d(p, q)$ is the distance between the midpoints *p* and *q* of the overlapping parts of the edges. This, then, is the procedure and the metric devised to match two edges.

In case of a human target, after every mean shift cycle, the local edges present in every part mean shift window are matched with the edges of their respective edge templates. That is, we need to find pairs of edges e_m from the current frame and e_t from the learnt template, corresponding to a particular part, that exhibit a match with the value of their metric $\gamma_{m,t}$ lying within a predefined threshold T. Every such pair of matching edges would lie at a certain Euclidean distance $d(p, q)$ with respect to each other. If we reposition the mean shift window such that this distance between the query edge and the template edge diminishes, they would show a better match (lower value of $\gamma_{m,t}$). In other words, if the average of Euclidean distances $d(p, q)$ for all the pairs of matching edges is used to alter the mean shift window location, an overall improvement in the edge matching result would accrue. Such adjustment in the mean shift window may also result in newer pairs of matching edges. Hence, the process of edge matching (estimate validation) and window repositioning (estimate refinement) are carried out iteratively until a convergence is reached. The overall edge matching confidence in each match cycle is calculated as follows:

$$C = \sum_{m,t} \Omega_{m,t} \times \log (T/\gamma_{m,t}) \quad (7)$$

The sum is over all pairs of matching edges (m, t). Fig. 4 shows the results of edge matching algorithm on some human test images.

4(a) 4(b) 4(c) 4(d)

Fig. 4. (a) and (c) represent the learnt templates of head and torso edges respectively. (b) and (d) show the corresponding matched edges.

3.4 Part Assignment and Target Localization

Following the part mean shift cycle, the edges present in the mean shift windows are matched with the part edges present in their respective learnt edge templates. Parts being accurately tracked would exhibit a high Bhattacharya matching as well as edge matching confidence. Based on the dimensions and locations of such part trackers, an elliptical bound is derived to mark the target. In each frame, its dimensions and positions are updated based on the part tracking responses. If one of the trackers deviates due to clutter in the background or occlusion, it ceases to show a good edge matching confidence (C in eq. 7). If the confidence falls below a set threshold, we ignore the tracker completely and update the target marker position and dimensions solely based on the remaining faithful trackers. If both the torso and the leg trackers of the human target diverge, only the position of the ellipse is updated according to that of the head tracker and the dimensions are kept unchanged.

4 Results

The proposed algorithm was used to track humans through various events of occlusions as seen in Fig. 4. We use the standard CAVIAR Dataset (*http://homepages.inf.ed.ac.uk/rbf/CAVIAR/*) to evaluate our tracking algorithm. The first and second sequences shot in a corridor of a mall show the satisfactory performance of our algorithm in presence of almost 70-80% occlusion (head and a small part of torso visible). We also test our algorithm on our IIT Kanpur dataset demonstrating various scene occlusion scenarios. The third and fourth sequences show a person being occluded by a shrub and parked two-wheelers respectively. Our combined Edge-Color tracker (ECT) maintains accurate track throughout the event of occlusion in both sequences.

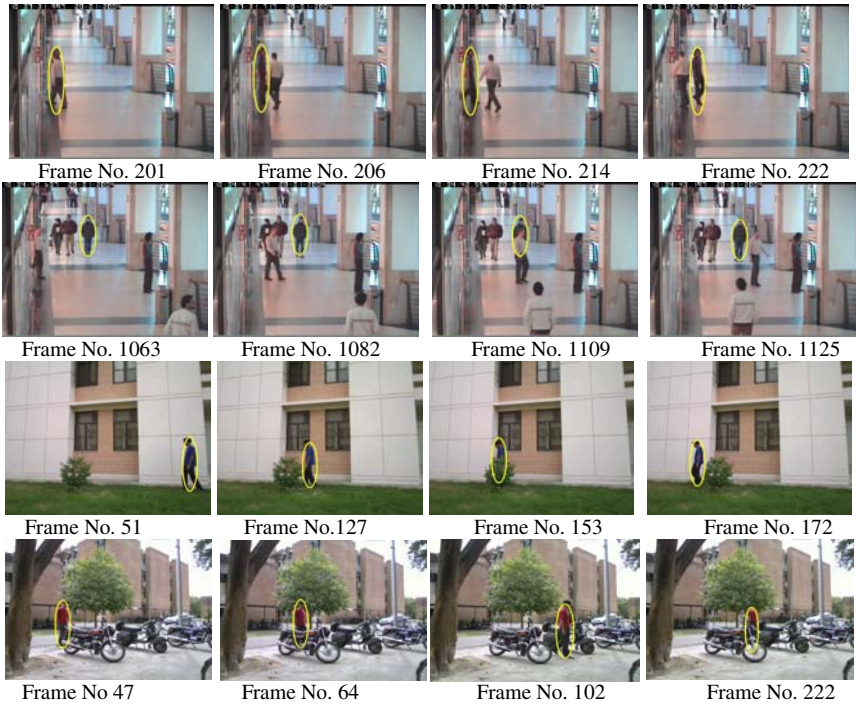


Fig. 5. Performance of ECT on CAVIAR test cases and IIT Kanpur Dataset

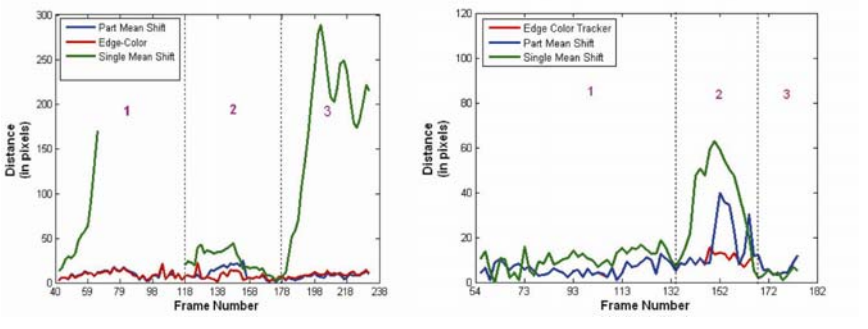


Fig. 6. Both the plots represent the difference (in pixels) between the tracker localizations and manually determined ground truth. Plot in Red indicates edge color tracker localization. Single mean shift tracker localization is shown in Green whereas a simple part mean shift performance is shown in Blue. The plot on the left shows the performance of both trackers in "parking space" sequence while the plot on right shows their performance on "shrub data". The discontinuity in the green line on the left plot shows that the mean shift tracker was lost and needed to be re-initialized while the edge color tracker kept good track.

Fig. 6 shows plots of localization errors of trackers with respect to manually marked ground truth for shrub and two-wheeler parking sequences of Fig. 5. A progressive improvement is seen from a single kernel mean shift (shown in green) through a simple part based mean shift tracker (blue) to the Edge Color Tracker (red). The edge matching algorithm provides means to verify the credibility of part mean shift trackers. As a result the localization of an ECT is more reliable than just a part mean shift tracker in presence of clutter or occlusion.

5 Conclusion and Future Work

In this paper, we have proposed a simple yet highly effective technique for tracking partially occluded humans in a standing/walking position using a combination of edge and color features. One of the drawbacks of our method is its failure to update scale during persistent occlusions. The scale adaptation approach we use requires the complete foreground blob of the object. Hence, scale cannot be updated during the event of occlusion. Incorporating scale invariance in a more reliable manner in the proposed tracking framework would be the focus of our future efforts.

References

[1] Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE Trans. PAMI 25(5) (2003)
[2] Elgammal, A.M., Davis, L.S.: Probabilistic Framework for Segmenting People under Occlusion. In: Proc. ICCV (2001)
[3] Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments based tracking using the integral histogram. In: CVPR 2006, pp. 798–805 (2006)
[4] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. on PAMI 19(7), 780–785 (1997)
[5] Collins, R.T.: Mean-shift blob tracking through scale space. In: IEEE CVPR, vol. 2, pp. 234–240 (2003)
[6] Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings CVPR, pp. 246–252 (1999)
[7] Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Foreground object detection from Videos containing complex background. In: ACM MM 2003 (2003)
[8] Beymer, D., McLauchlan, P., Coifman, B., Malik, J.: A real-time computer vision system for measuring traffic parameters. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (1997)
[9] Freedman, D., Zhang: Active contours for tracking distributions. IEEE Trans. Image Proc. 13(4), 518–527 (2004)
[10] Satake, J., Shakunaga, T.: Multiple target tracking by appearance-based condensation tracker using structure information. In: ICPR (2004)
[11] Zivkovic, Z., Krose, B.: An em-like algorithm for color histogram-based object tracking. In: IEEE CVPR (2004)
[12] Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. In: CVPR, vol. II, pp. 406–413 (2004a)
[13] Wu, B., Nevatia, R.: Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. In: IJCV 2007 (2007)
[14] `http://sourceforge.net/projects/opencvlibrary/`

Incremental Multi-view Face Tracking Based on General View Manifold

Wei Wei and Yanning Zhang

Shaanxi Key Laboratory of Speech & Image Information Processing, School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
weiwin1979@gmail.com, ynzhang@nwpu.edu.cn

Abstract. A novel incremental multi-view face tracking algorithm is proposed in the graphic model, which includes a general view manifold and specific incremental face model. We extend a general view manifold to the state-space model of face tracking to represent the view continuity and nonlinearity in the video data. Particularly, a global constraint on the overall appearance of the tracked multi-view faces is defined based on the point-to-manifold distance to avoid drifting. This novel face tracking model can successfully track faces under unseen views, and experimental results proved the new method is superior to two state-of-art algorithms for multi-view face tracking.

1 Introduction

Face tracking aims to detect faces in the video by utilizing the temporal continuity. It can be widely used in many fields, such as human-computer interaction, video surveillance, video communication and access control etc. [1]. However, the robust face tracking is a challenging topic because faces are non-rigid objects and the imaging condition is often influenced by the complex environment. Usually, it is not reasonable to request one person to stay still in strictly defined pose. Rich literature aims to build face models which can represent variations of the face appearance in videos, such as subspace-based tracking methods [2]-[6], pixel-based tracking algorithms[7], contour-based algorithms[8], [9], global statistics of color algorithms [10]. We focus on incremental subspace learning based multi-view/multi-pose face tracking.

Suppose the face representation model involves all prior information of the face variations, we can track faces effectively in a complicated environment without model updating. Unfortunately, the priors are not always available, which lead to the difficulties in constructing a robust object-specific face model. For example, Black *et. al.* proposed a robust eigen-tracking algorithm without updating for rigid and articulated objects [3]. The unseen information in the test image was regarded as "outlier", which are eliminated in face tracking. It is not suitable for the non-rigid objects tracking especially when the environment is changing. To build an object-specific face model, the online updating mechanism is involved in [2] and [4]. Brand proposed an incremental singular value decomposition method

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 150–159, 2010.

for incomplete data. However, the mean face of the feature space was intact [5], which was actually changed during tracking. So Ross *et. al.* used a sequential Karhunen-Loeve (SKL) method to update the eigenbasis as well as the mean face [4]. They also introduced an empirical forgetting factor to focus on the recently-acquired images.

In multi-view face tracking, view changing is strongly nonlinear which is caused by the head rotation and self-occlusion of faces. This nonlinearity is difficult to be coped with only one view basis. Thus a generic face appearance model containing the feature spaces of five views was constructed offline in [2]. The views were connected by a probability transition matrix. During tracking, the generic appearance model was updated into an object-specific one gradually. However, without the guide of the generic information, once drifting occurs, this method is prone to lose the tracking target. Therefore a hybrid generic and specific face model in the dynamic Bayesian network framework was proposed in [6]. The object-specific model is represented by a mixture of the basis obtained by online updated probabilistic principal component analysis. The generic face model is offline trained by AdaBoost algorithm. It contains five pose-based classifiers to verify the accuracy of the tracked faces before updating to avoid drifting. This model has a proper balance between generality and specificity.

As we all know, head rotation is continuous and has regular patterns. It is not easy to estimate the intermediate views accurately by discrete view subspaces. Thus we build a general and continuous view manifold to represent the nonlinear view variation [11]. And an object-specific model is built based on the feature space of discrete views. We embed the general view manifold and the object-specific face model in a probabilistic graphic model. The main contributions of the presented method include: (1) We extend a recently proposed general view manifold to the dynamic view estimation; (2) A novel online updating mechanism for the specific model is defined based on the point-to-manifold distance to avoid drifting.

The rest of this paper is organized as follows. The next section describes the multi-view face tracking model. In Section 3, we propose the view estimation based online updating mechanism. In Section 4, the online updating strategy is introduced. The experimental results are shown on three video databases in Section 5. Conclusions and future research lines are drawn in Section 6.

2 Model Description of Incremental Multi-view Face Tracking

In the tracking task, the face characteristic, such as position, scale, etc., determines the *state* of the object, which is denoted as X. The observation at time t is represented as I_t. Tracking is to estimate the state of I_t by maximizing the probability $p(X_t|I_t)$. From the Bayesian perspective, it is formulated as a posterior probability estimation problem as Formula (1). It is a *state-space* model which depicts the dynamic characteristic between the adjacent states by $p(X_t|X_{t-1})$ and the relationship between the state and the observation variable by $p(I_t|X_t)$.

$$p(X_t|I_{1:t}) \propto p(I_t|X_t) \int p(X_t|X_{t-1})p(X_{t-1}|I_{1:t-1})dX_{t-1} \tag{1}$$

In Formula (1), $X = \{\alpha, \beta\}$. $\alpha = \{x, y, s, \theta, r\}$ where (x, y) is the position of the tracked object, s denotes the scale information, θ represents the in-plane rotation angle and r is the aspect ratio. Parameters in α obey the Gaussian *i. i. d.* β is a view parameter. We separate β from other state parameters because view is changing along the manifold while others obey the Gaussian distribution. Since α and β are independent, Formula (1) can be factorized as follows [6].

$$p(\alpha_t, \beta_t|I_t) \propto p(I_t|\alpha_t, \beta_t) \int\int p(\alpha_t|\alpha_{t-1})p(\beta_t|\beta_{t-1})p(\alpha_{t-1}, \beta_{t-1}|I_{t-1})d\alpha_{t-1}d\beta_{t-1} \tag{2}$$

According to the Bayes' theorem, $p(I_t|\alpha_t, \beta_t)$ can be written as Eq. (3).

$$p(I_t|\alpha_t, \beta_t) = p(\beta_t|I_t, \alpha_t)p(I_t|\alpha_t)/p(\beta_t) \tag{3}$$

Using Eq. (3), $p(\alpha_t, \beta_t|I_t)$ can be further unfolded. $p(\beta_t|I_t, \alpha_t)$ is a general face model which is used as a view estimator for I_t at α_t. To build such an independent general view manifold, we adopt a recently proposed one based on tensor decomposition and conceptual design [11]. Since this manifold is a general one, we will not update it.

The observation model $p(I_t|\alpha_t)$ defines the probability of Z_t belonging to an object-specific face, where Z_t is the image patch cropped by α_t from the video image. The image patch which has the maximum possibility of $p(I_t|\alpha_t)$ corresponding to I_t. To adapt to the face variation during tracking, $p(I_t|\alpha_t)$ has to be online updated. An object-specific model including the feature spaces of five discrete views as $S = \{L_n\}_{n=1}^{5}$ is built in this paper. In the feature space $L_n = \{c_n, \Phi_n, \wedge_n, P_n\}$, c_n, Φ_n, $\wedge_n$ and P_n records the mean face, eigenbasis, diagonal matrix of eigenvalues and the number of images used to construct the n-th view feature space, respectively. For simplicity, we only use the subscript n when it is necessary.

Given the object-specific model, the probability of $p(I_t|\alpha_t)$ is translated to the likelihood of $p(Z_t|S)$, which can be further expressed as the function of two distances: distance-from-feature-space (DFFS) and distance-in-feature-space (DIFS) as Eq. (4)

$$p(Z_t|L) = \left[\frac{exp\left(-1/(2\rho)d^2(Z_t, L)\right)}{(2\pi\rho)^{\frac{N-M}{2}}}\right]\left[\frac{exp\left(-1/2\sum_{i=1}^{M}(y_i^2/\lambda_i)\right)}{(2\pi)^{\frac{M}{2}}\prod_{i=1}^{M}\lambda_i^{\frac{1}{2}}}\right] \tag{4}$$

where N denotes the dimensionality of the image space. M is the number of principle components retained in Φ for the calculation of DIFS, and the rest $\{\Phi_i\}_{i=M+1}^{N}$ is used for the DFFS estimation. $d(Z_t, L)$ represents the distance from Z_t to $\{\Phi_i\}_{i=1}^{M}$. y is the projection coefficient from Z_t to L. The second

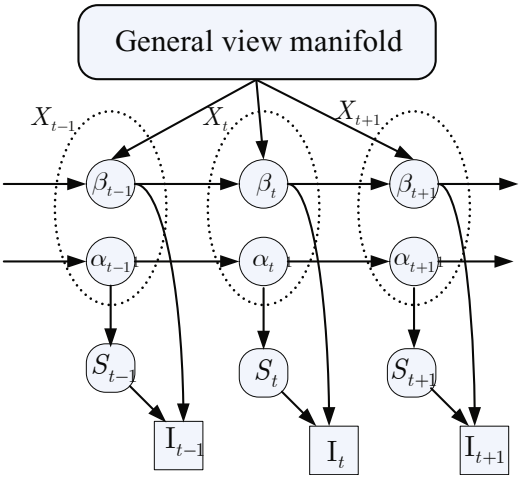


Fig. 1. The graphic model of the state-space model for multi-view face tracking

Gaussian can be interpreted as the likelihood of the distance between y and subspace center c.

To realize a robust tracking, the object-specific model should be update to adapt to the environmental changes. We incorporate a forgetting factor to let the model focus more on the current tracked accurate information. Tracking is realized by the following steps. A graphical model of it is illustrated as Fig.1.

Initialization: The face object I_1 in the first video frame is initialized manually.

Do for $t = 1, 2, \ldots, T$ (T is the number of frames in the video):

(1)Estimate the pose of I_t using the pose estimator $p(\beta_t|I_t, \alpha_t)$;

(2)According to the pose estimation result, we use I_t to update the feature space L_n nearest to β_t under the supervision of the point-to-manifold distance;

(3) Use the particle filter to estimate the face object I_{t+1}, which is determined by the biggest similarity $p(Z|L)$ between the particles and the specific face model S_t. The state of the chosen particle is α_{t+1}.

3 View Estimation Based on the General View Manifold

In [2], pose estimation is realized by finding the best match feature space L_n of I_t. If the face model only contains several discrete view feature spaces L_n, the drifting usually occurs during the view transferring. We use an offline trained general view manifold to describe the intermediate view more accurately. And we also use the view estimation result as a general supervision mechanism to the object-specific model updating. Given the I_t, pose estimation is determined by the point-to-manifold distance [12] in this work. We introduce the hybrid view manifold generation method in brief first. We refer the readers to [11] for more details.

3.1 The View Manifold Generation Method

Essentially, when the hybrid view manifold is involved, high order singular value decomposition (HOSVD) [13][14]is used twice. First, it is used to abstract the independent view vectors which form the global structure of the view manifold. Then this view manifold is used for nonlinear tensor decomposition where HOSVD is used again to abstract the identity coefficients for multi-factor face modeling.

Since only the view information is useful in the first HOSVD, we applied it on the tensorized multi-view faces $\mathcal{I}$ to extract the view information embedded in the view mode matrix $\mathbf{U}_{view}$ as Eq. (5)

$$\mathcal{I} = \mathcal{C} \times_1 \mathbf{U}_{view}, \tag{5}$$

where $\mathcal{C}$ is the core tensor. Rows of $\mathbf{U}_{view}$ span the parameter space of various views, which is independent with identity or other factors. To model the continuously and nonlinearly changed face views in the dynamic video, we interpolate the intermediate view information between the view vectors in $\mathbf{U}_{view}$ by the cubic Spline fitting. The view vector in the manifold is represented as $\mathbf{v}_\beta = g(\beta)$.

3.2 View Estimation Based on Point-to-Manifold Distance

The face image I_β^k under identity k and view β is decomposed as Eq. (6) according to multi-view face model in [11]. $\mathbf{p}^k$ is the identity vector.

$$I_\beta^k = \mathcal{C} \times_2 \mathbf{p}^k \times_3 \psi(\mathbf{v}_\beta) \tag{6}$$

$\mathcal{C}$ is the core tensor, which governs the interaction between the view and identity. $\psi(\mathbf{v}_\beta) = [\phi(dis(\mathbf{v}_\beta, \mathbf{z}_1)), \cdots, \phi(dis(\mathbf{v}_\beta, \mathbf{z}_N)), \quad 1, \quad \mathbf{v}_\beta^{\mathrm{T}}]^{\mathrm{T}}$ represents the radial basis function mapping between the view manifold and the face image space [15]. $\phi(\cdot)$ are Gaussian kernels. $\mathbf{z}$ are centers of these kernels sampled from the manifold. Given two vectors $\mathbf{v}_\beta$ and $\mathbf{z}$ of the same size, we adopt the cosine distance as dis, viz. $dis(\mathbf{v}_\beta, \mathbf{z}) = 1 - \mathbf{v}_\beta \cdot \mathbf{z}/(||\mathbf{v}_\beta||||\mathbf{z}||)$, to measure the similarity between them in the manifold space. "$\cdot$" denotes the dot-product of the two vectors. $||\cdot||$ is the norm of the vector.

Assuming the identity of the tracked object I_t is k ($k = 1, \ldots, K$), view vector $\mathbf{v}^k$ is calculated by solving the linear part of $\psi(\mathbf{v}^k)$. Since there are K identities during the manifold training, we combine the corresponding K view vectors by Eq. (7) to get the single view vector $\mathbf{v}_\beta$ of I_t.

$$\mathbf{v}_\beta = \sum\nolimits_k p_g(\mathbf{v}^k)\mathbf{v}^k \tag{7}$$

$p_g(\mathbf{v}^k)$is the normalized likelihood of $\mathbf{v}^k$ belonging to the view manifold $g(\cdot)$. It can be formulated as a function of the point-to-manifold distance $dis(\mathbf{v}^k, g(\cdot))$ as follows.

$$p_g(\mathbf{v}^k) \propto \exp\{-dis(\mathbf{v}^k, g(\cdot))/(2\sigma^2)\} \tag{8}$$

In our task, if $p_g(\mathbf{v}^k) < th_1$, the point $\mathbf{v}^k$ will be regarded as an outlier, whose $p_g(\mathbf{v}^k) = 0$. The number of outliers is N_{outl}. The view estimation of I_t at α_t is formulated as

$$p(\beta_t | I_t, \alpha_t) = p_g(\mathbf{v}_\beta) \tag{9}$$

4 Immediately Emphasized Online Updating

The specific model S only contains five feature spaces. To determine which feature space should be updated, we divided the view manifold into five groups: left profile, left half-profile, frontal, right half-profile, and right profile, which are corresponding to the five feature spaces in the specific model. Each feature space $L = \{c, \Phi, \wedge, P\}$. We adopt the SKL [4] to update the feature space incrementally.

Given three data matrixes A, B and C composed by the tracked objects $I_{1:n}$, $I_{n+1:n+m}$ and $I_{1:n+m}$, respectively, note that C is the concatenation of A and B. The feature space of A is denoted as $\{c_A, \Phi_A, \wedge_A, P_A\}$. When the feature space of A and the successive data B after A is available, we aim to obtain the feature space of C as $\{c_C, \Phi_C, \wedge_C, P_C\}$ by efficiently compute the SVD of the concatenation of A and B. This effective incremental online updating process is introduced in brief in Table1.

In Table 1, f is defined by the probability of the tracked view $p_g(\mathbf{v}_\beta)$ belonging to the view manifold $g(\cdot)$. If the outliers are too many in the view estimation process, say, $N_{outl} > th_2$, we believe that the tracked view is drifting away from the training benchmarks, where th_2 is an empirical value. We will not update the

Table 1. SKL online algorithm with penalty mechnism

Input: Data matrixes A, B, C. Feature space of A:$\{c_A, \Phi_A, \wedge_A, P_A\}$

Step1.Calculate the mean data of B as $c_B = {}^1\!/_m \sum_{t=n+1}^{n+m} I_t$, thus the mean of C can be obtained by $c_C = (fn/(fn+m))c_A + (m/(fn+m))c_B$ with the forgetting factor f.

Step2. Form a data matrix $\hat{B}$ as follows.

$$\hat{B} = \left[(I_{m+1} - c_B) \ldots (I_{m+n} - c_B)\sqrt{nm/(n+m)}(c_B - c_A)\right]$$

Step 3. Calculate matrix $\tilde{B}$, which represents the components of B orthogonal to Φ_A, by $\tilde{B} = orth(\hat{B} - \Phi_A \Phi_A^T \hat{B})$. $orth$ performs orthogonalization.

Step 4. Calculate the R matrix of the QR decomposition on $[\Phi_A \wedge_A B]$ as follows, which is used for further feature space calculation of C.

$$R = \begin{bmatrix} f\wedge_A & \Phi_A^T \hat{B} \\ 0 & \tilde{B}(\hat{B} - \Phi_A \Phi_A^T \hat{B}) \end{bmatrix}$$

Step 5. Do singular value decomposition on R: $R = \tilde{U}\tilde{\wedge}\tilde{V}^T$.

Step 6. Φ_C, $\wedge_C$ and P_C can be obtained by $\Phi_C = \left[\Phi_A \;\; \tilde{B}\right]\tilde{U}$, $\wedge_C = \tilde{\wedge}$ and $P_C = n + m$, respectively.

Output: The feature space of C is $\{c_C, \Phi_C, \wedge_C, P_C\}$.

observation model when the drifting occurs. Otherwise, we define the forgetting factor as $f \propto p_g(\mathbf{v}_\beta)$.

5 Experimental Results and Analysis

We test the proposed approach on three different video databases including Honda/UCSD[2], the davidin300 dataset from [4] and the home-brewed dataset, where faces have complex motions. The proposed algorithm is compared with two state-of-art algorithms: The online learned probabilistic appearance manifolds (PAM) based tracking in [2] and the incremental visual tracking (IVT) in [4]. To distinguish the tracking results, faces tracked by our method, PAM and IVT are included in green, red and magenta rectangles, respectively, in the following.

The generic view manifold is constructed offline by detecting and aligning faces of 30 identities under 9 views in part of Honda/UCSD video database, Oriental Face database. The view manifold constructed from manually aligned faces is slightly sensitive to the state initialization of face tracking. The generic face model used in PAM is constructed from the same training data of the generic view manifold. PAM contains 5 piece-wised manifolds according to [2]. In order to reduce the influence of illumination, we normalize the intensity of I_t to obey the intensity distribution of the training data before pose estimation in our method. In these three methods, the tracked target images are normalized to 32×32 pixels, the number and the variance of the particles are the same. Each testing video is initialized manually with the same location for all the three tracking methods.

(1) Tracking on Honda/UCSD database
HONDA database has large variations of out-of-plane head rotation. The representative frames of tracking results are illustrated in Fig.2, which shows that the proposed method adapts well to the pose transferring. But the tracking result in the 63-*th* frame is not accurate. It is because the out-of plane rotation in tilt is not included in our view manifold. PAM has good tracking results only when the pose estimation is correct and the tracked faces are cropped well before updating. However, there is no supervision on the tracked faces before updating. And the specific discrete feature spaces in PAM have limited ability to cope with the nonlinearity of view transferring. Once the tracking errors accumulate over frames, drifting will happen. IVT learns the model of the object from scratch, whose accuracy depends on the view transfer speed. And there is no supervision on the accuracy of the tracked object before updating.

In Fig. 3, we illustrate the representative frames of tracking results of the proposed method with (in green) or without (in white) the surveillance of the updating mechanism. We can see that the miss-tracking in green in the 49-*th* and 50-*th* frames are corrected soon. However, it takes a longer time to obtain the proper particles without the updating constrain.

(2) Tracking with davidin300 dataset
The davidin300 dataset includes extreme shadowing changes. Fig.4 exhibits the representative frames of tracking results with three methods. The object-specific

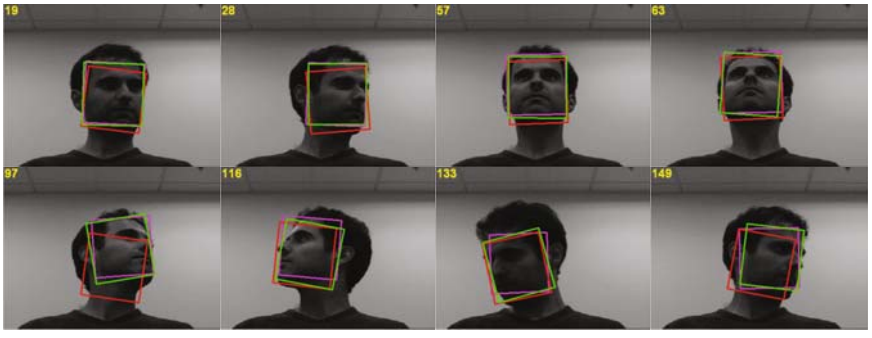


Fig. 2. Tracking comparison among our method (in green), PAM (in red) and IVT (in magenta) on HONDA/UCSD dataset

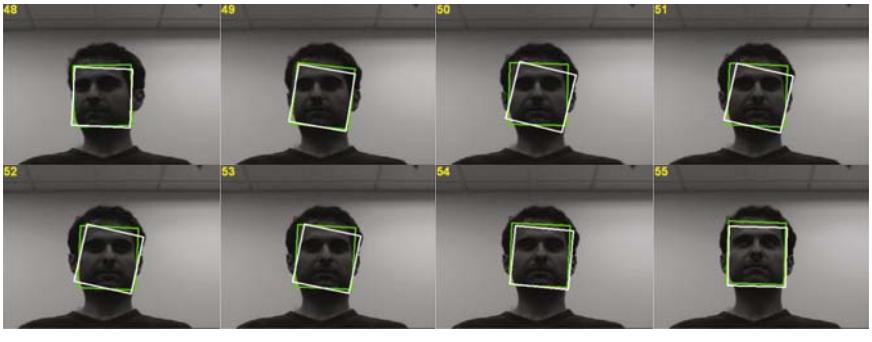


Fig. 3. Tracking results of the proposed method with (in green) or without (in white) the surveillance of updating mechanism

model of PAM degenerates from the general model, which cannot adapt to the different environment very soon. The accumulated inaccuracy causes the drifting. While, the specific model of our method and the feature space of IVT are built based on the specific object in the video. Thus, the proposed method and IVT have better results. Furthermore, our method adapts to the fast head rotation better compared with IVT method.

(3) Home-brewed video tracking results

To further verify the effectiveness of the proposed tracking algorithm, we apply it to our home-brewed video dataset, where the illumination is different with the training data and complex background are involved. The representative frames of tracking results are given in Fig.5. Though the result of pose estimation is correct in PAM, but the illumination difference between the training data of PAM and the testing video data leads to the inaccurate locations during tracking, which are used to update the general model in PAM for particle filter. Thus drifting happens in the first several frames of the home-brewed video. IVT and our method have a better tracking result compared with PAM because the feature

Fig. 4. Tracking results of IVT (in magenta), PAM (in red) and our method (in green) on davidin300 dataset

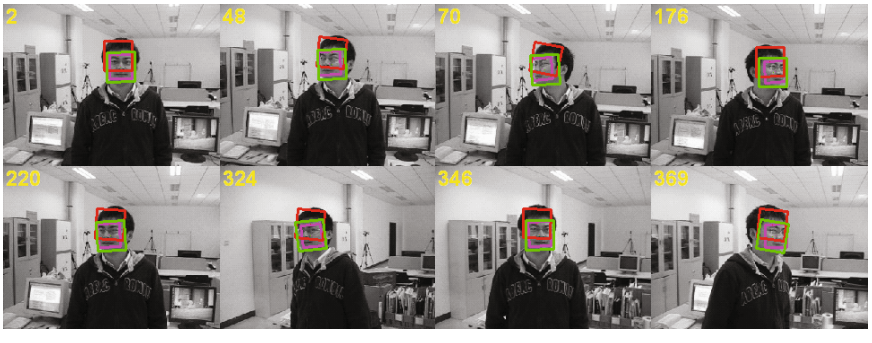


Fig. 5. Tracking results of IVT (in magenta), PAM (in red) and our method (in green) on the home-brewed dataset

space in both of the methods are built directly on the accurately located faces in the testing video. Our method has comparative results with IVT in this database since the head rotates smoothly in this video.

6 Conclusions and Future Work

We proposed an online updating multi-view face tracking method in the graphic model. A general view manifold is involved to provide a general appearance constrain for the tracked multi-view faces before updating and determine which feature space should be updated. The experimental results show great promise of the proposed method on multi-view face tracking. However, the method cannot handle the very large out-of-plane rotation in tilt well, which is our further study focus.

Acknowledgments. The Oriental Face database collected under the research of the Artificial Intelligence and Robotics (AI & R) at Xi'an Jiaotong

University, China. The authors would like to thank David Ross for providing the code of incremental visual tracking algorithm. This work is supported by the National High Technology Research and Development Program (863) of China (No.2009AA01Z315), National Natural Science Foundation of China (No.60872145), China Postdoctoral Science Foundation (No.20090451397) and Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (No.708085).

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)
2. Lee, K.-C., Kriegman, D.: Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 852–859 (2005)
3. Black, M.J., Jepson, A.D.: Eigentracking: Robust Matching and Tracking of Articulated Objects Using a View-based Representation. Int. J. Computer. Vision 26(1), 63–84 (1998)
4. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental Learning for Robust Visual Tracking. Int. J. Computer. Vision 77(1), 125–141 (2008)
5. Brand, M.: Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 707–720. Springer, Heidelberg (2002)
6. Wang, P., Ji, Q.: Robust Face Tracking via Collaboration of Generic and Specific Models. IEEE Trans. Image Processing 17(7), 1189–1199 (2008)
7. Jepson, A., Fleet, D., EI-Maraghi, T.: Robust Online Appearance Models for Visual Tracking. IEEE Trans. Pattern Analysis and Machine Intelligence 25(10), 1296–1311 (2003)
8. Chen, Y., Rui, Y., Huang, T.: JPDAF Based HMM for Real-time Contour Tracking. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 543–550 (2001)
9. Wu, Y., Huang, T.: A Co-inference Approach to Robust Visual Tracking. In: Proc. IEEE Int. Conf. on Computer Vision, vol. 2, pp. 26–33 (2001)
10. Comanicu, D., Ramesh, V., Meer, P.: Real-time Tracking of Non-rigid Objects Using Mean Shift. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 142–149 (2000)
11. Tian, C., Fan, G., Gao, X.: Multi-view Face Recognition by Nonlinear Tensor Decomposition. In: Proc. Int. Conf. Pattern Recognition, pp. 1–4 (2008)
12. Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-Manifold Distance with Application to Face Recognition based on Image Set. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 1–8 (2008)
13. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) Proceedings of the Seventh European Conference on Computer Vision, pp. 447–460 (2002)
14. Tamara, G.K.: Orthogonal Tensor Decompositions. SIAM Journal on Matrix Analysis and Applications 23(1), 243–255 (2001)
15. Poggio, T., Girosi, F.: Networks for Approximation and Learning. Proc. of the IEEE 78(9), 1481–1497 (1990)

Hierarchical Model for Joint Detection and Tracking of Multi-target

Jianru Xue, Zheng Ma, and Nanning Zheng

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
No 28, Xianningxilu. Xi'an, China, 710049
jrxue@mail.xjtu.edu.cn
http://www.aiar.xjtu.edu.cn

Abstract. We present a hierarchical and compositional model based on an And-or graph for joint detecting and tracking of multiple targets in video. In the graph, an And-node for the joint state of all targets is decomposed into multiple Or-nodes. Each Or-node represents an individual target's state that includes position, appearance, and scale of the target. Leaf nodes are trained detectors. Measurements that supplied by the predictions of the tracker and leaf nodes are shared among Or-nodes.There are two kinds of production rules respectively designed for the problems of varying number and occlusions. One is association relations that distributes measurements to targets, and the other is semantic relations that represent occlusion between targets. The inference algorithm for the graph consists of three processing channels: (1) a bottom-up channel, which provides informative measurements by using learned detectors; (2) a top-down channel, which estimates the individual target state with joint probabilistic data association; (3) a context sensitive reasoning channel, which finalizes the estimation of the joint state with belief propagation. Additionally, an interaction mechanism between detection and tracking is implemented by a hybrid measurement process. The algorithm is validated widely by tracking peoples in several complex scenarios. Empirical results show that our tracker can reliably track multi-target without any prior knowledge about the number of targets and the targets may appear or disappear anywhere in the image frame and at any time in all these test videos.

1 Introduction

The problem of multi-target tracking in video (MTTV) requires recursive localizing targets and labeling their identities within each frame of the video clip. Recent research demonstrates that tracking of isolated targets or a small number of targets having transient occlusion is no problem. However, automatic tracking of an unknown number of interacting targets in video is still a challenging problem, especially in situations such as occlusion between targets passing in front and behind each other and occlusion behind a part of the scene, targets having little distinction in their appearance, varying number of targets, and change in pose and in the illumination of the scene.

We believe that the problem of accurate identification of targets in MTTV needs a hierarchical and probabilistic model to address the aforementioned difficulties, and should take three observations into consideration: (1) data association techniques considering occlusion are needed to maintain an unique identifier for each track when visual

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 160–171, 2010.

measurements are indistinguishable or absent temporarily due to occlusion. (2) target existence should be treated jointly with the target state in data association process. A suitable prior process can then be specified for the existence variables, which facilitates the joint inference with the state and data association processes. This in essence is an important component of the approach of this paper. (3) the detecting and tracking of targets should be combined together instead of being separated into two independent procedures. An interactive mechanism between detection and tracking is essential to an robust tracker for MTTV. This not only allows the detection to make use of temporal consistency, but also facilitates robust tracking of multi-target.

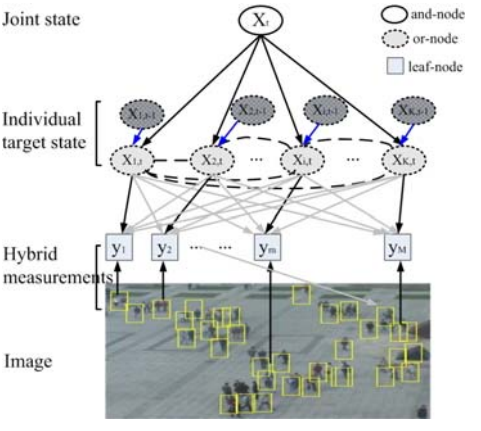


Fig. 1. The hierarchial composition model for detection and tracking multi-target. The joint state of targets is decomposed into individual target states, and share measurements between them.

We address these observations with a hierarchical and compositional model for MTTV, as shown in Fig 1. There are two contributions in this work: (1) an And-Or graph which explicitly and jointly models target state, occlusion among targets, varying number, and data association; (2) an Monte Carlo inference algorithm for the And-or graph. The proposed model provides a general modeling framework for MTTV, which is reconfigurable to track many classes (even different classes) of targets.

1.1 Related Work

MTTV is usually formulated within a sequential Monte Carlo filtering framework (SMCF) [1,2,3,4]. These methods try to explain the foreground or motion blobs by fitting multi-target hypotheses, and handle data association implicitly. They deal with occlusions by computing joint image likelihood of multi-target. An efficient optimization algorithm, such as MCMC [1], particle filtering [4,3], or EM [2] is often required because of the high dimensionality of the multi-target hypothesis space. These methods have shown experiments with a stationary camera only, where the background subtraction provides relatively robust target motion blobs.

Recently some strategies have been proposed to accommodate the data association within the SMCF framework. The feasibility has been claimed in [5], but the examples there deal only with a single target. Simply inserting data association into the multi-target hypothesis space within a SMCF framework makes the complexity of inference algorithm increase exponentially with the number of targets. More recently, the data association technique is used as a complement in [6]. A two-level computing architecture consists of particle-based JPDAF and belief propagation is proposed in [7], which treats the varying number problem and occlusion among targets in a post-processing manner. These works demonstrate an interesting research trend. However, these MCMC-based MHT methods are iterative and batched-processing in nature. They need an unknown number of iterations to converge, and are not entirely suitable for the online tracking applications. Moreover, most of these data association methods typically do not address the occlusion problem.

2 Hierarchical Composition Model

2.1 And-Or Graph for MTTV

We model the MTT problem with an And-Or graph [8], as shown in Fig 1. The graph has a recursive structure with two types of nodes: And-node and Or-node. Each And-nod is a MRF, and each Or-node is a switch node in a Markov tree. The And-Or graph combines Markov tree and Markov random fields (MRF), which makes it possible to avoid the connections of all pairs of variables that are conditionally dependent even for a single choice of values of other variables.

At each time step t, an And-Or graph denoted by a 6-tuple,$\mathbb{G} =< U, V, T, \mathcal{R}, \Sigma, \mathcal{A} >$, is configured for the joint state of all targets. The And-node U represents the joint state $\mathbf{X}_t$, which decompose into a set of K Or-nodes, $V = \{\mathbf{x}_{i,t}\}_{i=1}^{K}$, where K is the maximum number of targets. Each Or-node represents an individual target's state, $\mathbf{x}_{i,t}$, and it also associates with a binary variable $e_{i,t}$ that denote the existence of each target and its previous state $\mathbf{x}_{i,t-1}$. We denote $e_{i,t} = 1$ as $E_{i,t}$ if the ith target exists, and $e_{i,t} = 0$ as $\bar{E}i, t$ if the ith target disappears. $\mathbf{e}_t = \{e_{i,t}\}_{i=1}^{K}$ denotes the existence status of all these K targets. The full joint state space of $\mathbf{X}_t$ is the union of spaces $\{\mathbf{x}_{i,t} : E_{i,t}\}_{i=1}^{K}$. The number of targets present at time t is denoted by k_t, which can be determined by $k_t = \sum_i^K e_{i,t}$. In the experiments, we define the state of each target as its 2D location and scale in image, along with the velocities of these quantities.

The terminal nodes T represent measurements $\mathbf{Y}_t = \{\mathbf{y}_j, j = 1, ..., m_t\}$ that comprises m_t detections. A target may disappear with a probability P_{de}, and a new target may appear with a probability P_{re}. More specifically, we assume that the target state can be measured with a detection probability P_d less than unity, and the number of false alarms follows a Poisson distribution parameterized by $\lambda_f V$, where V is the volume of a surveillance region in the view of the camera. The measurements are unlabeled and maybe due to the target or clutter. Measurements from the predictions of the tracker and detection of the trained classifier are shared by Or-nodes.

Two sets of production rules $\mathcal{R}$ are designed for the graph: $R^{(1)}$ for associating relations and $R^{(2)}$ for semantic contexts. Each rule $r_{i,t} \in R^{(1)}$ associates one measurement in $\mathbf{Y}_t$ with the ith target, and each rule $\gamma_{k,l} \in R^{(2)}$ represents occlusion between the

kth and the lth targets. Attributes $\mathcal{A} = \{A_j, j = 1, ..., m_t\}$ represent the photometric attributes of the measurements. In this paper, each A_j denotes the color-histogram of the jth measurement.

The And-Or graph can generate a large number of configurations, which is denoted as $Sigma$. Each configuration $G \in \Sigma(G)$ carries the context and Markov constraints in the graph, and represent a possible joint state. Specifically, each G has the following constituents.

1. $V(G) = \{\mathbf{x}_{i,t}|E_{i,t}\}$ is a set of Or-nodes that are used in configuration G, which is determined by the value of $\mathbf{e}_t$.
2. $R(G) = \{R^{(1)}, R^{(2)}\}$ is the set of valid relations. Each relation $r_{i,t} \in R^{(1)}$ is a target to measurement association variable, and is defined as

$$r_{i,t} = \begin{cases} 0 & \text{if the } i\text{th target is undetected,} \\ j & \text{if } \mathbf{x}_{i,t} \text{ associates with } \mathbf{y}_j. \end{cases}$$

   where $j \in \{1, ..., m_t\}$. Relations $\gamma_{k,l} \in R^{(2)}, \forall l, k \in \{1, ..., K\}$, are defined over each two targets.
3. $T(G) = \{\mathbf{y}_{r_{i,t}}|\mathbf{x}_{i,t} \in V(G)\}$ is the terminal nodes in configuration G. Each terminal node is a measurement associated with one Or-node in the G. It determines the track of the target in the temporal-spatial space.
4. $S(G)$ is the photometric attribute of the measurement.

2.2 Inference Algorithm

The goal of MTTV now is to compute the optimal configuration G given measurements $\mathbf{Y}_{1:t} = \{\mathbf{Y}_1, ..\mathbf{Y}_t\}$, and this is equivalent to optimize a posterior probability,

$$G^* = \arg \max_{G \in \Sigma(G)} p(\mathbf{Y}_{1:t}|G)p(G). \tag{1}$$

where $p(\mathbf{Y}_{1:t}|G)$ is the measurement likelihood function, and $p(G)$ is the probabilitic model for the And-or graph. A configuration G integrates the Markov tree model for the Or-nodes and the Markov random filed model for the And-nodes, and leads to a mixed Markov model [9]. Thus the posterior probability for G given measurements $\mathbf{Y}_{1:t}$ can be computed as

$$p(G|\mathbf{Y}_{1:t}) = \frac{1}{Z(G)} \prod_{\mathbf{x}_{i,t} \in V(G)} p(\mathbf{x}_{i,t}) \prod_{r_{i,t} \in R^{(1)}} p(\mathbf{x}_{i,t}, \mathbf{Y}_{1:t}) \prod_{\gamma_{k,l} \in R^{(2)}} p(\mathbf{x}_{k,t}, \mathbf{x}_{l,t}). \tag{2}$$

where $Z(G)$ is the partition function which is related to the And-or Graph $\mathbb{G}$, and is common to all graph configurations in $\Sigma(G)$. Thus we need not compute $Z(G)$ when we switch between different configurations in the inference algorithm.

We solve Eq(2) by an SMCF algorithm, which consists of three inference channels: First, a top-down channel factorizes the posterior of the joint state over individual targets by Monte Carlo data association, and each individual target's state is then estimated independently. Then with the marginal filtering distribution for each target as initialization,

a context sensitive channel runs a Particle-based Belief propagation [10] to deal with the occlusions between targets. A hybrid measurement process integrates detected measurements by detectors in the bottom-up channel and predicted measurements by the SMCF in the top-down channel.

3 Top-Down Channel

Without considering occlusions between targets, the first two terms of Eq(2) represents a factorization of the joint state $\mathbf{X}_t$ over individual targets if we model $r_{i,t}$ and $e_{i,t}$ explicitly. The top-down channel approximates the factorization by recursively outputting the posterior probability of existence $P_{E_{i,t}} = p(E_{i,t}|\mathbf{Y}_{1:t})$, and the filtering distribution $p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t})$, for each individual target.

$$p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t}) = \sum_{r_{i,t}} p(\mathbf{x}_{i,t}|r_{i,t}, E_{i,t}, \mathbf{Y}_{1:t}) p(r_{i,t}|E_{i,t}, \mathbf{Y}_{1:t}), \tag{3}$$

where $p(r_{i,t}|E_{i,t}, \mathbf{Y}_{1:t})$ is the posterior of the association variable $r_{i,t}$, and $p(\mathbf{x}_{i,t}|r_{i,t}, E_{i,t}, \mathbf{Y}_{1:t})$ is the filtering distribution conditional on the association $r_{i,t}$.

The first term of Eq(3) can be computed as

$$p(\mathbf{x}_{i,t}|r_{i,t}, E_{i,t}, \mathbf{Y}_{1:t}) = \frac{p_T(\mathbf{y}_{r_{i,t},t}|\mathbf{x}_{i,t}) p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1})}{\int_{\mathbf{x}_{i,t}} p_T(\mathbf{y}_{r_{i,t},t}|\mathbf{x}_{i,t}) p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1})} \tag{4}$$

by using the Bayes' rule and the fact that the measurements at a time step are independent conditional on the target state, existence and association variables, where $p_T(\mathbf{y}_{r_{i,t}}|\mathbf{x}_{i,t})$ is the likelihood function of the measurement $\mathbf{y}_{r_{i,t}}$ conditioned on the state $\mathbf{x}_{i,t}$.

We substitute Eq(4) into Eq(3), and obtain the filtering distribution for each individual target

$$p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t}) = \sum_{j=0}^{m_t} \beta_{ij} \frac{p_T(\mathbf{y}_j|\mathbf{x}_{i,t}) p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1})}{p(\mathbf{y}_j|E_{i,t}, \mathbf{Y}_{1:t-1})}. \tag{5}$$

where $\beta_{ij} = p(r_{i,t} = j|E_{i,t}, \mathbf{Y}_{1:t})$ denotes the association probability that the jth measurement $\mathbf{y}_j$ associates with the ith target, $i \in \{1, ..., K\}, j \in \{1, ..., m_t\}$. Similarly, β_{i0} means that the ith target goes undetected.

To recursively update the filtering distribution Eq(5), we need to compute three unknown terms: the state prediction distribution $p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1})$, the posterior of the association variable $p(r_{i,t}|E_{i,t}, \mathbf{Y}_{1:t})$, and the posterior probability of existence $P_{E_{i,t}}$. These are discussed in the subsequent two subsections.

3.1 The State Prediction Distribution

The state prediction can be computed as

$$p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1}) = \sum_{e_{i,t-1}} p(e_{i,t-1}|E_{i,t}, \mathbf{Y}_{i:t-1})$$

$$\times \int_{\mathbf{x}_{i,t-1}} p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}, E_{i,t}, e_{i,t-1}) p(\mathbf{x}_{i,t-1}|e_{i,t-1}, \mathbf{Y}_{1:t-1})$$

by introducing the previous state and existence $\mathbf{x}_{i,t-1}, e_{i,t-1}$, and using the total probability.

With the Bayes' rule, assumption that $p(E_{i,t}|e_{i,t-1}, \mathbf{Y}_{1:t-1}) = p(E_{i,t}|e_{i,t-1})$, and the defined probability terms P_{re} and P_{de} (Section 2.1), the state prediction distribution finally follows as

$$p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t-1}) = \frac{1}{\kappa}[P_{re}(1 - P_{E_{i,t-1}})p_0(\mathbf{x}_{i,t}) + (1 - P_{de})P_{E_{i,t-1}}p(\mathbf{x}_{i,t}|\mathbf{Y}_{1:t-1})]$$

where $\kappa = P_{re} + P_{E_{i,t-1}}(1 - P_{re} - P_{de})$, $p_0(\mathbf{x}_{i,t})$ is the initial state distribution of the ith target, $P_{E_{i,t-1}} = p(E_{i,t-1}|\mathbf{Y}_{1:t-1})$, and $p(\mathbf{x}_{i,t}|\mathbf{Y}_{1:t-1})$ is the state prediction distribution of the ith target which continuously exists from time $t-1$ to t.

3.2 Posteriors of Association and Existence

Thus far, we need the marginal posterior of existence variable $p(E_{i,t}|\mathbf{Y}_{1:t})$, and the conditional posterior of the association variable β_{ij} to update the filtering distribution $p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t})$. They both can be computed by marginating the joint posterior of existence and association as follows:

$$p(E_{i,t}|\mathbf{Y}_{1:t}) = \sum_{j=o}^{m_t} p(E_{i,t}, r_{i,t} = j|\mathbf{Y}_{1:t}), \text{ and } \beta_{i,j} = \frac{p(E_{i,t}, r_{i,t} = j|\mathbf{Y}_{1:t})}{p(E_{i,t}|\mathbf{Y}_{1:t})}. \quad (6)$$

Once a pair $(\mathbf{e}_t, \mathbf{r}_t)$ is given, M_C and M_T are deterministically calculated. For the time being, we assume that association vectors take a form that factorizes sequentially over the individual associations,

$$q(\mathbf{r}_t|\mathbf{e}_t) = p_c(M_C) \prod_{k=1}^{k_t} q(r_{k,t}|r_{1:k-1,t}), \text{ and } p_c(M_C) = \frac{(\lambda_f V)^{M_C}}{M_C!} e^{-\lambda_f V}, \quad (7)$$

where p_C denotes the clutter likelihood. It should be noted that the sequential factorization can be performed over any permutations of the individual targets.

The components of an association vector can be sampled sequentially conditional on each other, and the proposal for the kth component is conditional on all the components sampled earlier in the sequence, since Eq(7) depends only on information available at the current time step. We make use of this property to ensure that the measurements associated with targets earlier in the sequence are not considered as candidates to be associated with the current target. In this way, it is guaranteed that only valid association hypotheses are generated. Thus we have

$$q(r_{i,t} = j|r_{1:i-1,t}) \propto \begin{cases} 1 - P_d & \text{if } j = 0 \\ 0 & \text{if } j > 0 \text{ and } j \in \{r_{1,t}, ..., r_{i-1,t}\} \\ \frac{P_d}{M_i} & \text{otherwise} \end{cases}$$

where $M_i = m_t - \sharp\{l : r_l \neq 0, l = 1, ..., i-1\}$ is the number of unassigned measurements, taking into account the assignments of the previous $i-1$ associations.

The joint posterior of the existence and association variables can then be approximated as

$$p(\mathbf{e}_t, \mathbf{r}_t|\mathbf{Y}_{1:t}) \propto p(M_C) \prod_{i=1}^{K} [q(r_{i,t}|r_{1:i-1}, t)p(e_{i,t}|\mathbf{Y}_{1:t-1})p(\mathbf{y}_{r_{i,t},t}|e_{i,t}, \mathbf{Y}_{1:t-1})]. \quad (8)$$

where the only unknown term $p(e_{i,t}|\mathbf{Y}_{1:t-1})$ can be computed as

$$p(e_{i,t}|\mathbf{Y}_{1:t-1}) = \sum_{e_{i,t-1}} p(e_{i,t}|e_{i,t-1})p(e_{i,t-1}|\mathbf{Y}_{1:t-1})$$

Then $p(E_{i,t}|\mathbf{Y}_{1:t})$ and $\beta_{i,j}$ can be computed by enumerating all the valid existence-association pairs, and evaluating the probabilities for each of these pairs with Eq(8). We use a soft-gating procedure to reduce the number of valid data association hypotheses. A validation region is constructed for each target, and only measurements that fall within the target validation region are considered as possible candidates to be associate with the target. The validated set of measurements for the ith target can be defined as $\mathbf{Y}_i = \{y_j : d_i^2(y_j) \leq \xi\}$, where $d_i^2(y_j)$ is the squared distance between the measurement and the center of the validation region of the ith target, and ξ is a parameter which determines the size of the validation region. The validation region of a target can be calculated by first assuming the set of particles as a Gaussian mixture model, then approximating the Gaussian mixture as a single Gaussian, and finally computing the mean and covariance matrix of the particle set.

3.3 Monte Carlo Data Association Filtering

The Monte Carlo data association filtering assumes that a set of particles $\{w_{i,t-1}^{(n)}, \mathbf{x}_{i,t-1}^{(n)}\}_{n=1}^{N}$ which approximately distributed according to $p(\mathbf{x}_{i,t-1}|E_{i,t-1}, \mathbf{Y}_{1:t-1})$, and $P_{E_{i,t-1}}$ are available. New particles for $\mathbf{x}_{i,t}$ could be sampled from a suitably proposal distribution. Instead of using traditional transition distributions modeled by autoregressive state dynamics, we sample from a mixture proposal (Eq(9)) to integrate information from both the target dynamics and the bottom-up detections.

$$p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}, \mathbf{Y}_t) = \alpha Q_{ada}(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}, \mathbf{Y}_t) + (1-\alpha)p(\mathbf{x}_{i,t}|\mathbf{x}_{i,t-1}) \quad (9)$$

where Q_{ada} is a multi-modality probability distribution constructed by the output of a learned detector. Each mode of the distribution centers around one detection, and is modeled by a Gaussian distribution approximated by a cluster of samples. Other kinds of detectors can also be adopted. The parameter $0 \leq \alpha \leq 1$ can be set dynamically. By decreasing α, one can place more importance on the prediction than on the detector.

With the samples from the Eq(9), the predictive likelihood is approximated straightforwardly as

$$p(\mathbf{y}_{j,t}|E_{i,t}, \mathbf{Y}_{1:t-1}) \approx \sum_{n=1}^{N} w_{i,t-1}^{(n)} p_T(\mathbf{y}_{j,t}|\mathbf{x}_{i,t}^{(n)}) \frac{p(\mathbf{x}_{i,t}^{(n)}|\mathbf{x}_{i,t-1}^{(n)})}{p(\mathbf{x}_{i,t}^{(n)}|\mathbf{x}_{i,t-1}^{(n)}, \mathbf{Y}_t)}. \quad (10)$$

Along with association probabilities β_{ij}, we finally obtain the filtering distribution $p(\mathbf{x}_{i,t}|E_{i,t}, \mathbf{Y}_{1:t})$, which is approximated by the particles $\{w_{i,t}^{(n)}, \mathbf{x}_{i,t}^{(n)}\}_{n=1}^{N}$. The weights are updated as

$$w_{i,t}^{(n)} \propto w_{i,t-1}^{(n)} \frac{p(\mathbf{x}_{i,t}^{(n)}|\mathbf{x}_{i,t-1}^{(n)})}{p(\mathbf{x}_{i,t}^{(n)}|\mathbf{x}_{i,t-1}^{(n)}, \mathbf{Y}_t)} p(\mathbf{Y}_t|\mathbf{x}_{i,t}^{(n)}), \sum_{n=1}^{N} w_{i,t}^{(n)} = 1.$$

4 Context Sensitive Reasoning Channel

The filtering distribution of Or-nodes output by the top-down channel are purely local, since occlusions are not taken into consideration. Initializing Or-nodes with these filtering distributions, we then run a particle-based belief propagation(PBP) [10] in the context sensitive channel to compute the final configuration G.

In PBP, local evidence function is defined as the matching cost function of state $\mathbf{x}_i$ given measurement $\mathbf{y}_i$. In the experiment, we use color histogram to represent the target appearance, and the Bhattacharyya distance as the matching cost. Interactive potential function for two neighboring Or-nodes is defined as

$$\Psi(\mathbf{x}_i, \mathbf{x}_j) = (1 - e_p) \exp\{-\frac{|\Omega(\mathbf{x}_i, \mathbf{x}_j)|}{\sigma_p}\} + e_p \tag{11}$$

where $\Omega(\mathbf{x}_i, \mathbf{x}_j)$ is maximal when two targets coincide and gradually falls off as they move apart. $|\Omega(\mathbf{x}_i, \mathbf{x}_j)|$ denotes the number of pixels overlapping between two targets. Different from the upside-down Gaussian in [11] [12], our Ψ is from the Total Variance (TV) model [13] with a potential function $\Psi(z) = |z|$ because of its discontinuity preserving property. By varying parameters e_p and σ_p, one can control the shape of $\Psi(\mathbf{x}_i, \mathbf{x}_j)$, therefore the final posterior probability. This is important in preventing particles for the occluded targets from a fast depletion.

5 Bottom-Up Channel and Hybrid Measurement Process

In the bottom-up channel, a detector is use to scan each input frame image to supply measurements. Any detector that can supply measurements can be used here. In the experiment, we trained an AdaBoost classifier as a detector.

The hybrid measurement process is implemented via a factored sampling procedure, which enhance the interaction between the detection and tracking. The factored sampling procedure can find nonlocal maxima of multi-modal image likelihoods. It is implemented as follows: it first eliminates low-fitness samples from the mixture proposal Eq(9), then it refines the remaining samples to obtain a local maxima $\mathbf{x}_i^{'}$ by using mean shift algorithm [14]. Specifically, starting with the fittest sample $\mathbf{x}_{best}$, all less fit samples $\mathbf{x}_i$ such as $|\mathbf{x}_{best} - \mathbf{x}_i| \leq \delta$ are eliminated. The purpose of δ is to compensate for any lack of precision in the mean shift algorithm. The thinning process is repeated for the next fittest sample and so on. Finally, we yield a set of m_t measurements. The amount m_t may vary due to the randomness of the sampling procedure and whether the image actually has only m_t target-like features.

We deal with the mutual occlusion problem by sequential factorization of the data association. Each permutation of the individual target association provides an occlusion relationship hypothesis. Under each permutation, the target in the front of the sequence is measured first from the measurements, then the target in the second can be matched from the masked measurements, and so on. In this way, measurement results of non-overlapping targets are equivalent for different depth ordering permutation.

6 Experiments and Discussion

We test the proposed framework on a challenging people tracking problem using several video clips from the public PETS2003 data set and our own captured data. Due to targets and camera motion in the video sequences, the number of targets in view varies continuously, and occlusion occurs frequently. For comparison purpose, we also implemented an MCJPDAF tracker proposed in [5] and an integrated tracker proposed in [15]. All these three trackers are configured with a same trained detector.

We have trained an AdaBoost classifier for people based on the histograms of oriented gradient [16] in the experiment. We collected 2250 images as the positive set. A fixed set of 13000 patches sampled randomly from 1350 human-free images are used as negative set. The resulted detector selected 1000 features from 13572 features, which is nested in a one-layer structure. Its detection rate and false alarm rate are respectively set as 90% and 20% when it is trained.

We use a color-histogram model as attribute for a terminal node. Each one is sampled in the first frame at which the target appears. In tracking, we assign 100 particles to each target. Parameters are set as $\alpha = 0.3$(in Eq(9)), $\lambda_f = 20$, $e_p = 0.1$, and $\sigma_p = 432$. It should be noted this parameter setting is not the optimal one. However, other parameter settings almost do not affect the tracker's performance.

6.1 Performance Evaluation

In order to quantitatively analyze performance of our system, we use 2 testing sequences provided by camera 3 in PETS2003 for outdoor people tracking[1]. In evaluation, we represent both the ground truth (GT) information and the tracked objects in terms of the bounding box in each frame. After building correspondence between the GT's tracks and the system's (when the average discrepancy between system's box and the GT's box is below 100 pixels), we calculated the object-based metric [17] for these three trackers, and the statistics are presented in Table 1. It shows that our tracker outperform other two trackers.

6.2 Tracking Football Players and Pedestrians

We also compare our tracker with the MCJPDAF tracker and the integrated tracker in two complex scenes: (1) tracking football players in video sequences, and (2) tracking pedestrians in a dynamic scene. Both video sequences are filmed by a handy camera.

[1] http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html

Table 1. Performance comparison of three tracking systems which is evaluated with PETS20003 data set. Ours outperforms the other two tracking systems. TRDR: tracker detection rate, FAR: false alarm rate, DR: detection rate, FNR: false negative rate, FPR: false positive rate.

Methods	TRDR	FAR	DR	FNR	FPR
Our tracker	0.93	0.09	0.93	0.07	0.18
Integrated tracker	0.81	0.15	0.81	0.19	0.22
MCJPDAF tracker	0.75	0.25	0.75	0.25	0.26

The video sequences used are with a resolution of 223×153, a length of 3000 frames. The number of people in a frame ranges from 6 to 9, and the pose of player changes. The camera is shaking sometimes during the video sequence. This makes the performance of the detector degrade dramatically, as indicated by the ROC curves shown in Fig 2. The curve $Test0$ is obtained using the testing set of MIT pedestrian data set [16], and the curve $Test1$ is obtained by the testing of 800 images which we collected from two test video sequences we used here.

Our tracker can track football players and maintain their identifiers through most of the video clip, while the MCJPDAF tracker failed within the first 20 frames, and the integrated tracker failed in managing the identifiers of all targets, and can only maintains two tracks less than 300 frames. In pedestrian tracking, our tracker failed in tracking some targets in a few frames because the failure of detector in detecting these targets lasts more than 15 frames. Some tracking results of our tracker and MCJPDAF are presented in Fig 3, and more are available from the video files "MTT-Football.avi".

The number of football players in about 400 frames counted by our tracker, MCJPDAF tracker, and the integrated tracker are presented in Fig 2. The MCJPDAF tracker failed when the first occlusion occurs in the video. There are some errors in the output of the integrated tracker. Furthermore, it always takes at least 5 frames for the integrated tracker to keep up with the change of the number of the targets. While our tracker can output the right number of players almost immediately (at most $1 \sim 2$ frames delay) through the whole video sequence.

In tracking pedestrians, the detection rate of the trained detector drops to 67% according to our statistics. This makes tracking more difficult, and MCJPDAF fails frequently during the tracking, while our tracker can. The bottom row of Fig 3 shows tracking results of our tracker. More results are available from the video "MTT-pedestrians.avi". We found through the experiments that the number of tracks itself does not make the

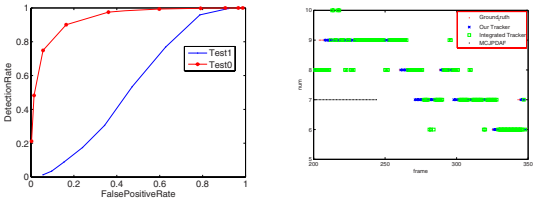


Fig. 2. Left: the ROC curves of the detector; Right:the comparison between the ground-truth and the number of players estimated by our tracker, the MCJPDAF tracker, and the integrated tracker

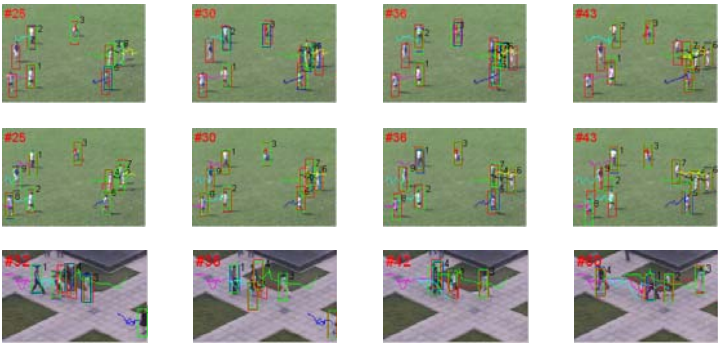


Fig. 3. Tracking performance comparison in dealing with an occlusion. Top row:results of MCJPDAF tracker; Middle row:results of our tracker. Bottom row: pedestrian tracking results of our algorithm. In each frame, the red box, blue box and green box is for the result of the detector, the predicted measurement, and the final result of the tracker, respectively.

problem more difficult if they are scattered apart, but the difficulty arises when there are many targets that are moving closely and crossing each other. Also, solving occlusion problem is the most time-consuming part in our tracker, even the BP can converge within 4 iterations.

7 Conclusion

We have proposed a hierarchical and compositional model based on an And-or graph for joint detecting and tracking of multiple targets in video, and an inference algorithm for the graph. The algorithm consists of three processing channels: (1) a bottom-up channel to provide informative measurements by detectors; (2) a top-down channel to estimate each individual target's state with joint probabilistic data association; (3) a context sensitive reasoning channel to refine the estimation of the joint state with belief propagation. The algorithm is tested by tracking many people in several real video sequences. Despite a constantly varying number of targets and frequent occlusion, the proposed tracker can reliably track many people. Future work will focus on the implementations for real-time MTT applications.

References

1. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. IEEE CVPR 2 (2004)
2. Rittscher, J., Tu, P., Krahnstoever, N.: Simultaneous estimation of segmentation and shape. IEEE CVPR 2 (2005)
3. Tao, H., Sawhney, H., Kumar, R.: A sampling algorithm for tracking multiple objects. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 53–68. Springer, Heidelberg (2000)

4. Isard, M., MacCormick, J.: BraMBLe: A Bayesian multiple-blob tracker. ICCV 2(5) (2001)
5. Schulz, D., Burgard, W., Fox, D., Cremers, A.: People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters. The Intl. Jnl. of Robotics Research 22(2), 99 (2003)
6. Cai, Y., de Freitas, N., Little, J.: Robust visual tracking for multiple targets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 107–118. Springer, Heidelberg (2006)
7. Xue, J.R., Zheng, N.N., Zhong, X.P.: An integrated Monte Carlo data association framework for multi-object tracking. ICPR 01, 703–706 (2006)
8. Zhu, S., Mumford, D.: Quest for a Stochastic Grammar of Images. In: Foundations and Trends in Computer Graphics and Vision (2007)
9. Fridman, A.: Mixed Markov models. Proceedings of the National Academy of Sciences 100(14), 8092–8096 (2003)
10. Xue, J., Zheng, N., Geng, J., Zhong, X.: Multiple Targets Tracking via Particle-based Belief Propagation. IEEE Trans. on SMC-B 38(1) (2008)
11. Yu, T., Wu, Y.: Decentralized multiple target tracking using netted collaborative autonomous trackers. IEEE CVPR 1 (2005)
12. Khan, Z., Balch, T., Dellaert, F.: An MCMC-based particle filter for tracking multiple interacting targets. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 279–290. Springer, Heidelberg (2004)
13. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60(1-4), 259–268 (1992)
14. Comaniciu, D., Ramesh, V.: Real-time tracking of non-rigid objects using mean shift, US Patent 6, 590, 999 (2003)
15. Vermaak, J., Pérez, P.: Monte carlo filtering for multi-target tracking and data association. IEEE Trans. on Aerospace and Electronic Systems 41(1), 309–332 (2005)
16. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of oriented gradients for human detection. In: IEEE CVPR, vol. 1 (2005)
17. Bashir, F., Porikli, F.: Performance evaluation of object detection and tracking systems. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 7–14 (2006)

Heavy-Tailed Model for Visual Tracking via Robust Subspace Learning

Daojing Wang[1], Chao Zhang[1], and Pengwei Hao[1,2]

[1] Key Laboratory of Machine Perception (Ministry of Education)
Peking University, Beijing 1000871, China
[2] Dept. of Computer Science, Queen Mary University of London, London E1 4NS,UK
c.zhang@pku.edu.cn

Abstract. Video-based target tracking, in essence, deals with nonstationary image streams, which is a challenging task in computer vision, because there always appear many abnormal motions and severe occlusions among the objects in the complex real-world environment. In a statistical perspective, an abnormal motion often exhibits non-Gaussian heavy-tailed behavior, which may take a long time to simulate. Most existing algorithms are unable to tackle this issue. In order to address it, we propose a novel tracking algorithm(**HIRPCA**) based on a heavy-tailed framework, which can robustly capture the effect of abnormal motion. In addition, since the conventional PCA is susceptible to outlying measurements in the sense of the least mean squared error minimisation, we extend and improve the incremental and robust PCA to learn a better representation of object appearance in a low-dimensional subspace, contributing to improving the performance of tracking in complex environment, such as light condition, significant pose and scale variation, temporary complete occlusion and abnormal motion. A series of experimental results show the good performance of the proposed method.

1 Introduction

Target tracking in video is an important task in computer vision, and it has many practical applications, such as automated surveillance, video indexing, traffic monitoring and human-computer interaction [1]. In the real-world scenarios, there often may occur appearance variations of an object due to light condition, significant pose and scale variation or complex occlusion. In addition, abnormal motion of moving objects brings a more complicated heavy-tailed issue, which has long been neglected by researchers. There are mainly two kinds of this behavior: camera shot change and rapid motion.

In order to overcome those difficulties, robust subspace learning method and heavy-tailed dynamical model are employed in this paper. First of all, to accurately track object, a good representation of object appearance is rather critical. David et al [2] adopted an incremental PCA to model the appearance information. However, it is susceptible to outliers. In this paper, we extend and improve the incremental and robust PCA introduced in [3], and then extend it to a robust version(IRPCA). IRPCA algorithm is capable of describing a much better appearance. Further more, abnormal motion exhibits a type of non-Gaussian

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 172–181, 2010.

heavy-tailed behavior. Conventional dynamical model can not capture this issue. Hence, it is necessary to introduce a heavy-tailed dynamical model. Our approach(**HIRPCA**) combines the good characteristics of both the heavy-tailed dynamical model and IRPCA method to improve the performance of tracking.

The rest of this paper is organized as follows. In the succeeding section, we begin with a review on several conventional tracking algorithms. Section 3 is devoted to extending and improving both the incremental and the robust PCA proposed by Li [3]. Then the tracker based on Bayesian inference framework is proposed to deal with visual tracking of abnormal appearance changes and abnormal motion. Applications of using the above methods for visual tracking are shown in Section 5. Finally, there is a conclusion section.

2 Overview of Previous Work

Over the past several decades, a large number of algorithms for visual tracking have been addressed. And they can be briefly divided into two categories: detection based method and stochastic based method.

Among the detection-based approaches, Stauffer and Grimson [4] utilized an adaptive Gaussian mixture model to model each pixel of the image for distinguishing the background and foreground. A kernel-based mean shift is presented in [5] for visual tracking of nonrigid objects. Normalized cut method is exploited to segment the graph so as to detect the target in [6].

For the stochastic-based approaches, the particle filter technique has shown its efficiency in tackling non-linear, non-Gaussian and multi-modality problem [7]. Okuma et al [8] developed a boosted particle filter algorithm for tracking a varying number of non-rigid objects . In [9], Kwon and Lee proposed a tracking algorithm based on Wang-Landau Monte Carlo(WLMC) to deal with the abrupt motion, which does not fully consider the drastic changes in appearance. Meanwhile, it is likely to be a search method in the whole image space. A spatial Log-Euclidean appearance model under Log-Euclidean Riemannian metric is presented to capture both the global and local spatial layout information of the target appearance [10] for visual tracking. Heavy-tailed models has been used in [11], however, it is difficult to solve complex optimization by numerical methods. Also, it is limited to tackle complex appearance variation of object.

3 Subspace Learning Method for Tracking

Subspace learning method has been widely applied for visual tracking [2,12,13]. And **Principal Component Analysis**(**PCA**) is one of the most popular subspace learning methods due to its powerful ability of dimensionality reduction. But, the conventional PCA is susceptible to outliers in the sense of the least mean squared error. In [3], only one new observation is considered in its incremental and robust PCA, here, we extend and improve the incremental and robust PCA to be fit for many recently acquired observation images. Also, a forgetting factor technique is incorporated to update the eigenspace.

3.1 The Extended and Improved Incremental PCA

The extended and improved incremental PCA is presented in this subsection by two steps. First, the eigenbasis and the mean are efficiently updated. Second, a forgetting factor is introduced to determine the weights on the previous observation and the new observation.

Incremental Update of Eigenbasis and Mean

For object tracking, it is necessary to adjust the appearance model of the object online, since there are many appearance variations while tracking. Given that a set of images $A = \{Y_1, \ldots, Y_n\}$ have been observed up to time n, we compute the eigenbasis U and the sample mean $\bar{I}_A$ to model the appearance. When additional m images $B = \{Y_{n+1}, \ldots, Y_{n+m}\}$ are newly acquired, it is naive to update the eigenbasis and the sample mean with singular value decomposition (SVD) performed in batch mode.

We begin to present an incremental update by the following lemma introduced in [2]:

Lemma 1. Let $A = \{Y_1, \ldots, Y_n\}$, $B = \{Y_{n+1}, \ldots, Y_{n+m}\}$ be data matrices and $C = [A, B]$ be their concatenation. Denote the means and scatter matrices of A, B, C as $\bar{I}_A$, $\bar{I}_B$, $\bar{I}_C$, and S_A, S_B, S_C respectively. It can be shown that

$$\bar{I}_C = \frac{n}{n+m}\bar{I}_A + \frac{m}{n+m}\bar{I}_B, \quad S_C = S_A + S_B + \frac{nm}{n+m}(\bar{I}_B - \bar{I}_A)(\bar{I}_B - \bar{I}_A)^T. \tag{1}$$

Denote the eigenbasis and eigenvalue matrix of observed images A as U_0 and D_0, the mean-normalized recently acquired observation matrix $B - \bar{I}_B = [Y_{n+1} - \bar{I}_B, \ldots, Y_{n+m} - \bar{I}_B]$, we form the matrix:

$$\Lambda = \Big[\sqrt{\frac{n}{n+m}}U_0 \cdot D_0^{\frac{1}{2}},\ \sqrt{\frac{1}{n+m}}(B - \bar{I}_B),\ \frac{\sqrt{nm}}{n+m}(\bar{I}_B - \bar{I}_A)\Big]. \tag{2}$$

Up to time n , the covariance matrix is $\Sigma_n = U_0 D_0 U_0^T$. With all the new observations, the new covariance matrix can be expressed by

$$\begin{aligned}\Sigma_{n+m} &= \frac{1}{n+m}S_C = \frac{1}{n+m}\big(S_A + S_B + \frac{nm}{n+m}(\bar{I}_B - \bar{I}_A)(\bar{I}_B - \bar{I}_A)^T\big) \\ &= \frac{n}{n+m}\Sigma_n + \frac{1}{n+m}S_B + \frac{nm}{(n+m)^2}(\bar{I}_B - \bar{I}_A)(\bar{I}_B - \bar{I}_A)^T = \Lambda\Lambda^T,\end{aligned} \tag{3}$$

And then eigen-decompose a smaller matrix Δ instead Σ_{n+m} to obtain the eigenbasis and the eigenvalue,

$$\Delta = \Lambda^T \cdot \Lambda \equiv U D U^T. \tag{4}$$

Thus, the update of eigenbasis and eigenvalue of Σ_{n+m} can be calculated respectively:

$$U^{update} = \Lambda \cdot U \qquad D^{update} = D. \tag{5}$$

Forgetting Factor

For accurate visual tracking, it is reasonable to focus more on recently acquired images than previous observations. As time progresses, the appearance of object changes all the time. We have to update the appearance in time by introducing forgetting factor, which is similar to temporal weights along the observation. It is quite straightforward to incorporate the forgetting factor $f \in (0, 1]$ into the proposed incremental algorithm to update the mean and eigenbasis as fellows:

$$\bar{I}_C = \frac{fn}{fn+m}\bar{I}_A + \frac{m}{fn+m}\bar{I}_B, \tag{6}$$

$$\Lambda = \Big[f\sqrt{\frac{n}{n+m}}U_0 \cdot D_0^{\frac{1}{2}},\ \sqrt{\frac{1}{n+m}}(B - \bar{I}_B),\ \frac{\sqrt{nm}}{n+m}(\bar{I}_B - \bar{I}_A)\Big], \tag{7}$$

where if $f < 1$, they will down-weight the contribution of previous observations.

3.2 Robust PCA

For the mean-normalized matrix $B^0 = B - \bar{I}_B$, we define the reconstruction error matrix:

$$\gamma = UU^T B^0 - B^0, \tag{8}$$

then the traditional PCA can be computed as a least squares problem:

$$\min \|\gamma\|^2 = \min \sum_{i,j} \gamma_{i,j}^2. \tag{9}$$

Since the sum of squares is susceptible to outliers, a robust function $\rho(\gamma)$ is introduced to replace it:

$$\min \sum_{i,j} \rho(\gamma_{i,j}). \tag{10}$$

Differentiating Eq.(10) by θ_k, the parameters to be estimated, i.e. the elements of U, we have

$$\sum_{i,j} \omega(\gamma_{i,j})\gamma_{i,j}\frac{\partial \gamma_{i,j}}{\partial \theta_k} = 0. \qquad k = 1, 2, \cdots, \tag{11}$$

where $\omega(t)$ is a weight function, defined as $\omega(t) = \frac{\partial \rho(t)/\partial t}{t}$. Note that, for a fixed weight function $\omega(t)$, Eq.(11) is the solution of the least squares problem:

$$\min \sum_{i,j} \omega(\gamma_{i,j})\gamma_{i,j}^2. \tag{12}$$

Thus, a new PCA problem can be formulated as:

$$\min \sum_{i,j} \|UU^T \hat{B}_{i,j} - \hat{B}_{i,j}\|^2, \tag{13}$$

where $\hat{B}_{i,j} \equiv \sqrt{\omega(\gamma_{i,j})}B_{i,j}^0$. Here, we use a Cauchy weight function $\omega(t)$ for robust analysis, whereas least squares's weight is a constant.

$$\omega(t) = \frac{1}{1 + (t/c)^2}, \tag{14}$$

where parameter c determines the convexity of the function.

Combining the robust analysis with the incremental PCA proposed above, the incremental version of robust PCA (IRPCA)algorithm is presented in Algorithm 1.

Algorithm 1. The incremental Algorithm of Robust PCA

Input: The SVD of observed images and recently acquired observations.
Output: The SVD of all the observation and the mean update.
1. **For** all new observations B **Do**
2. Update the mean vector by Eq.(6);
3. Compute the residual matrix γ by Eq.(8);
4. Compute the weight $\omega(\gamma_{i,j})$ by Eq.(14) for each element of the residual matrix γ;
5. Form matrix Λ by Eq.(7), replacing $B - \bar{I}_B$ by $\hat{B} \equiv (\hat{B}_{i,j}) = (\sqrt{\omega(\gamma_{i,j})} B^0_{i,j})$;
6. Eigen-decompose matrix Δ by Eq.(4) to obtain eigenbasis U and eigenvalue D;
7. Update the eigenbasis and eigenvalue by Eq.(5).
8. **End for**

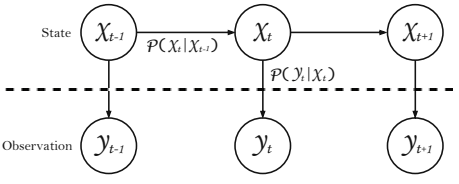


Fig. 1. State space model

4 Bayesian Inference for Visual Tracking

The visual tracking problem is usually considered as a temporal Markov progress (Fig. 1). Given a set of observed images $Y_{1:t} = \{Y_1, \ldots, Y_t\}$ up to time t, by Bayes's formula, the posterior probability can be expressed as:

$$p(X_t|Y_{1:t}) \propto p(Y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y_{1:t-1}) dX_{t-1}, \tag{15}$$

where $p(X_t|X_{t-1})$ represents the dynamical model, and $p(Y_t|X_t)$ is the observation model (likelihood function). Both these two models determine the whole tracking process. We utilize the particle filter technique to obtain the MAP estimate over the N samples of the object's state at each time t

$$X_t^{MAP} = arg \max_{X_t^n} p(X_t^n|Y_{1:t}) \quad for\ n = 1, \ldots, N, \tag{16}$$

where X_t^{MAP} represents the best estimate used to model the current state with the given observation.

4.1 Heavy-Tailed Dynamical Model

Since the abnormal motion exhibits non-Gaussian behavior, conventional Brownian motion often fails to track. A heavy-tailed dynamical model is exploited to represent the transition between consecutive frames:

$$p(X_t|X_{t-1}) = MS(X_t; X_{t-1}, \Omega, n). \tag{17}$$

Where $MS(X, \mu, \Omega, n)$ denotes a multivariate student distribution with mean μ, covariance matrix Ω, and degrees of freedom n, which is one of the most important heavy-tailed distributions(fall off slowly), having more powerful ability of describing abnormal issue than Gaussian. $X_t = (x_t, y_t, \eta_t, s_t, \alpha_t, \phi_t)$, where the six parameters respectively denote the x, y translations, the rotation angle, the scale, the aspect ratio and the skew direction at time t, and they are assumed to be independent.

4.2 IRPCA-Based Observation Model

Many features of object have been utilized to learn a good representation of the object's appearance. In this paper, we employ IRPCA algorithm proposed above to model the appearance information of the object every several frames. Assume that the eigenbasis and the mean are U_t and μ_t at time t, respectively, we can briefly define the likelihood function $p(Y_t|X_t)$ by the reconstruction error norm:

$$p(Y_t|X_t) \propto exp(-\|U_t U_t^T (Y_t - \mu_t) - (Y_t - \mu_t)\|^2 / 2\sigma^2), \tag{18}$$

where σ is a parameter. The smaller the reconstruction error norm, the larger the likelihood function. Up to now, our novel tracking algorithm(**HIRPCA**) has been presented.

5 Implementation and Experiments

In order to evaluate the performance of the proposed method, we employ six videos totally more than 3000 frames in different complex environments. These videos almost cover all the common difficulties for visual tracking, such as small target, low-resolution image, large illumination variation, complex noise, temporary severe occlusion, abrupt motion, camera shot change and significant pose variation. Implemented in MATLAB, the HIRPCA can efficiently and accurately track target at almost 2 fps for 320×240 videos on a Pentium 4 3GHz computer. Note that our code is not optimized.

We begin with the results of HIRPCA tracker, and then compare it with three state-of-the-art tracking algorithm: IPCA[2], Mean Shift[5] and WSL[14] by 600 samples($N = 600$).

5.1 Experimental Results

First, our tracker is applied to a video of a vehicle driving in a complex night environment(Fig. 2), and the size of the target is only about 30×30, much smaller than the image. HIRPCA can accurately track the vehicle over the whole low-resolution video.

The second experiment is conducted to a video, where a vehicle moves underneath an overpass and trees. In the whole video, the light condition has largely

Fig. 2. A small vehicle moving at night with large illumination changes

Fig. 3. A vehicle moving underneath an overpass and trees in a complex noise and varying illumination condition

changed at times. Besides, while the vehicle goes through the overpass(between frame #190 and frame #234), there is complex noise as shown in Fig. 3(especially frame #209). HIRPCA is able to follow the vehicle very well owing to the incremental algorithm of robust PCA.

The third image sequence, shown in Fig. 4, a hockey player runs rapidly with abnormal motion. At the same time, there are many temporary severe occlusions between frame #38 and frame #45, and those hockey players are rather similar to each other. The HIRPCA based on heavy-tailed motion is capable of capturing the abnormal issue, while the incremental PCA efficiently describes the appearance of object.

In the forth experimental video, a doll moving in different pose, scale, and lighting conditions is used to test the validity and robustness of HIRPCA as shown in Fig. 5. Despite it experiences significant pose variation (#66, #102, #271, #384, #614, #920, #1091, #1149), scale change(#200, #212, #227) and varying illumination(#545, #772, #787), HIRPCA performs well.

The results of these four videos also empirically validate that our HIRPCA has a powerful ability of modeling the appearance of object nearly in arbitrary environment.

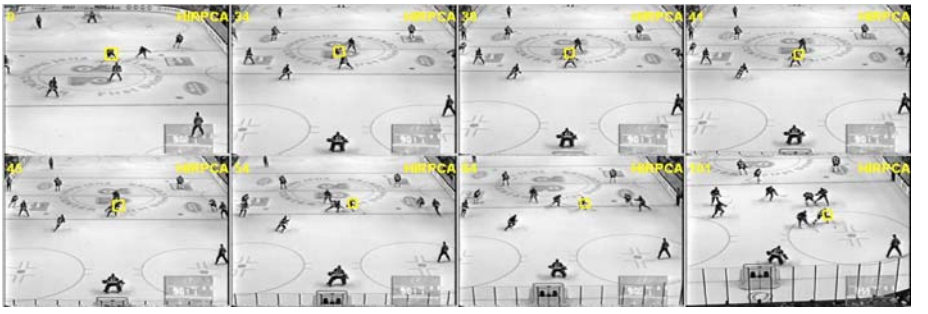


Fig. 4. A hockey player rapidly moving with temporary severe occlusion among many similar objects

Fig. 5. An animal doll moving with significant pose, lighting and scale variation

5.2 Qualitative Comparison

Fig. 6 and Fig. 7 show the comparison for tracking objects in the two camera videos. HIRPCA successfully tracked the target, as robustly as WSL [14], while IPCA [2] tracker fails after frame #614 and #323, respectively, as a result of a combination of significant pose and illumination changes. Thus, we can say that our robust PCA has a more compact low-dimensional representation of objects, which can efficiently improve the performance of visual tracking. On the other hand, Mean Shift [5] frequently fails to track(Fig. 6: #75, #274, #358 and Fig. 7: #160, #309), because appearance model of this tracker does not update in time.

Fig. 8 presents the tracking comparison in the camera shot change case. Video sequence with camera shot change is a new interested difficulty for visual tracking [9], where a sampling method has been used. Our tracker based on heavy-tailed dynamical model can partly solve it (#1 → #2 and #248 → #249) whereas IPCA [2] can not accurately track(#249). Even it fails to track when motion is smooth(#325) as a result of error accumulation.

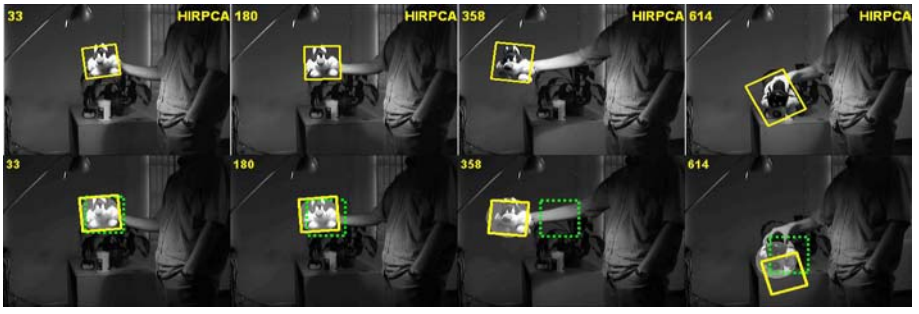


Fig. 6. Tracking comparison between our tracker (row 1) and IPCA [2] (yellow box), the WSL [14] (highlighted ellipse) and the Mean Shift [5] (green dashed box) (row 2) with a doll video sequence

Fig. 7. Tracking comparison between our tracker (row 1) and IPCA [2] (yellow box), the WSL [14] (highlighted ellipse) and the Mean Shift [5] (green dashed box) (row 2) with a large illumination change video sequence

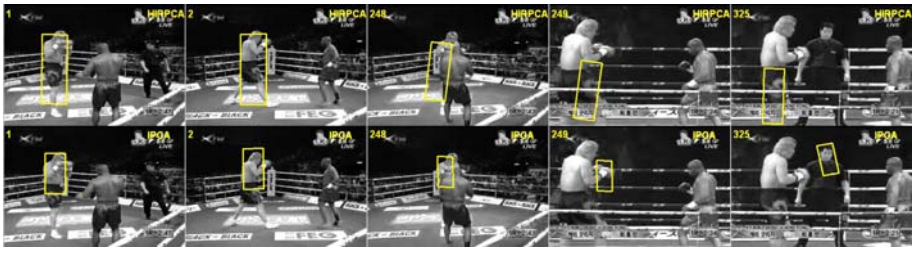


Fig. 8. Tracking comparison between our tracker (row 1) and IPCA [2] (row 2) with a camera shot change video sequence provided by Kwon [9]

6 Conclusions and Future Work

In this paper, we extend and improve the incremental and robust PCA. Based on the proposed HIRPCA method and heavy-tailed dynamical model, a unified visual tracking framework is built to overcome several types of tracking difficulties, small target, low-resolution image, significant pose, scale and illumination

variation, complex noise, temporary severe occlusion, abrupt motion, camera shot change and so on. The experiments have demonstrated that our tracker performs very well. However, since only local spatial information is considered, our tracker can not completely solve the tracking problem for video sequences with camera shot change. In future, we will focus on other techniques using global spatial information to capture abnormal motion.

Acknowledgements. This work was supported by research funds of NSFC No.60572043 and the NKBRPC No.2004CB318005.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. 38(4) (2007)
2. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77(2), 125–141 (2008)
3. Li, Y.: On incremental and robust subspace learning. Pattern Recognition 37, 1509–1518 (2004)
4. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 246–252 (1999)
5. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans.Pattern Anal. Mach. Intell. 25(5), 564–577 (2003)
6. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
7. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
8. Okuma, K., Taleghani, A., Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
9. Kwon, J., Lee, K.M.: Tracking of abrupt motion using wang-landau monte carlo estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 387–400. Springer, Heidelberg (2008)
10. Li, X., Hu, W., Zhang, Z., Zhang, X.: Robust visual tracking based on an effective appearance model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 396–408. Springer, Heidelberg (2008)
11. Loxam, J., Drummond, T.: Student-t mixture filter for robust, real-time visual tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 372–385. Springer, Heidelberg (2008)
12. Ho, J., Lee, K., Yang, M., Kriegman, D.: Visual tracking using learned linear subspaces. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 782–789 (2004)
13. Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Robust visual tracking based on incremental tensor subspace learning. In: International Conference on Computer Vision (ICCV), pp. 1–8 (2007)
14. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans.Pattern Anal. Mach. Intell. 25(10), 1296–1311 (2003)

Efficient Scale-Space Spatiotemporal Saliency Tracking for Distortion-Free Video Retargeting

Gang Hua[1], Cha Zhang[2], Zicheng Liu[2], Zhengyou Zhang[2], and Ying Shan[1]

[1] Microsoft Corporation
[2] Microsoft Research
{ganghua,chazhang,zliu,zhang,yingsh}@microsoft.com

Abstract. Video retargeting aims at transforming an existing video in order to display it appropriately on a target device, often in a lower resolution, such as a mobile phone. To preserve a viewer's experience, it is desired to keep the important regions in their original aspect ratio, i.e., to maintain them distortion-free. Most previous methods are susceptible to geometric distortions due to the anisotropic manipulation of image pixels. In this paper, we propose a novel approach to distortion-free video retargeting by scale-space spatiotemporal saliency tracking. An optimal source cropping window with the target aspect ratio is smoothly tracked over time, and then isotropically resized to the retargeted display. The problem is cast as the task of finding the most spatiotemporally salient cropping window with minimal information loss due to resizing. We conduct the spatiotemporal saliency analysis in scale-space to better account for the effect of resizing. By leveraging integral images, we develop an efficient coarse-to-fine solution that combines exhaustive coarse and gradient-based fine search, which we term scale-space spatiotemporal saliency tracking. Experiments on real-world videos and our user study demonstrate the efficacy of the proposed approach.

1 Introduction

Video retargeting aims at modifying an existing video in order to display it appropriately on a target display of different size and/or different aspect ratio [1,2,3]. The vast majority of the videos captured today have 320×240 pixels or higher resolutions and standard aspect ratio 4:3 or 16:9. In contrast, many mobile displays have low resolution and non-standard aspect ratios. Retargeting is hence essential to video display on these mobile devices. Recently, video retargeting has been applied in a number of emerging applications such as mobile visual media browsing [3,4,5,6], automated lecture services [7], intelligent video editing [8,9], and virtual directors [10,7].

In this work, we focus on video retargeting toward a smaller display, such as that of a mobile phone. Directly resizing a video to the small display may not be desirable, since by doing so we may either distort the video scene, which is visually disturbing, or pad black bars surrounding the resized video, which wastes precious display resources. To bring the best visual experiences to the users, a good retargeted video should preserve as much the visual content in the original video as possible, and it should ideally be distortion-free. To achieve this goal, we need to address two important problems: 1) how to quantify the importance of visual content? 2) How to preserve the visual content while ensuring distortion-free retargeting?

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 182–192, 2010.

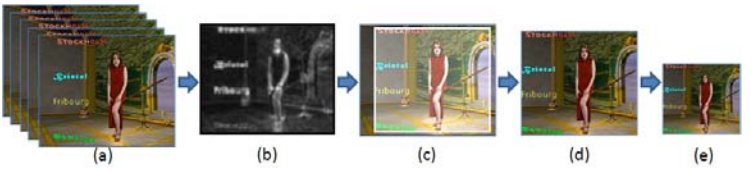


Fig. 1. Retargeting system overview: scale-space spatiotemporal saliency map (b) is calculated from consecutive n video frames (a). A minimal information loss cropping window with the target aspect ratio is identified via smooth saliency tracking (c), and the cropped image (d) is isotropically scaled to the target display (e). This example retargets 352×288 images to 100×90.

Previous works [11,12,1,4,2] approach to the first problem above by combining multiple visual cues such as image gradient, optical flow, face and text detection results etc. in an ad hoc manner to represent the amount of content information at each pixel location (a.k.a. the saliency map). It is desirable to have a simple, generic and principled approach to accounting for all these different visual information. In this paper, we improve and extend the spectrum residue method for saliency detection in [13] to incorporate temporal and scale-space information, and thereby obtain a *scale-space spatiotemporal saliency map* to represent the importance of visual content.

Given the saliency map, retargeting should preemptively preserve as many salient image pixels as possible. Liu and Gleicher [1] achieve this by identifying a cropping window which contains the most visual salient pixels and then anisotropically scale it down to fit with the retargeting display (i.e., allowing different scaling in horizontal and vertical directions). The cropping window is restricted to be of fixed size within one shot, and the motion of the cropping window can only be one of the three types, i.e., static, horizontal pan, or a virtual cut. It can not perform online live retargeting since the optimization must be performed at the shot level. Avidan and Shamir [11] use dynamical programming to identify the best pixel paths to perform recursive cut or interpolation for image resizing. Wolf et. al [2] solve for a saliency aware global warping of the source image to the target display size, and then resample the warped image to the target size. Nevertheless, it is not uncommon for all the aforementioned methods to introduce geometry distortions to the video objects due to the anisotropic manipulation of the image pixels.

In this paper, we propose to smoothly track an optimal cropping window with the target aspect ratio across time, and then isotropically resize it to fit with the target display. Our approach is able to perform online retargeting. We propose an efficient coarse-to-fine search method, which combines coarse exhaustive search and gradient based fine search, to track an optimal cropping window over time. Moreover, we only allow isotropic scaling during retargeting, and therefore guarantee that the retargeted video is distortion-free. An overview of our retargeting system is presented in Fig. 1.

There are two types of information loss in the proposed retargeting process. First, when some regions are excluded due to cropping, the information that they convey are lost. We term this the *cropping information loss*. Second, when the cropped image is scaled down, details in the high frequency components are thrown away due to the low pass filtering. This second type of loss is called the *resizing information loss*. One may

always choose the largest possible cropping window, which induces the smallest cropping information loss, but may potentially incur huge amount of resizing information loss. On the other hand, one can also crop with exactly the target display size, which is free of resizing information loss, but may result in enormous cropping information loss. Our formulation takes both of them into consideration and seeks for a trade-off between the two. An important difference between our work and [1] is that the resizing information loss we introduce is *content dependent*, which is based on the general observation that some images may be downsized much more than some other images without significantly degrading their visual quality. This is superior to the naive content independent scale penalty (a cubic loss function) adopted in [1].

The main contributions of this paper therefore reside in three-fold: **1)** we propose a distortion-free formulation for video retargeting, which yields to a problem of *scale-space spatiotemporal saliency tracking*. **2)** By leveraging integral images, we develop an efficient solution to the optimization problem, which combines a coarse exhaustive search and a novel gradient-based fine search for scale-space spatiotemporal saliency tracking. **3)** We propose a computational approach to scale-space spatiotemporal saliency detection by joint frequency, scale space, and spatiotemporal analysis.

2 Distortion-Free Video Retargeting

2.1 Problem Formulation

Consider an original video sequence with T frames $\mathcal{V} = \{I_t, t = 1, \cdots, T\}$. Each frame is an image array of pixels $I_t = \{I_t(i,j), 0 \leq i < W_0, 0 \leq j < H_0\}$, where W_0 and H_0 are the width and height of the images. For retargeting, the original video has to be fit into a new display of size $W_r \times H_r$. We assume $W_r \leq W_0, H_r \leq H_0$.

To ensure that there is no distortion during retargeting, we allow only two operations on the video – cropping and isotropic scaling. Let $\mathcal{W} = \{(x,y),(W,H)\}$ be a rectangle region in the image coordinate system, where (x,y) is the top-left corner, and W and H are the width and the height. The cropping operation on frame I_t can be defined as $\mathcal{C}_{\mathcal{W}}(I_t) \triangleq \{I_t(m+x, n+y), 0 \leq m < W, 0 \leq n < H\}$, where m and n are the pixel index of the output image. The isotropic scaling operation is parameterized with a single scalar variable s (for scaling down, $1.0 \leq s \leq s_{max}$), i.e., $\mathcal{S}_s(I_t) \triangleq \{I_t(s \cdot m, s \cdot n), s \cdot m < W_0, s \cdot n < H_0\}$. Distortion-free video retargeting can be represented as a composite of these two operations on all the video frames such that $\hat{I}_t(s_t, x_t, y_t) = \mathcal{S}_{s_t}(\mathcal{C}_{\mathcal{W}_t}(I_t)), t = 1, \cdots, T$, where $\mathcal{W}_t = \{(x_t, y_t), (s_t W_r, s_t H_r)\}$ is the cropping window at frame I_t. We further denote $\hat{\mathcal{V}} = \{\hat{I}_t, t = 1, \cdots, T\}$ to be the retargeted video, and $\mathcal{P} \triangleq \{(s_t, x_t, y_t), t = 1, \cdots, T\}$ to be the set of unknown scaling and cropping parameters, where $\mathcal{P} \in \mathfrak{R} = \{s_t, x_t, y_t | 1.0 \leq s_t \leq s_{max}, 0 \leq x_t < W_0 - s_t W_r, 0 \leq y_t < H_0 - s_t H_r\}$.

Both cropping and scaling will lead to information loss from the original video. We propose to exploit the information loss with respect to the original video as the cost function for retargeting, i.e.:

$$\mathcal{P}^* = \arg\max_{\mathcal{P} \in \mathfrak{R}} \mathbf{L}(\mathcal{V}, \hat{\mathcal{V}}), \tag{1}$$

where $\mathbf{L}(\mathcal{V}, \hat{\mathcal{V}})$ is the information loss function, which shall be detailed in Sec. 2.2. Since ensuring the smooth transition of the cropping and resizing parameters is essential to the visual quality of the retargeted video, we also introduce a few motion constraints that shall be included when optimizing Eq. (1) in Sec. 2.3.

2.2 Video Information Loss

The *cropping* and *resizing* information loss are caused by very different reasons, hence they can be computed independently. We represent the video information loss function with two terms, i.e.,

$$\mathbf{L}(\mathcal{V}, \hat{\mathcal{V}}) = \mathbf{L}_c(\mathcal{V}, \hat{\mathcal{V}}) + \lambda \mathbf{L}_r(\mathcal{V}, \hat{\mathcal{V}}), \tag{2}$$

where λ is the control parameter to obtain a tradeoff between the cropping information loss $\mathbf{L}_c$ and the resizing information loss $\mathbf{L}_r$, which are detailed as follows.

Cropping information loss. We compute the cropping information loss based on spatiotemporal saliency maps. We assume in this section such a saliency map is available (see Sec. 4 for our computation model for the spatiotemporal saliency map).

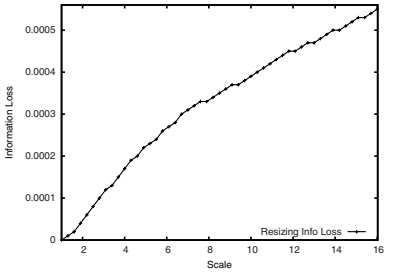


Fig. 2. Resizing information loss curve

For frame I_t, we denote the per-pixel saliency map as $\{S_t(i,j), 0 \leq i < W_0, 0 \leq j < H_0\}$. Without loss of generality, we assume that the saliency map is normalized such that $\sum_{ij} S_t(i,j) = 1$. Given $\mathcal{W}_t$, the cropping information loss at time instant t is defined as the summation of the saliency values of those pixels left outside the cropping window, i.e.,

$$\mathbf{L}_c(\mathcal{W}_t) = 1 - \sum_{(i,j)\in\mathcal{W}_t} S_t(i,j). \tag{3}$$

The cropping information loss between the original video and the retargeted video is thereby defined as $\mathbf{L}_c(\mathcal{V}, \hat{\mathcal{V}}) = \sum_{t=1}^{T} \mathbf{L}_c(\mathcal{W}_t) = T - \sum_{t=1}^{T} \sum_{(i,j)\in\mathcal{W}_t} S_t(i,j)$.

Resizing information loss. The resizing information loss $\mathbf{L}_r(\mathcal{V}, \hat{\mathcal{V}})$ measures the amount of details lost during scaling, where low-pass filtering is necessary in order to avoid aliasing in the down-sampled images. For a given frame I_t, the larger the scaling factor s_t, the more aggressive the low-pass filter has to be, and the more details will be lost due to scaling. In the current framework, the low-pass filtered image is computed as $I_{s_t} = \mathcal{G}_{\sigma(s_t)}(I_t)$, where $\mathcal{G}_\sigma(\cdot)$ is a 2D Gaussian low-pass filter with isotropic covariance σ, which is a function of the scaling factor s_t, i.e., $\sigma(s_t) = \log_2(s_t), 1.0 \leq s_t \leq s_{max}$. The resizing information loss is defined as the *squared error* between the cropped image in the original resolution and its low-pass filtered image before down-sampling, i.e.,

$$\mathbf{L}_r(\mathcal{W}_t) = \sum_{(i,j)\in\mathcal{W}_t} (I_t(i,j) - I_{s_t}(i,j))^2. \tag{4}$$

The image pixel values are normalized to be in $[0,1]$ beforehand. For the whole video sequence, we have $\mathbf{L}_r(\mathcal{V}, \hat{\mathcal{V}}) = \sum_{t=1}^{T} \mathbf{L}_r(\mathcal{W}_t) = \sum_{t=1}^{T} \sum_{(i,j)\in\mathcal{W}_t} (I_t(i,j) - I_{s_t}(i,j))^2$.

Fig. 2 presents the resizing information loss curve calculated for the cropping window presented in Fig. 1(c) using Eq. 4. As we expected, the loss function increases monotonically with the increase of the scaling factor.

2.3 Constraints for Video Retargeting

If there is no other additional cross-time constraints, Eq. 1 can indeed be optimized frame by frame. However, motion smoothness constraints of the cropping window, for both scaling and translation, is very important to produce visually pleasant retargeted video. To ease the optimization, we do not model motion constraints directly in our cost function. Instead we pose additional smoothness constraints on the solution space of $\mathcal{P}$ at each time instant t, i.e., the optimal $\mathcal{W}_t$ is constrained by the optimal solutions of $\mathcal{W}_{t-1}$ and $\mathcal{W}_{t-2}$. By doing so, an additional benefit is that retargeting can be performed online. Mathematically, we have

$$\left|\frac{\partial s_t}{\partial t}\right| \leq v^z_{\max}, \left\|\left(\frac{\partial x_t}{\partial t}, \frac{\partial y_t}{\partial t}\right)\right\| \leq v_{\max}, \left|\frac{\partial^2 s_t}{\partial t^2}\right| \leq a^z_{\max}, \left\|\left(\frac{\partial^2 x_t}{\partial t^2}, \frac{\partial^2 y_t}{\partial t^2}\right)\right\| \leq a_{\max} \quad (5)$$

where $v^z_{\max}$, $v_{\max}$, $a^z_{\max}$ and $a_{\max}$ are the maximum zooming and motion speed, and the maximum zooming and motion acceleration during cropping and scaling, respectively. Such first and second order constraints ensure that the view movement of the retargeted video is small, and ensure that there is no abrupt change of motion or zooming directions. They are both essential to the aesthetics of the retargeted video. Additional constraints may be derived from rules suggested by professional videographers [7]. It is our future work to incorporate these professional videography rules.

3 Detecting and Tracking Salient Regions

We develop a two stage coarse-to-fine strategy for detecting and tracking salient regions, which is composed of an efficient exhaustive coarse search, and a gradient-based fine search as well. Since this two stage search process is performed at each time instant, to simplify the notation and without sacrificing clarity, we shall leave out the subscript t for some equations in the rest of this section.

Both search processes are facilitated by integral images, we employ the following notations for the integral image [14] of the saliency image $\mathcal{S}(x, y)$ and its partial derivatives, i.e., $\mathcal{T}(x, y) = \int_0^x \int_0^y \mathcal{S}(x, y)dxdy$, $\mathcal{T}_x(x, y) = \frac{\partial \mathcal{T}}{\partial x} = \int_0^y \mathcal{S}(x, y)dy$, and $\mathcal{T}_y(x, y) = \frac{\partial \mathcal{T}}{\partial y} = \int_0^x \mathcal{S}(x, y)dx$. All these integral images can be calculated very efficiently by accessing each image pixel only once. We further denote $\hat{x}(x, s) = x + sW_r$, and $\hat{y}(y, s) = y + sH_r$. Using $\mathcal{T}(x, y)$, the cropping information loss can be calculated in constant time, i.e., $\mathbf{L}_c(s, x, y) = 1 - (\mathcal{T}(\hat{x}, \hat{y}) + \mathcal{T}(x, y)) - (\mathcal{T}(\hat{x}, y) + \mathcal{T}(x, \hat{y}))$.

The calculation of the resizing information loss can also be speeded up greatly using integral images. We introduce the squared difference image $D_s(x, y)$ for scaling by s as $D_s(x, y) = (I(x, y) - I_s(x, y))^2$. We then also define the integral images of $D_s(x, y)$ and its partial derivatives, which are denoted as $\mathcal{D}^s(x, y)$, $\mathcal{D}^s_x(x, y)$, and $\mathcal{D}^s_y(x, y)$. We immediately have $\mathbf{L}_r(s, x, y) = (\mathcal{D}^s(\hat{x}, \hat{y}) + \mathcal{D}^s(x, y)) - (\mathcal{D}^s(\hat{x}, y) + \mathcal{D}^s(x, \hat{y}))$. In run

time, we keep a pyramid of the integral images of $D^s(x, y)$ for multiple s. Since both $\mathbf{L}_c$ and $\mathbf{L}_r$ can be calculated in constant time, we are able to afford the computation of an exhaustive coarse search over the solution space for the optimal cropping window.

Once we have coarsely determined the location of a cropping window $\mathcal{W}$, we further exploit a gradient-based search to refine the optimal cropping window. By simple chain rules, it is easy to figure out that $\frac{\partial \mathbf{L}}{\partial a} = \mathcal{T}_a(\hat{x}, y) + \mathcal{T}_a(x, \hat{y}) - \mathcal{T}_a(x, y) - \mathcal{T}_a(\hat{x}, \hat{y}) + \lambda[\mathcal{D}_a^s(\hat{x}, y) + \mathcal{D}_a^s(x, \hat{y}) - \mathcal{D}_a^s(x, y) - \mathcal{D}_a^s(\hat{x}, \hat{y})]$, for $a = x$ or $a = y$, and $\frac{\partial \mathbf{L}}{\partial s} = A(x, y, s)W_r + B(x, y, s)H_r + \lambda \frac{\partial \mathbf{L}_r}{\partial s}$, where $A(x, y, s) = \mathcal{T}_x(\hat{x}, y) - \mathcal{T}_x(\hat{x}, \hat{y})$, $B(x, y, s) = \mathcal{T}_y(x, \hat{y}) - \mathcal{T}_y(\hat{x}, \hat{y})$, $\frac{\partial \mathbf{L}_r(x,y,s)}{\partial s} = \frac{\mathbf{L}_r(x,y,s+\triangle s) - \mathbf{L}_r(x,y,s-\triangle s)}{2\triangle s}$ is evaluated numerically. Then we perform a gradient descent step with backtracking line search to refine the optimal cropping window. Note that the gradient descent step is also very efficient because all derivatives can be calculated very efficiently using integral images and its partial derivatives. This two-step coarse-to-fine search ensures us to obtain the optimal cropping window very efficiently.

The feasible solutions $\Omega_t = \{[x_t^{min},\ x_t^{max}],\ [y_t^{min}, y_t^{max}],\ [s_t^{min},\ s_t^{max}]\}$ are derived from Eqs. 5 and strictly reenforced in tracking. Denote $\mathcal{W}_{t-1}^* = (x_{t-1}^*, y_{t-1}^*, s_{t-1}^*)$ be the optimal cropping at the time instant $t-1$, and let the optimal cropping window after these two stage search process at time instant t be $\hat{\mathcal{W}}_t$, we perform an exponential moving average scheme to further smooth the parameters of the cropping window, i.e., $\mathcal{W}_t^* = \alpha\hat{\mathcal{W}}_t + (1-\alpha)\mathcal{W}_{t-1}^*$. We use $\alpha = 0.7 \sim 0.95$ in the experiments. It in general produces visually smooth and pleasant retargeted video, as shown in our experiments.

4 Scale-Space Spatiotemporal Saliency

We propose several extensions of the spectrum residue method for saliency detection proposed by Hou and Liu [13]. We refer the readers to [13] for the details of their algorithm. Fig. 4(a) presents one result of saliency detection using the spectrum residue method proposed in [13]. On one hand we extend the spectrum residue method temporally, and on the other hand, we extend it in scale-space. The justification of our temporal extension may largely be based on the statistics of optical flows in natural images revealed by Roth and Black [15], which shares some common characteristics with the natural image statistics. It is also revealed by Hou and Liu [13] that when applying the spectrum residue method to different scales of the same image, different salient objects of different scales will pop out. Since for retargeting, we would want to retain salient object across different scales, we aggregate the saliency results from multiple scales together to achieve that.

Moreover, we also found that it is the phase spectrum [16] which indeed plays the key role for saliency detection. In other words, if we replace the magnitude spectrum residue with constant 1, the resulted saliency map is almost the same as that calculated from the spectrum residue method. We call such a modified method to be the *phase spectrum* method for saliency detection. The difference of the resultant saliency maps is almost negligible but it saves significant computation to avoid calculating the magnitude spectrum residue, as we clearly demonstrate in Fig. 4. Fig. 4(a) is the saliency map obtained from the spectrum residue and Fig. 4(b) is the saliency map produced from the phase spectrum only. Note the source image from which these two saliency maps are

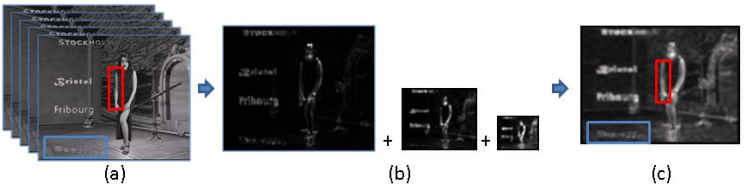


Fig. 3. Scale-space spatiotemporal saliency detection

generated is presented as the top image in Fig. 3(a). The difference is indeed tiny. This is a common phenomenon that has been verified constantly in our experiments.

More formally, let $\mathcal{V}_t^n(i,j,k) = \{I_{t-n+1}(i,j), I_{t-n+2}(i,j), \ldots, I_t(i,j)\}$ be a set of n consecutive image frames and k indexes the image. Denote $\mathbf{f} = (f_1, f_2, f_3)$ as the frequencies in the fourier domain, where (f_1, f_2) represents spatial frequency and f_3 represents temporal frequency. The following steps are performed to obtain the spatiotemporal saliency map for $\mathcal{V}_t^n$:

1. Let $\Theta(\mathbf{f}) = Pha(\mathfrak{F}[\mathcal{V}_t^n])$ be the phase spectrum of the 3D FFT of $\mathcal{V}_t^n$.
2. Perform the inverse FFT and smoothing, i.e., $\mathbf{S}_t(i,j,k) = g(i,j) * \mathfrak{F}^{-1}[\exp\{j\Theta(\mathbf{f})\}]^2$. The smoothing kernel $g(i,j)$ is applied only spatially, since the temporal information will be aggregated.
3. Combine $\mathbf{S}(i,j,k)$ to be one single map, i.e., $\mathcal{S}_t(i,j) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{S}_t(i,j,k)$

The above steps present how to compute the spatiotemporal saliency map at a single scale. We aggregate the visual saliency information calculated from multiple scales together, this leads to the *scale-space spatiotemporal saliency*. More formally, let $\mathcal{V}_t^n(s)$ be the down-sampled version of $\mathcal{V}_t^n$ by a factor of s, i.e., each image in $\mathcal{V}_t^n$ is down-sampled by a factor of s in $\mathcal{V}_t^n(s)$. Denote $\mathcal{S}_t^s(i,j)$ as the spatiotemporal saliency image calculated from $\mathcal{V}_t^n(s)$ based on the algorithm presented above. We finally aggregate the saliency map across different scales together, i.e.,$\mathcal{S}_t(i,j) = \frac{1}{n_s}\sum_s S_t^s(i,j)$, where n_s is the total number of levels of the pyramid. Fig. 3 presents the results of using the proposed approach to scale-space spatiotemporal saliency detection. The current image frame is the top one showing in Fig 3(a). We highlight the differences between the scale-space spatiotemporal saliency image (Fig. 3(c)) and the saliency maps (Fig. 4(a) and (b)) produced by the spectrum residue method [13] and the phase spectrum method, using color rectangles.

The proposed method successfully identified the right arm (*the red rectangle*) of the singer as a salient region, while the saliency map in Fig. 4(a) and (b) failed to achieve that. The difference comes from the scale-space spatiotemporal integration (the arm is moving) of saliency information. Moreover, in the original image, the gray level of the string in the blue rectangle is very close to the background. It is very difficult to detect its saliency based only on one image (Fig. 4 (b)). Since the string is moving, the proposed method still successfully identified it as a salient region (Fig. 3 (c)).

5 Experiments

The proposed approach is tested on different videos for various retargeting purpose, including both standard MPEG-4 testing videos and a variety of videos downloaded

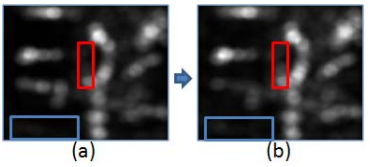


Fig. 4. Saliency detection using (a) spectrum residue [13], and (b) phase spectrum. The source image is shown in Fig. 3(a).

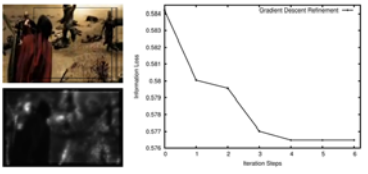


Fig. 5. Left column: the source image and its saliency map. Right column: the progress of the gradient search.

Fig. 6. Retargeting from 368×240 to 132×120 for movie video "300". The first four columns present the saliency tracking results and the corresponding saliency map. The fifth column shows our retargeting results. The sixth column shows the results by directly scaling.

from the Internet. All experiments are running with $\lambda = 0.3$ in Eq.2, which is empirically determined to achieve a good tradeoff. Furthermore, $n = 5$ video frames and an $n_s = 3$ level pyramid are used to build the scale-space spatiotemporal saliency map. We recommend the readers to look into the supplemental video for more details of our experimental results.

Spatiotemporal saliency tracking. To better understand the proposed approach to scale-space spatiotemporal saliency detection and tracking, we show a retargeting example on a video sequence from the battle scene of the movie "300". The video sequence has 1695 frames in total, we present some sample results in Fig. 6. As we can clearly see, the proposed saliency detection and tracking algorithms successfully locked onto the most salient regions. The fifth column of Fig. 6 presents our retargeting results. For comparison, the sixth column of Fig. 6 shows the results of directly resizing the original image frame to the target size. It is clear that in our retargeting results, the objects look not only larger but also keep their original aspect ratios even though the image aspect ratio changed from 1.53 to 1.1. To demonstrate the effectiveness of the gradient-based refinement step, we present the intermediate results of the gradient search at frame #490 in in Fig. 5.

Content-aware v.s. content independent resizing cost. One fundamental difference between our approach and Liu and Gleicher [1] is that our resizing cost (Eq. 4) is dependent on the content of the cropped image. In contrast Liu and Gleicher only adopt an naive cubic loss $(s-1.0)^3$ to penalize large scaling. To better understand the difference,

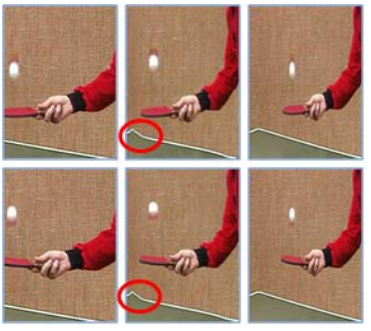


Fig. 7. Retargeting MPEG-4 standard test sequence "tennis". From left to right: **first column**–our approach, **second column**– Wolf et. al[2]'s method (by courtesy), **third column**– direct scaling.

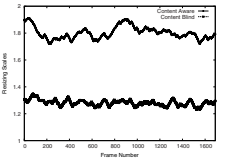


Fig. 8. The scaling factors associated with each video frame of the retargeting video "300". **Top**: content aware; **Bottom**: content blind.

Fig. 9. Retargeting MPEG-4 standard test sequence "Akiy" to be half of its original size: (a) direct scaling; (b) proposed approach; (c) Wolf et. al [2] (by courtesy)

we implemented a different retargeting system by replacing Eq. 4 with the naive cubic loss. The other steps remain the same. Therefore the differences in results are solely decided by the two different resizing costs. We call them *content aware* scheme and *content blind* scheme, respectively.

We analyze the behaviors of the two methods based on the retargeting results of "300" video. Both cost values are normalized to be between 0 and 1 for fair comparison. For the content blind scheme, the λ is empirically determined on this video to be 0.2 for the best retargeting result. All other parameters are the same for the two methods. The curves in the upper and lower part of Fig. 8 present the scaling parameters from content aware resizing, and content blind resizing across the video, respectively.

It is clear that the content blind loss strongly favors small scaling. This bias may be very problematic because of the potentially large cropping information loss. In contrast, the content aware resizing does not have such a bias and also shows much larger dynamic range. This indicates that it is more responsive to capture the video content change. To achieve good results, we find that for the content blind scheme, the λ needs to be carefully tuned for each video, and its variance is large across different videos. In contrast, for the content aware scheme, a constant $\lambda = 0.3$ usually works well.

Video re-targeting results. We tested the proposed approach in a wide variety of long range video sequences for different retargeting tasks. Here we only present the retargeting result on the standard MPEG-4 test video sequence "tennis". (More experiments are described in [17].) We re-target the source video to 176×240. The retargeted results from our approach on frame $\#10$ and $\#15$ are shown in the first column of Fig. 7. For comparison, we also present the retargeting results from Wolf et. al[2][1], and the results by direct scaling, in the second and third columns of Fig. 7, respectively. Due to the nonlinear warping of image pixels in Wolf et. al's method [2], visually

[1] We thank Prof. Lior Wolf and Moshe Guttmann for their result figures.

disturbing distortion appears, as highlighted by the red circles in Fig. 7. In Fig. 9, we further compare our results with Wolf et. al [2] on the standard MPEG-4 testing video "Akiy". The task is to re-target the original video down to half of its original width and height. As we can clearly observe, the retargeted result from Wolf et. al [2] (Fig. 9 (c)) induces heavy nonlinear distortion, which makes the head size of the person in the video to be unnaturally big compared to her body size. In contrast, the result from the proposed approach keeps the original relative size and distortion free. Moreover, compared with the result from the direct scaling method in Fig. 9 (a), our result shows more details of the broadcaster's face when presented in a small display.

More experiments, including the details of a user study with 30 subjects, can be found in our Tech Report [17].

6 Conclusion and Future Work

We proposed a novel approach to distortion-free video retargeting by scale-space spatiotemporal saliency tracking. Extensive evaluation on a variety of real world videos demonstrate the good performance of our approach. Our user study also provide strong evidences that users prefer the retargeting results from the proposed approach. Future works may include further investigating possible means of integrating more professional videography rules into the proposed approach.

References

1. Liu, F., Gleicher, M.: Video retargeting: automating pan and scan. In: Proc. ACM international conference on Multimedia, pp. 241–250. ACM, New York (2006)
2. Wolf, L., Guttmann, M., Cohen-Or, D.: Non-homogeneous content-driven video-retargeting. In: Proceedings IEEE International Conference on Computer Vision (2007)
3. Setlur, V., Takagi, S., Raskar, R., Gleicher, M., Gooch, B.: Automatic image retargeting. In: Proc. International Conference on Mobile and Ubiquitous Multimedia (2005)
4. Chen, L.Q., Xie, X., Fan, X., Ma, W.Y., Zhang, H.J., Zhou, H.Q.: A visual attention model for adapting images on small displays. ACM Multimedia Systems Journal 9, 353–364 (2003)
5. Luis Herranz, J.M.M.: Adapting surneillance video to small displays via object-based cropping. In: Proc. International Workshop on Image Analysis for Multimedia Interactive Services, pp. 72–75 (2007)
6. Liu, H., Xie, X., Ma, W.Y., Zhang, H.J.: Automatic browsing of large pictures on mobile devices. In: Proc. ACM international conference on Multimedia. ACM, New York (2003)
7. Rui, Y., Gupta, A., Grudin, J., He, L.: Automating lecture capture and broadcast: technology and videography. ACM Multimedia Systems Journal 10, 3–15 (2004)
8. Kang, H.W., Matsushita, Y., Tang, X., Chen, X.Q.: Space-time video montage. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 1331–1338 (2006)
9. Gal, R., Sorkine, O., Cohen-Or, D.: Feature-aware texturing. In: Proceedings of Eurographics Symposium on Rendering, pp. 297–303 (2006)
10. He, L., Cohen, M.F., Salesin, D.: The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In: Proc. Annual Conference on Computer Graphics (SIGGRAPH), pp. 217–224. ACM, New York (1996)
11. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Transaction on Graphics, Proc. of SIGGRAPH 2007 26, 10 (2007)

12. Rubinstein, M., Shamir, A., Avidan, S.: Improved seam carving for video retargeting. In: ACM Transaction on Graphics, Proc. of SIGGRAPH 2008 (2008)
13. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
15. Roth, S., Black, M.J.: On the spatial statistics of optical flow. In: Proc. IEEE International Conference on Computer Vision, vol. 1, pp. 42–49 (2005)
16. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 1–8 (2008)
17. Hua, G., Zhang, C., Liu, Z., Zhang, Z., Shan, Y.: Efficient scale-space spatiotemporal saliency tracking for distortion-free video retargeting. Technical Report MSR-TR-2009-87, Microsoft Research (2009)

Visual Saliency Based Object Tracking

Geng Zhang[1], Zejian Yuan[1], Nanning Zheng[1], Xingdong Sheng[1], and Tie Liu[2]

[1] Institution of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
{gzhang,zjyuan,nnzheng,xdsheng}@aiar.xjtu.edu.cn
[2] IBM China Research Lab
liutiel@gmail.com

Abstract. This paper presents a novel method of on-line object tracking with the static and motion saliency features extracted from the video frames locally, regionally and globally. When detecting the salient object, the saliency features are effectively combined in Conditional Random Field (CRF). Then Particle Filter is used when tracking the detected object. Like the attention shifting mechanism of human vision, when the object being tracked disappears, our tracking algorithm can change its target to other object automatically even without re-detection. And different from many other existing tracking methods, our algorithm has little dependence on the surface appearance of the object, so it can detect any category of objects as long as they are salient, and the tracking is robust to the change of global illumination and object shape. Experiments on video clips of various objects show the reliable results of our algorithm.

1 Introduction

Object detection and tracking is an essential technology used in computer vision system to actively sense the environment. As the robotic and unmanned technology develops, automatically detecting and tracking interesting objects in unknown environment with little prior knowledge becomes more and more important.

The main challenge of object tracking is the unpredictable of the environment which always makes it hard to estimate the state of the object. The changing of illumination, clutter background and the occlusion also badly affects the tracking robust. In order to overcome these difficulties, a variety of tracking algorithms have been proposed and implemented. The representative ones include condensation [3], meanshift [4], and probabilistic data association filter [5] and so on.

Generally speaking, most of the tracking algorithm has two major components: the representation model of the object and the algorithm framework. The existing frameworks can be classified into two categories: deterministic methods and stochastic methods. Deterministic methods iteratively search for the optimistic solution of a similarity cost function between the template and the current image. The cost functions widely used are the sum of squared differences (SSD) between the template and the current image [6] and kernel based

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 193–203, 2010.

cost functions [4]. In contrast, the stochastic methods use the state space to model the underlying dynamics of the tracking system and view tracking as a Bayesian inference problem. Among them, the sequential Monte Carlo method, also known as particle filter [7] is the most popular approach.

There are many models to represent the target including image patch [6], color histogram [4] and so on. However, color based models are too sensitive to the illumination changes and always confused with background colors. The contour based features [8] [9] are more robust to illumination variation but they are sensitive to the background clutter and are restricted to simple shape models.

When human sense the environment, they mostly pay attention to the objects which are visually salient. Saliency values the difference between object and background, they are not depend on the objects intrinsic property and is robust to illumination and shape changes. One of the representative visual attention approaches is visual surprising analysis [10] which proves that static and motion features are both important to video attention detection. Itti [11] has proposed a set of static features in his saliency model. More static features are proposed these years [1]. For video series, [2] introduces an method to detect salient object in video series, which combines static and motion features in Dynamic Conditional Random Field (DCRF) under the constraint of global topic model. This approach achieves good results on many challenging videos, but it needs the whole video series to compute the global topic model, which makes it can only be used off-time.

In this paper, we elaborate static and motion saliency features into the framework of particle filter to formulate an online salient object tracking method. When computing the color spatial-distribution feature, we use a graph-based segmentation algorithm [12] as the color clustering method instead of Gaussian Mixture Model (GMM) which is used in [1]. Sparse optical flow [13] is used to get motion field for motion saliency feature computing. All these features are adaptively selected and combined.

The main contributions of our approach are summarized as follows. First, we propose a novel method to tracking salient object online, which is robust to the illumination and shape changes, it can also automatically rebuild attention to the object being tracked disappears. Second, a segmentation based feature is proposed as the global static feature which is more effective than the feature based on GMM.

This paper is organized as follows. We introduce the framework of our algorithm in section 2. The detail of saliency feature computing and combination appears in section 3. In section 3 we also introduce the spatial and temporal coherence used. Section 4 is the collaborative particle filter. Experiment results are shown in section 5.

2 Problem Description and Modeling

Object tracking is an important procedure for human to sense and understand the environment. For human, this procedure can be roughly divided into three

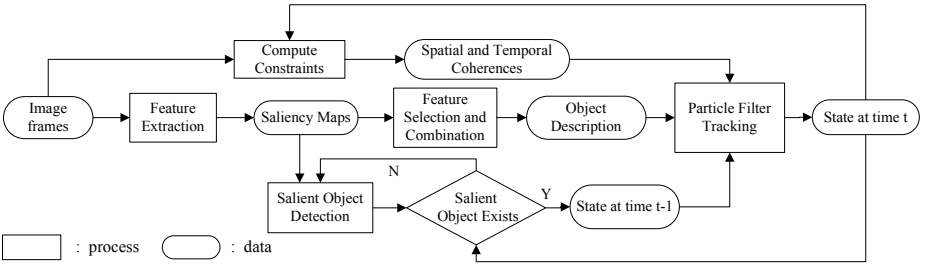


Fig. 1. The flow chart of salient object tracking

parts: attention establishing, attention following and attention shifting. To apply these parts to computer vision, for the input video series $I_1, \cdots, I_t, \cdots$, detection algorithm is used to find the interesting object and build a description model for it. Usually, when there is no high level knowledge, the objects we are interested in are those who are visual salient. After the attention is established, the object's appearance description X_1 is built and the initial object state is gotten in a short time. The state can be the shape, position or scale of the object. Tracking is to estimate the state of the object at time t (X_t) given the initial state X_1 and the observation up to time t ($\mathbf{I}_t = (I_1, \cdots, I_t)$). This process is also called filtering. The flow chart of our method is shown in Fig.1.

The tracking model is usually built under the probabilistic framework of Hidden Markov Model (HMM):

$$\begin{cases} X_t = G(X_{t-1}) + v_{t-1} \\ I_t = H(X_t) + n_t \end{cases}, \tag{1}$$

where $G(\cdot)$ and $H(\cdot)$ are the system transition function and observation function while v_{t-1} and n_t are the system noise and observation noise. When tracking a single object, we formulate the problem as computing the maximum a posteriori (MAP) estimation of X_t. We predict the posterior at time t as

$$P(X_t|I_t) \propto P(I_t|X_t) \int P(X_t|X_{t-1}) P(X_{t-1}|\mathbf{I}_{t-1}) dX_{t-1}, \tag{2}$$

According to (2), we can use the method of statistical filtering to solve the problem. But the state space is extremely huge, computation of the integration in (2) is 'NP hard'. So we choose to use the sequential Monte Carlo method [7].

When filtering, the state X_t can be defined in various forms. Some people try to track the contour of the object [8]. But contour tracking can be easily disturbed by the background clutters and is time-consuming. So people always simplify the state to a rectangle surround the object: $X_t = (x_t, y_t, w_t, h_t) \quad X_t \in \mathbb{R}^4$, where x_t, y_t, w_t, h_t are the position and size of the rectangle at time t.

In our method, the observation at time t is the image frame I_t . The observation model includes the saliency features and the spatial and temporal coherence constraints. So the observation likelihood $P(I_t|X_t)$ can be formulated as

$$P(I_t|X_t) \propto \exp\left\{-\left(\bar{F}_t(X_t, I_t) + Sc_t(X_t, I_t) + Tc_t(X_{t-1,t}, I_{t-1,t})\right)\right\}, \quad (3)$$

where $\bar{F}_t$ is the description of the object which comes from the final saliency feature in our method. Sc_t and Tc_t are the spatial and temporal coherence constraints. The detail of these features and constrains will be described in the following section.

3 The Saliency Features and Constraints

3.1 Static Saliency Features

Visual saliency can be seen as a binary labeling problem which separates the object from the background. Each kind of saliency feature provides a normalized feature map $f(p, I) \in [0, 1]$ which indicates the probability that the pixel p belongs to the salient object. We compute the local and regional features using the method in [1].

Multi-scale Contrast: Contrast is the most common local feature used for attention detection. Without knowing the size of the object, we compute contrast feature $f_{sc}(p, I)$ under a Gaussian image pyramid as

$$f_{sc}(p, I) = \sum_{l=1}^{L} \sum_{p' \in N(p)} \left\| I^l(p) - I^l(p') \right\|_2, \quad (4)$$

where I^l is the image in the l-th level of the pyramid and $N(p)$ is the 8-neighborhood of pixel p.

Center-Surround Histogram: The salient object can always be distinguished by the difference of it and its context. Suppose the object is enclosed by a rectangle. The regional center-surround histogram feature $f_{sh}(p, I)$ is defined as

$$f_{sh}(p, I) \propto \sum_{\left\{p' | p \in R^*(p'), R_S^*(p')\right\}} w_{pp'} \chi^2(R^*(p'), R_S^*(p')), \quad (5)$$

where R^* is the most distinct rectangle centered pixel p' and containing the pixel p. R_S^* is the surrounding contour of R* and have the same area of it. And the weight $w_{pp'} = \exp(-0.5\sigma_{p'}^{-2}\|p - p'\|^2)$ is a Gaussian falloff weight with variance $\sigma_{p'}^2$.

Color Spatial distribution: The salient object usually has distinguishing color with the background. So the wider a color is distributed in the image, the less possible a salient object contains this color. The global spatial distribution of a specific color can be used to describe the saliency of an object. We propose a novel and more effective method to compute this feature.

The first step of computing this feature is color clustering. We use a fast image segmentation algorithm instead of Gaussian Mixture Model to improve

Fig. 2. The left one is the original image. The middle one is the map of segmentation and the right one is the map of the color spatial distribution feature.

the speed and robustness to noise. This algorithm fuses the pixels with similar property, for example color, in a graph-based way [12]. Having the segmentation result, we unify the RGB value in the i-th image segment seg_i to its average color. Then we convert the image to index color representation and compute the distribution variance of every color. So the color spatial distribution feature $f_{sd}(p, I)$ is defined as

$$f_{sd}(p, I) = \left(\sum_{ind(x,y)=ind(p)} xy \left| x - \bar{x} \right| \left| y - \bar{y} \right| \right)^{-1}, \tag{6}$$

where $ind(x, y)$ and $ind(p)$ are the indexing color of point (x, y) and point p. The segmentation result and the feature map are shown in Fig.2.

3.2 Motion Saliency Features

Compared to static object, human's attention is more sensitive to moving objects. The static saliency features can be extended to motion field. In this paper, we use the Lucas/Kanade's motion estimation under a pyramidal implementation [14]. The computed motion field is a 2-D map M with the displacement of every pixel in X and Y directions. In order to compute features from the motion map using the method of computing static featureswe do the lighting operation on M to make the moving area connective. The lighting operation is a Gaussian weighting of the spot areas centered at every sparse points in M. The motion saliency features are computed on the motion field as follows.

Multi-scale Motion Contrast: This local feature f_{Mc} shows the difference of motion. It is computed from the Gaussian pyramid of motion field map:

$$f_{Mc}(p, M) = \sum_{l=1}^{L} \sum_{p' \in N(p)} \left\| M^l(p) - M^l(p') \right\|_2, \tag{7}$$

where M^l is the motion map in the l-th level of the pyramid.

Center-Surround Motion Histogram: This is the regional feature which represents the motion of a block. f_{Mh} is defined as

$$f_{Mh}(p, M) \propto \sum_{\{x'|x \in R^*(p'), R_S^*(p')\}} w_{pp'} \chi^2({R_M}^*(p'), R_{MS}^*(p')), \tag{8}$$

where the weight $w_{pp'}$ has the similar definition as that of the regional static feature.

Motion Spatial Distribution: The global static feature is also extended to the motion field. Motion is first represented using GMM. Then we compute the distribution variance $V_M(m)$ of each component m. So the spatial distribution of motion f_{Md} is defined as

$$f_{Md}(p, M) \propto \sum_m P(m|M_p)(1 - V_M(m)), \tag{9}$$

where $P(m|M_p)$ represents the probability that pixel p belongs to component m.

See details of the motion saliency features in [2].

3.3 Feature Selection and Combination

During the process of tracking, the feature space that best distinguishes between object and background is the best feature space to use for tracking [15]. To achieve best performance using the features mentioned above, we adaptively select and combine them to get the final saliency map F_t.

We notice that when the background of the video is nearly still and the object is moving, the motion features are decisive. In contrast, when the object and background have the similar form of movement, still or moving, we can hardly verify them in the motion field. At this time, static features are more distinctive.

Frame difference is used to decide whether the object has the similar motion form as the background. First, we smooth the adjacent frames I_{t-1}, I_t by Gaussian filtering. Then, the frame difference of margin and the whole image are computed to judge the movement of background and the whole scene. If both the background and the whole scene are moving or still, we use static features, otherwise, motion features are selected.

The final saliency map F_t is defined as a linear combination of the selected features:

$$F_t = \sum_k w_{tk} f_{tk}, \tag{10}$$

where f_{tk} is the k-th selected feature at time t which could be any of the features mentioned above. The weight w_{tk} represents the distinguishing ability of f_{tk}, which is measured by the Saliency Ratio (SR) of the feature:

$$w_{tk} \propto SR_{tk} = \sum_{p \in X_t^*} f_{tk}(p) \Big/ \sum_{p \notin X_t^*} f_{tk}(p), \tag{11}$$

where X_t^* is a simple estimation of X_t by extending the area of X_{t-1}. $p \in X_t^*$ represents that pixel p is in the corresponding rectangle of X_t^*. We normalize SR_{tk} to get w_{tk}.

Given the final saliency map, the object's description $\bar{F}_t(I_t, X_t)$ is defined by the sum square of F_t as

$$\bar{F}_t(X_t, I_t) = \frac{1}{w_t \cdot h_t} \cdot \sum_{p \in X_t} (\mathbb{F}(I_t, p))^2, \tag{12}$$

where $\mathbb{F}(I_t, p) = F_t(p)$, represents the process of feature computing.

3.4 Coherence Constraints

For the tracking problem, the rectangle should fit the object. In our method, that is to say, the border of the rectangle should be close to the edge of the salient object. So we define the spatial coherence as the sum of edge values near the border of the rectangle as

$$Sc_t(X_t, I_t) = \lambda_S \sum_{p \in N(X_t)} \mathbb{E}(I_t, p), \tag{13}$$

where $N(X_t)$ is the area near the corresponding rectangle of X_t , and $\mathbb{E}(I_t, p)$ is the edge value of I_t at pixel p. $\lambda_S = \alpha / w_L h_L$ is the normalizing factor.

Temporal coherence models similarity between two consecutive salient objects. We use the coherence mentioned in [2]:

$$Tc_t(X_{t-1,t}, I_{t-1,t}) = \beta_1 \left\| X_t - X_{t-1} \right\| + \beta_2 \chi^2(h(X_t), h(X_{t-1})), \tag{14}$$

where $\chi^2(h(X_t), h(X_{t-1}))$ is the χ^2 distance between the histogram of two adjacent state. β_1 and β_2 are normalizing factors.

4 Particle Filter Tracking

The particle filter [7] tracker consists of an initialization of the template model and a sequential Monte Carlo implementation of a Bayesian filtering for the stochastic tracking system.

We use the method mentioned in [1] to initialize the system. In order to track moving object, the saliency features we organize in CRF are not only static but can also be motion. The initial state we get from detection is $X_1 = (x_1, y_1, h_1, w_1)$.

In the prediction stage, the samples in the state space are propagated through a dynamic model. In our algorithm, we use a first-order regressive process model:

$$X_t = X_{t-1} + v_{t-1}, \tag{15}$$

where v_{t-1} is a multivariate Gaussian random variable.

Fig. 3. The results of color spatial distribution feature computed with our approach and approach mentioned in [1]. The top row are the original images. The middle row are the results of our approach. The bottom row are the comparing results.

In the update stage, the particles' importance weight is defined by the object description and coherence constraints. The weight of the i-th particle at time t is:

$$w_t^i = \bar{F}_t(X_t^i, F_t) \cdot Sc_t(X_t^i, I_t) \cdot Tc_t(X_{t-1,t}^i, I_{t-1,t}^i), \tag{16}$$

where X_t^i is the corresponding system state of the i-th particle at time t. During update, a direct version of Monte Carlo importance sampling technology [7] is applied to avoid the degeneracy.

5 Experiments

We show here the saliency map of color spatial distribution feature computed with our method and the results of salient object tracking under a variety of situations, including multifold objects tracking, tracking with object appearance changes, and automatically attention rebuilding.

5.1 Color Spatial Distribution

When computing the feature of color spatial distribution. A graph-based segmentation algorithm is used to cluster the adjacent pixels with similar colors. The segmentation is done to every channel of the RGB image and the results are merged to get the final segmentation map. In Fig.3, We compare our feature maps with the maps computed using the approach mentioned in [1]. The lighter area has higher probability to be saliency. As we can see, the results of our approach shows the salient area more clearly than the comparing results. We use images from the Berkeley segmentation dataset [16] for comparing convenience.

5.2 Tracking Results

Our approach is implemented and experiments are performed on video series of various topics. We have collected a video dataset of different object topics, including people, bicycles, cars, animals and so on. Most of our test videos are real-life data collected with a hand-held camera while others are downloaded from the internet.

The objects of interest in our experiments are initiated using the detection approach mentioned in [1]. Different from the original detection process, we manually set them to be 0.3, 0.45 and 0.25 for local, regional and global features. During tracking with sequential Monte Carlo method, we sampled 100 particles for each filter.

Fig.4 shows some tracking results of multifold objects, including bicycle, car, people, and bird. Instead of object's intrinsic property, the saliency based detection and tracking method depends only on the distinction between object and background. So we can track any object as long as it is salient.

Fig.5 shows the tracking results of our approach when the appearance color feature of the object changes. In this experiment, we manually alter the global illumination. Besides, the red car in the video is occluded by tree leaves in some frames which also causes the changing of its appearance feature. We track the red car using our approach and meanshift [4] for comparison. For the meanshift tracker, we manually set the initial position of object. From Fig.5 we can see, our approach gives good results inspite of illumination changes and partial occlusion while meanshift fails when the appearance of the object is changed.

Fig. 4. The results of multifold objects tracking

Fig. 5. The results of tracking under illumination changes and occlusion. The top row are the results of our approach. The bottom row are the results of meanshift.

Fig. 6. The results of tracking while the shape of object changes

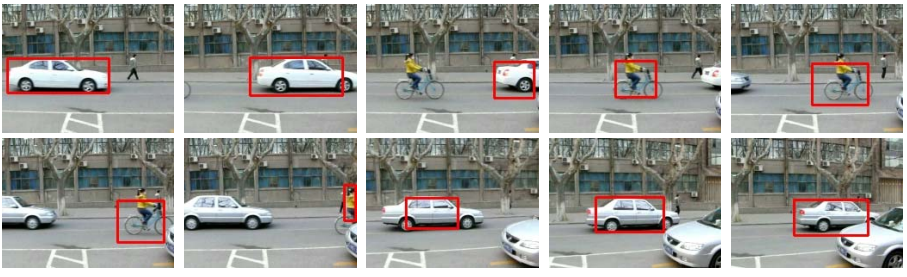


Fig. 7. The results of attention rebuilding

Fig.6 shows the results when the shape of the object changes. In this experiment, the girl comes nearer and near to the camera while making different gestures, which causes obvious changes of the object shape. As we can see, our approach achieve good results under this condition.

In Fig.7 we show that our tracking method can automatically rebuild attention when the object being tracked is out of sight. In this experiment, the detection algorithm set attention on the white car as the initial state. When this car goes out of the scene, attention is rebuilt on the bicycle. Finally, when the bicycle disappears and another car comes, this car becomes the salient object and draws the attention.

6 Conclusion

This paper presents a novel approach of online salient object tracking. In this method, object's appearance is described by its difference to the background which is compute from the static and motion saliency features locally, regionally and globally. A new segmentation based color spatial distribution feature is proposed which is more distinctive between the object and the background. Features are adaptively selected and combined and the sequential Monte Carlo technology is used to track the saliency object. Our approach can track any salient object and is robust to illumination changes and partially occlusions. Moreover, attention can be automatically rebuilt without re-detection in our approach. We are now preparing to extend this approach to multi-object tracking, which involves the modeling of objects interactions.

Acknowledgment

This research is supported in part by the National Basic Research Program of China under Grant No.2007CB311005 and the National Natural Science Foundation of China under Grant No.90820017.

References

1. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to Detect A Salient Object. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007)
2. Liu, T., Zheng, N.N., Ding, W., Yuan, Z.J.: Video Attention: Learning to Detect A Salient Object Sequence. In: 19th International Conference on Pattern Recognition (2008)
3. Isard, M., Blake, A.: Condensation: conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25(5), 564–577 (2003)
5. Rasmussen, C., Hager, G.D.: Probabilistic Data Association Methods for Tracking Complex Visual Objects. IEEE Trans. Pattern Analysis Machine Intell. 23(6), 560–576 (2001)
6. Hager, G.D., Hager, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Trans. Pattern Analysis Machine Intell. 20(10), 1025–1039 (1998)
7. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
8. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Proc. European Conf. on Computer Vision, vol. 1, pp. 343–356 (1996)
9. Leymarie, F., Levine, M.: Tracking deformable objects in the plane using an active contour model. IEEE Trans. Pattern Analysis Machine Intell. 15(6), 617–634 (1993)
10. Carmi, R., Itti, L.: Visual causes versus correlates of attentional selection in dynamic scenes. Vision Research 46(26), 4333–4345 (2006)
11. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Analysis Machine Intell. 20(11), 1254–1259 (1998)
12. Felzenszwalb, P.F., Huttenlocher, D.F.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
13. Smith, S.M., Brady, J.M.: ASSET-2: Real-Time Motion Segmentation and Shape Tracking. IEEE Trans. Pattern Analysis Machine Intell. 17(8), 814–820 (1995)
14. Bouguet, J.Y.: Pyramidal Implementation of the Lucas-Kanade Feature Tracker. Tech. Rep., Intel Corporation, Microprocessor Research Labs (1999)
15. Collins, R.T., Liu, Y.: On-Line Selection of Discriminative Tracking Features. In: Proc. IEEE Conf. on Computer Vision (2003)
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: Proc. IEEE Conf. on Computer Vision (2001)

People Tracking and Segmentation Using Efficient Shape Sequences Matching

Junqiu Wang, Yasushi Yagi, and Yasushi Makihara

The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, Japan
jerywangjq@gmail.com

Abstract. We design an effective shape prior embedded human silhouettes extraction algorithm. Human silhouette extraction is found challenging because of articulated structures, pose variations, and background clutters. Many segmentation algorithms, including the Min-Cut algorithm, meet difficulties in human silhouette extraction. We aim at improving the performance of the Min-Cut algorithm by embedding shape prior knowledge. Unfortunately, seeking shape priors automatically is not trivial especially for human silhouettes. In this work, we present a shape sequence matching method that searches for the best path in spatial-temporal domain. The path contains shape priors of human silhouettes that can improve the segmentation. Matching shape sequences in spatial-temporal domain is advantageous over finding shape priors by matching shape templates with a single likelihood frame because errors can be avoided by searching for the global optimization in the domain. However, the matching in spatial-temporal domain is computationally intensive, which makes many shape matching methods impractical. We propose a novel shape matching approach that has low computational complexity independent of the number of shape templates. In addition, we investigate on how to make use of shape priors in a more adequate way. Embedding shape priors into the Min-Cut algorithm based on distances from shape templates is lacking because Euclidean distances cannot represent shape knowledge in a fully appropriate way. We embed distance and orientation information of shape priors simultaneously into the Min-Cut algorithm. Experimental results demonstrate that our algorithm is efficient and practical. Compared with previous works, our silhouettes extraction system produces better segmentation results.

1 Introduction

Shape matching has been found useful in object recognition. In specific, shape matching based on silhouette information has been proved effective in human gait recognition. Gait recognition overcomes a few problems in an elegant manner that other people identification methods find difficult to handle. For example, a gait recognition system can identify a person from a distance. It is possible to recognize persons using silhouettes extracted from low-resolution images.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 204–213, 2010.

Reliable and accurate silhouettes are crucial for gait recognition. A gait recognition system tends to have poor performance when extracted silhouettes deviate from the real shapes in image sequences. Most gait recognition algorithms assume that silhouette information has been extracted precisely. However, silhouette extraction is a very challenging task especially when image sequences are captured by a moving camera, or the background contains clutters. In fact, silhouette extraction is not only important for gait recognition, but also can be used in human pose analysis and other applications. Human tracking and segmentation are challenging because of articulated structures, pose variations, and background clutters. Although some human tracking algorithm can provide foreground likelihood images [1], it is too difficult to calculate precise human silhouettes based on these likelihood images using simple image morphing techniques. As other segmentation methods, the Min-Cut algorithm also meets difficulties in human silhouette extraction. Markov Random Fields, which are the foundation of the Min-Cut algorithm, seldom present realistic shape priors. Therefore, the Min-Cut algorithm gives poor performance in human silhouette extraction, especially in cluttered backgrounds.

Shape priors play an important role in improving the performance of the Min-Cut algorithm. We develop a silhouette extraction algorithm based on the standard Min-Cut algorithm. Although shape priors have been incorporated in the Min-Cut algorithm in previous works [2], it is not trivial to compute shape priors automatically especially for human silhouettes. The likelihood images given by tracking algorithms are helpful in computing shape priors. Unfortunately, these likelihood images contain many errors. Matching a single likelihood image with a set of silhouettes templates is not reliable due to these errors. Matching shape sequences in spatial-temporal domain is advantageous over finding shape priors by matching shape templates with a single likelihood frame because errors can be avoided by searching for the global optimization in the domain. However, the matching in spatial-temporal domain is computationally intensive, which makes many shape matching methods impractical. We propose a novel shape matching approach that has low computational complexity independent of the number of shape templates.

Incorporating shape prior knowledge alleviates the problems in silhouette extraction. The Min-Cut algorithm allows for a straightforward incorporation of prior knowledge into its formulation. An important problem in employing shape priors is how to apply shape prior knowledge in an appropriate manner. Embedding shape priors into the Min-Cut algorithm based on distances [2] from shape templates is lacking because Euclidean distances cannot represent shape knowledge in a fully appropriate way. We embed distance and orientation information of shape priors simultaneously into the Min-Cut algorithm.

The rest of the paper is arranged as follows. Following the literature review, We describe a novel shape matching method and its application in optimal path searching in Section 3. We incorporate shape priors into the Min-Cut algorithm in Section 4. Both distance and orientation information of shape priors are

embedded within the Min-Cut algorithm. Experimental results for image sequences are presented in Section 5. Section 6 concludes this work.

2 Previous Work

Human silhouette extraction is found challenging because of articulated structures, pose variations, and background clutters. Segmentation methods based solely on low-level information often provide poor performance in these difficult scenarios. Many segmentation algorithms meet difficulties in human silhouette extraction. The Min-Cut algorithm [3], which has achieved great success in interactive segmentation, faces problem in silhouette extraction.

The evident power of shape priors as an additional cue has been noticed by many researchers. Freedman and Zhang [2] define the coherence part of the Min-Cut algorithm by considering the shape distance transform results. In their work, shape priors are given manually, which is tedious for segmentation in video sequence. It is desirable to computer shape priors automatically for human silhouette extraction. Wang *et al.* [1] proposed a shape prior seeking algorithm by searching for the best path in spatio-temporal domain. The major drawback of their work is the heavy computational costs in shape matching, which makes their algorithm not practical for real applications. Although there is an effort in accelerating the shape matching process [4], the performance is still not efficient especially when the number of shape templates is large.

While pedestrian model representations have been employed for refining silhouettes in previous works [5,6], they all assume that an foreground likelihood images can be obtained by background subtraction. In addition, these works do not address the shape matching problem, which is crucial for the applicability of silhouette extraction.

In visual tracking literature, temporal coherency was employed in particle filtering. Rathi *et al.* [7] formulated a particle filtering algorithm in a geometric active contour framework in which temporal coherency and curve topology are handled. In addition, shape and appearance information were considered in a unified metric framework by Toyama and Blake [8]. The use of exemplars alleviates the difficulty of constructing complex motion and shape models. Although these algorithms do improve the performance of tracking, few of them deal with silhouette extraction.

3 Computing Shape Priors

We adopt an adaptive mean-shift tracking approach [9,10]. The adaptive tracker provides bounding boxes and Foreground Likelihood Images (FLI). We match FLI sequence with silhouette templates in the standard gait models. A Standard Gait Model (SGM) is constructed for the matching. Tanimoto distance [11] is taken as the similarity measure between FLIs and silhouette templates. Matching shape sequences in spatial-temporal domain [1] is advantageous over finding shape priors by matching shape templates with a single likelihood frame because

errors can be avoided by searching for the global optimization in the domain. However, the matching in spatial-temporal domain is computationally intensive, which makes many shape matching methods impractical. We will introduce an efficient shape matching approach that has low computational complexity independent of the number of shape templates.

3.1 Matching Measure

FLIs generated by the tracker should be normalized to have the same size as the silhouette templates. An FLI in the nth frame is denoted by $\boldsymbol{f}(n)$. The center and the height of a human region's bounding box are denoted by (c_x, c_y) and h, respectively. Registration and scaling based on the bounding box are processed in the same way as the SGM, thus producing the normalized FLI $\boldsymbol{f}_N(n; c_x, c_y, h)$ in the nth frame.

Tanimoto distance [11] is exploited as the measure between the FLI $\boldsymbol{f}_N$ and SGM $\boldsymbol{g}$:

$$D_T(\boldsymbol{f}_N, \boldsymbol{g}) = 1 - \frac{\sum_{(x,y)} \min\{f_N(x,y), g(x,y)\}}{\sum_{(x,y)} \max\{f_N(x,y), g(x,y)\}}, \tag{1}$$

where $f_N(x,y)$ and $g(x,y)$ are the likelihood and silhouette values, respectively, at (x,y) . The Tanimoto distance between an FLI and SGM is 1 if they have identical shapes, and 0 if there is no overlap between them.

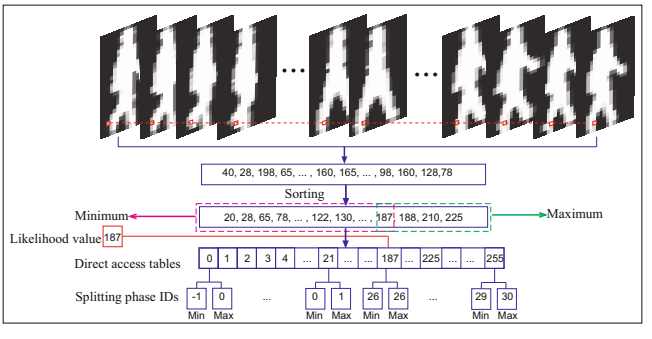


Fig. 1. The efficient Tanimoto distance method

3.2 $O(1)$ Tanimoto Distance Computation

The matching between foreground likelihood images and silhouette templates using Tanimoto distance is computationally expensive because every pixel in every image has to be calculated individually. It takes more than 120 seconds when 30 silhouette templates are employed in the sequence matching. (The algorithm is run on a 1.6GHz laptop). This expensive Tanimoto distance computation makes the proposed approach impractical. The problem is exacerbated if we wish to include more shape variations by adding silhouette template images.

We present a novel distance calculation method whose matching complexity is $O(1)$. The calculation of the minimum and maximum in Eq. 1 is expensive if they are calculated individually. We noticed, however, that the silhouette template and likelihood images can be quantized in a limited range ($[0, 255]$ in this work) without any negative effect on the matching results. Because the silhouette template images given in the initialization have been aligned, we sort the values at each position in the template images. The phase information of these templates is kept in the sorted results. Based on the sorted results, a direct access table is built for each position of the silhouette template images. Each table has two splitting phase IDs corresponding to the calculation of the maximum and minimum in Eq. 1.

As shown in Figure 1, to calculate the minimum and maximum in Eq. 1, we do not need to compare the input likelihood value with all the values in the silhouette template images. To make the illustration clear, we use 30 phases here. In the initialization, we sort the values at each position in the template images. Then we build one direct access table for each value in the range [0, 255]. Each table contains two splitting phase IDs: the maximum phase ID p^S_{max} and the minimum phase ID p^S_{min}. For a given input likelihood value, the direct access table is found directly. Then the maximums and minimums can be assigned based on the splitting phase IDs corresponding to the input likelihood value. The value in a phase in the sorted results is assigned as the minimum, when its phase ID is equal to or smaller than the minimum splitting phase ID p^S_{min}, or as the maximum, when its phase ID is equal to or greater than the maximum splitting phase ID. For instance, if the input likelihood value is 0, all the values in the template images are assigned as maximums and the input likelihood value is always assigned as the minimum. The computational complexity of the matching process is $O(1)$. In other words, the matching is independent to the number of silhouette template images. This method is particularly important when many templates are necessary to cover large variations in shape. The template values are sorted only once during the initialization.

Tanimoto distance measures the overlapping regions of two input images. Its computation time is further reduced by reusing the calculated overlapping regions [12]. Tanimoto distance can be formulated as $D_T(\boldsymbol{f}_N, \boldsymbol{g}) = \frac{G+F-C}{C}$, where $F = \sum_{(x,y)} f_N(x, y)$, $G = \sum_{(x,y)} g(x, y)$, and $C(f_N, g) = \sum_{(x,y)} \min\{f_N(x, y), g(x, y)\}$. Based on this formulation, G (sum of gait template values) is calculated only once during the initialization. For each input foreground likelihood image sequence, F is also calculated only once. $C(f_N, g)$ is calculated based on the efficient method.

Using the method described above, it takes around 0.6 seconds to calculate distances between an input foreground likelihood sequence and all templates (including shifting and scaling) on the 1.6GHz laptop. Further computational cost reduction is expected when the number of shape templates becomes larger. In addition, the proposed distance calculation method can be used in other applications where silhouette template matching is necessary [12].

4 Embedding Shape Priors in Min-Cut Segmentation

Let $\mathcal{L} = \{1 \dots K\}$ be a set of labels. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Segmentation is formulated in terms of energy minimization in the Min-Cut algorithm [13]. We embed shape priors into the algorithm.

4.1 Embedding Shape Priors

Shape priors can be embedded in the Min-Cut algorithm by adding an energy term [2] [14]:

$$E(A) = E_{smoothness}(A) + E_{data}(A) + E_{shape}(A). \quad (2)$$

The Min-Cut algorithm with shape priors includes shape fitness, smoothness and initial labeling. The energy function E_{shape} is penalized if the segmented contour deviates from the boundary of the silhouette. Shape priors are represented by a distance transform result.

We found that the method in [2] is deficient: the embedded shape priors need to be very accurate, otherwise the distance transform can misguide the segmentation. In contrast, we introduce orientation information in the shape priors to encourage smoothness of the segmentation. It has been found that the statistics of steered filters for human limbs are different from those of other natural scenes [15]. In this work, we learn a vocabulary that includes position and gradient orientations in human silhouettes. We calculate gradient orientations and normalized positions (in $[0, 1]$) in 400 segmented people silhouette images and detect edges in the silhouette templates using Canny edge detector. Then we calculate gradient orientations and normalized coordinates on the edges. We apply K-Means to form an initial vocabulary. An EM algorithm is adopted to get the final vocabulary. We compute the mean and covariance matrix for each word. The vocabulary has 10 words finally, allowing edges belonging to a same word formulate as an oriented template. Thus 10 oriented templates are gotten for every template image. Then Euclidean distance transform is applied to these oriented templates.

To improve the first term in Eq. 2, based on the distance transform results of the oriented templates, we calculate the minimum distance in corresponding to a pixel in the input image. If the minimum distance is greater than the threshold d_{min}^{DT}, the pixel is set as background. This method is effective in dealing with inaccurate shape priors. In contrast, the shape priors used in [2,1,14] have to be very accurate, otherwise they can misguide the segmentation.

We also found that probabilities decrease too quickly near a contour. We decrease the distance values obtained from the distance transform by applying a local search. We then extract edges in the input images. The distance is kept as is if there are edges near the shape prior. Otherwise the distance values are multiplied by a constant factor c_{edge}. (The factor is set to 0.8 in this work). The cost function of shape priors is well described in the transformed image. The shape prior energy is written as $E_{shape} = \sum_{(pq)\in\mathcal{N}:A_p \neq A_q} \frac{\psi_{min}(p)+\psi_{m}in(q)}{2}$, where ψ_{min} is the minimum distance on the transformed image.

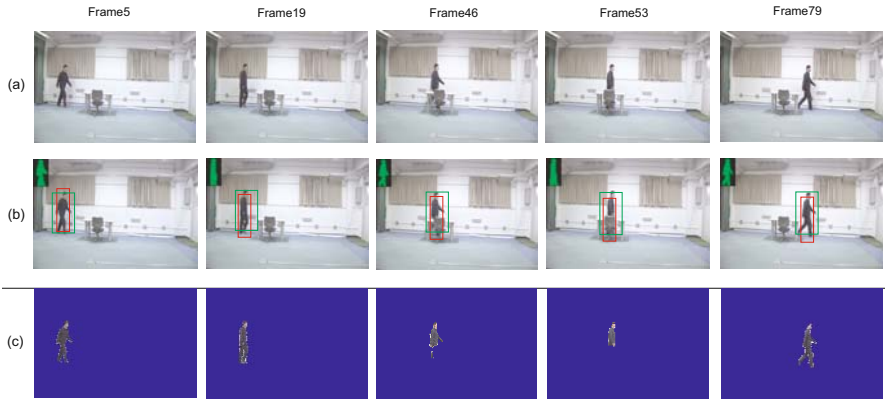


Fig. 2. Tracking and segmentation results for the indoor sequence. (a) Input images. (b) Initial bounding boxes (in red) generated by the tracker, optimal bounding boxes (in green) and gait models (phase) obtained using optimal path searching. (c) Segmentation results by embedding the shape priors in the Min-Cut algorithm.

5 Experimental Results

We tested the proposed algorithm on 8 sequences with tracking and segmentation ground truths. The size of all images in these sequences is 360 × 240 pixels. We show the results for two sequences in detail. Performance is evaluated with respect to refinement of bounding boxes and phase estimation and the improvement in segmentation. The Quantitative evaluation of other sequences is given in 5.3.

5.1 Refinement of Bounding Boxes and Phase Estimation

The results for the two sequences are shown in Figures 2 and 3, respectively. The initial bounding boxes produced by the tracker are not well aligned with the people regions and the initial foreground likelihoods are low for some parts.

Based on the optimal path searching results, the tracking bounding boxes are shifted to better positions. The bounding boxes are not accurately aligned with the person. The vertical centers in the initial bounding boxes deviate from their correct positions. The positions are adjusted downwards based on the optimal path searching results. The horizontal centers of the initial bounding boxes are relatively more accurate. They need to be shifted less frequently than the vertical centers.

The selected silhouette templates provided by the searching results are shown in Figure 2(b). The gait phases corresponding to the walking person are correct. The shape priors are incorporated in the Min-Cut algorithm giving the segmentation results shown in Figure 2(c).

Next we evaluate the smoothness of the walking phase transfer in Figure 4. The phases estimated with and without using shape sequence matching are

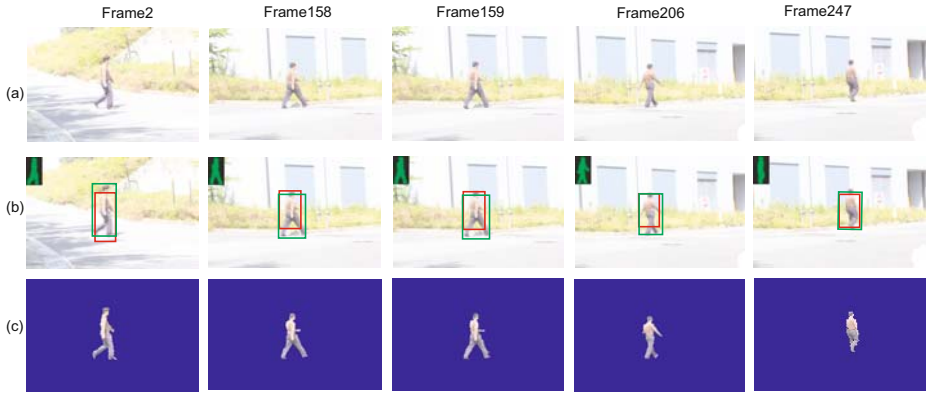


Fig. 3. Tracking and segmentation results for the outdoor sequence. (a) Input images. (b) Initial bounding boxes (in red) generated by the tracker, optimal bounding boxes (in green) and gait models (phase) calculated by the optimal path searching using shape sequence matching. (c) Segmentation results by embedding the shape priors in the Min-Cut algorithm.

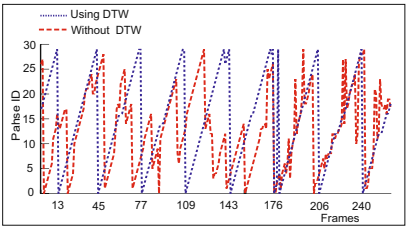


Fig. 4. Phase transition estimation for the sequence in Figure 3

compared. The estimation results without using shape sequence mathcing are obtained by matching an input likelihood image with all the silhouette templates. The phases estimated using shape sequence matching are much more accurate than those without shape sequence matching. This demonstrates the importance of shape sequence matching in optimal path searching. The phase estimation also verifies the necessity of searching in a spatiotemporal space instead of in a single frame. When the view changes, the phase estimation result is not as accurate as the side view. However, it is still much better than the estimation obtained from single image matching.

5.2 Segmentation Results

The segmentation ground truths of these sequences are obtained by labeling the images manually. Each pixel is labeled as background, foreground, or ambiguous. The ambiguous label is used to mark mixed pixels along the boundaries between

Table 1. Segmentation errors for eight of the test sequences

Test seqeunces	3	4	5
No Prior	0.18	0.30	0.32
Using Prior	0.092	0.24	0.23
Test seqeunces	6	7	8
No Prior	0.23	0.36	0.4
Using Prior	0.15	0.22	0.31

foreground and background. We measure the error rate as a percentage of mis-segmented pixels, ignoring ambiguous pixels.

5.3 Quantitative Evaluation

Segmentation results using shape priors are shown for the indoor sequence (Figure 2) and the outdoor sequence (Figure 3). A quantitative evaluation of the segmentation results with and without shape priors is shown in Figure. The segmentation results with shape priors embedded are compared with those without shape priors. The incorporation of shape priors improves the performance of the segmentation. Compared with the indoor sequence, the use of shape priors is more helpful in the outdoor sequence. Thus shape priors play a more important role in the challenging outdoor sequence.

Table 1 shows segmentation errors with respect to ground truth for six of our test sequences. Among them, sequence 3, 4, and 5 are indoor sequences, and 6, 7, 8 are outdoor sequences. The segmentation using shape priors has lower error rate in these sequences.

6 Conclusions

We find optimal paths for an input likelihood sequence by matching it with silhouette templates. The novel efficient shape matching method makes the proposed approach practical for real applications. The shape sequence matching provides shape priors for silhouette extraction. The proposed prior embedding method is effective. The segmentation performance is also improved based on shape constraints.

References

1. Wang, J., Makihara, Y., Yagi, Y.: Human tracking and segmentation supported by silhouette-based gait recognition. In: Proc. of IEEE Int. Conf. on Robotics and Automation (2008)
2. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: Proc. of CVPR, pp. 755–762 (2004)
3. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proc. of ICCV, pp. 105–112 (2001)

4. Wang, J., Makihara, Y., Yagi, Y.: People tracking and segmentation using spatiotemporal shape constraints. In: Proc. of 1st ACM International Workshop on Vision Networks for Behavior Analysis, in conjunction with ACM Multimedia (2008)
5. Baumberg, A., Hogg, D.: Learning flexible models from image sequences. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 800, pp. 299–308. Springer, Heidelberg (1994)
6. Lee, L., Dalley, G., Tieu, K.: Learning pedestrian models for silhouette refinement. In: Proc. of ICCV, pp. 663–670 (2003)
7. Rathi, Y., Vaswani, N., Tannenbaum, A., Yezzi, A.: Particle filtering for geometric active contours with application to tracking moving and deforming objects. In: Proc. of CVPR, pp. 2–9 (2005)
8. Toyama, K., Blake, A.: Probabilistic tracking in a metric space. In: Proc. of ICCV, pp. 50–57 (2001)
9. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25(5), 564–577 (2003)
10. Wang, J., Yagi, Y.: Integrating color and shape-texture features for adaptive real-time tracking. IEEE Trans. on Image Processing 17(2), 235–240 (2008)
11. Sloan Jr., K.R., Tanimoto, S.L.: Progressive refinement of raster images. IEEE Trans. on Computers 28(11), 871–874 (1979)
12. Marszalek, M., Schmid, C.: Accurate object localization with shape masks. In: Proc. of CVPR, pp. 1–8 (2007)
13. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1124–1137 (2004)
14. Bray, M., Kohli, P., Torr, P.H.S.: Posecut: simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 642–655. Springer, Heidelberg (2006)
15. Sidenbladh, H., Black, M.J.: Learning the statistics of people in images and video. Int'l Journal of Computer Vision 54(3), 183–209 (2003)

Monocular Template-Based Tracking of Inextensible Deformable Surfaces under L_2-Norm*

Shuhan Shen, Wenhuan Shi, and Yuncai Liu

Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University, Shanghai 200240, China

Abstract. We present a method for recovering the 3D shape of an inextensible deformable surface from a monocular image sequence. State-of-the-art method on this problem [1] utilizes L_∞-norm of reprojection residual vectors and formulate the tracking problem as a Second Order Cone Programming (SOCP) problem. Instead of using L_∞ which is sensitive to outliers, we use L_2-norm of reprojection errors. Generally, using L_2 leads a non-convex optimization problem which is difficult to minimize. Instead of solving the non-convex problem directly, we design an iterative L_2-norm approximation process to approximate the non-convex objective function, in which only a linear system needs to be solved at each iteration. Furthermore, we introduce a shape regularization term into this iterative process in order to keep the inextensibility of the recovered mesh. Compared with previous methods, ours performs more robust to outliers and large inter-frame motions with high computational efficiency. The robustness and accuracy of our approach are evaluated quantitatively on synthetic data and qualitatively on real data.

1 Introduction

3D shape recovery of objects from 2D monocular image sequences has been studied for decades. Great successes have been achieved for rigid and articulated-rigid objects. However, most objects in the real world vary their shapes over time, such as faces, papers, clothes etc. The problem of reconstructing the shape of such deformable objects has much interest recently.

Structure-from-motion based methods are widely used for the non-rigid shape and motion recovery. Bregler et al. [2] are the first to use a factorization-based method for the recovery of non-rigid structures, in which the 3D shape in each frame is formulated as a linear combination of a set of basis shapes. Torresani et al. [3] model the time-varying shape as a rigid transformation combined with a non-rigid deformation. This model is a form of Probabilistic Principal Components Analysis (PPCA) shape model whose parameters can be learned in the reconstruction process. In the case of perspective cameras, Xiao et al. [4] present a closed-form solution for perspective reconstruction given the assumption that there exists a set of independent deformable basis shapes.

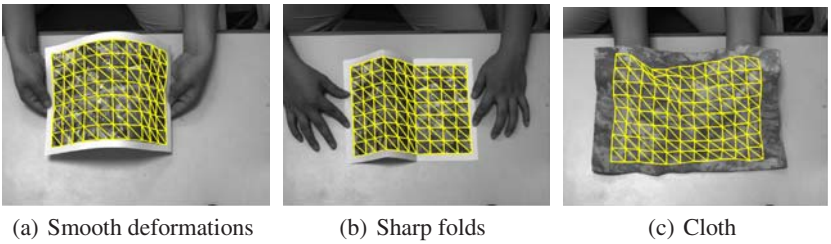
* This work was supported by National Natural Science Foundation of China under Grant 60833009, National 973 Key Basic Research Program of China under Grant 2006CB303103, and Graduate Innovation Foundation of SJTU.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 214–223, 2010.

Bartoli et al. [5] propose a low-rank structure-from-motion method which handles missing data, automatically selects the number of deformation modes and makes use of several different priors. Structure-from-motion based methods always make very strong assumptions about the deformations which makes it not suitable for objects undergoing large deformations. Besides, this kind of method needs the whole image sequence to compute the solution and thus is not suited for reconstruction on the fly.

Modeling the surface as a 3D triangulated mesh is a popular way to represent the behavior of general deformations. Gay-Bellile et al. [6] present an 2D intensity-based non-rigid registration method with self-occlusion reasoning. This method constrains the 2D warp to shrink in self-occluded regions while detecting them based on this property and successfully deals with extreme self-occlusions. However, this method is only suited for 2D registration and hard to be generalized to 3D cases. Salzmann et al. [1] represent surfaces as triangulated meshes and disallow large changes of edge orientation between consecutive frames, and formulate the tracking problem as a Second Order Cone Programming (SOCP) feasibility problem. This method yields a convex formulation with a unique minimum and enables us to handle highly-deformable surfaces without adding unwarranted smoothness constraints. Based on [1], Zhu et al. [7] reformulate the problem into an unconstrained quadratic optimization problem which can be solved more efficiently and robustly than SOCP. However, these methods introduces strong constraints to bound the vertex displacements from on frame to the next, which makes it fail to deal with large inter-frame motions. Besides, these methods rely on the L_∞-norm of reprojection residual vectors, which are sensitive to outliers.

In this paper we present a method for tracking inextensible deformable surfaces in 3D under L_2-norm. Generally, using L_2 leads a non-convex optimization problem which is difficult to minimize. Instead of solving the non-convex problem directly, we design an iterative L_2-norm approximation process to approximate the non-convex objective function, in which only a linear system needs to be solved at each iteration. Furthermore, we introduce a shape regularization term into this iterative process in order to keep the inextensibility of the recovered mesh. The proposed method does not involve any smoothness constraint which makes it applicable for surfaces with various kinds of deformations such as those in Fig. 1. Compared with state-of-the-art approach under L_∞-norm [1], our method performs more robust to outliers and large inter-frame motions with high computational efficiency.



(a) Smooth deformations (b) Sharp folds (c) Cloth

Fig. 1. Reconstructing the structure of a deformable surface from monocular image sequences using our approach. In all the graphs, we overlay the recovered mesh on the original image.

The rest of the paper is organized as follows. The proposed deformable surface 3D tracking method under L_2-norm is detailed in section 2. Experimental results on both synthetic and real data are reported in section 3. Finally, conclusions are presented in section 4.

2 Deformable Surface 3D Tracking under L_2-Norm

2.1 Problem Statement

The deformable surface is represented as a n_v-vertex triangulated mesh. We denote the 3D coordinate of each vertex of the mesh by $\mathbf{v}_i = [x_i, y_i, z_i]^T$, and the 3D structure of the mesh can be parameterized as a long vector $\mathbf{V}$ of dimension $3n_v$ by concatenating the three coordinates of all $\mathbf{v}_i$, as $\mathbf{V} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_{n_v}^T]^T$.

The shape vector $\mathbf{V}$ is the variable we want to estimate. Our method relies on 3D-to-2D correspondences between the surface and the image. Let $\mathbf{x}_i$ be a 3D surface point which can be expressed in terms of its barycentric coordinate of the facet where $\mathbf{x}_i$ lies on, as:

$$\begin{aligned} \mathbf{x}_i &= a_i\mathbf{v}_p + b_i\mathbf{v}_q + c_i\mathbf{v}_r \\ &= \mathbf{T}_i\mathbf{V}, \end{aligned} \tag{1}$$

where $\mathbf{v}_p$, $\mathbf{v}_q$ and $\mathbf{v}_r$ are the vertices of the facet that $\mathbf{x}_i$ lies on, a_i, b_i and c_i are the barycentric coordinate of $\mathbf{x}_i$, and $\mathbf{T}_i$ is a transformation matrix dependent on the barycentric coordinate.

Assuming $\mathbf{x}_i$ is in the camera referential, given the internal parameters matrix $\mathbf{K}$, the projection of $\mathbf{x}_i$ is:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{T}_i\mathbf{V}$$

The reprojection residual vector for $\mathbf{x}_i$ with respect to image measurement $(\hat{u}_i, \hat{v}_i)^T$ is given by:

$$\delta_i = \left(\frac{\mathbf{K}_1\mathbf{T}_i\mathbf{V}}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}} - \hat{u}_i, \frac{\mathbf{K}_2\mathbf{T}_i\mathbf{V}}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}} - \hat{v}_i \right)^T \tag{2}$$

where $\mathbf{K}_1$, $\mathbf{K}_2$ and $\mathbf{K}_3$ are the first, second and third rows of $\mathbf{K}$ respectively.

Generally, using different norms of δ_i leads different optimization problems. In this paper, we use (L_p, L_q) to represent a combination of two vector norms, where the first norm L_p corresponds to the image norm used and the second one L_q corresponds to the norm of residual vector. In this notion the (L_2, L_∞)-case have been solved using SOCP [1] or quadratic optimization [7]. However, the problem with residual vector measured under L_2 norm, i.e. the (L_2, L_2)-case, remains challenging since minimizing the sum-of-squares objective function is known to be a troublesome non-convex problem with multiple local minima.

Recently, a few approaches have been successfully used to obtain globally optimal solutions for some geometric vision problems such as triangulation and camera

resectioning under (L_2, L_2)-norm [8] [9]. However, these methods are usually computationally expensive and not applicable for high-dimensional variables such as the triangulated mesh $\mathbf{V}$.

2.2 L_2-Norm Approximation

In the (L_2, L_2)-case, the cost function for 3D-to-2D correspondences can be formulated as:

$$\begin{aligned} f(\mathbf{V}) &= \sum_{i=1}^{n} \|\delta_i\|^2 \\ &= \sum_{i=1}^{n} \left\| \begin{array}{c} \dfrac{(\mathbf{K}_1\mathbf{T}_i - \hat{u}_i\mathbf{K}_3\mathbf{T}_i)\mathbf{V}}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}} \\ \dfrac{(\mathbf{K}_2\mathbf{T}_i - \hat{v}_i\mathbf{K}_3\mathbf{T}_i)\mathbf{V}}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}} \end{array} \right\|^2 \end{aligned} \tag{3}$$

where $\|\cdot\|$ represents the L_2 norm, n is the number of 3D-to-2D correspondences. Minimizing (3) directly is difficult. However, in the tracking context, the tracking result for previous frame could be used as an reasonable initial guess of current frame, which gives us an efficient iterative method for minimizing $f(\mathbf{V})$.

At time instance t, we use the tracking result for previous frame $\mathbf{V}^{t-1}$ as an initial guess $\mathbf{V}_0$, and seek for a better result $\mathbf{V}_1$. Point $\mathbf{V}_1$ can be expressed as $\mathbf{V}_1 = \mathbf{V}_0 + \delta_0$. Consider a scenario where we are in the k-th iteration and try to update $\mathbf{V}_k$ to $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$. In this case, the denominators in (3), $\mathbf{K}_3\mathbf{T}_i\mathbf{V}$, could be approximated by $\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k$, which gives an approximation function of $f(\mathbf{V})$, namely $g(\delta_k)$, as:

$$\begin{aligned} g(\delta_k) &= \sum_{i=1}^{n} \left\| \begin{array}{c} \dfrac{(\mathbf{K}_1\mathbf{T}_i - \hat{u}_i\mathbf{K}_3\mathbf{T}_i)(\mathbf{V}_k + \delta_k)}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \\ \dfrac{(\mathbf{K}_2\mathbf{T}_i - \hat{v}_i\mathbf{K}_3\mathbf{T}_i)(\mathbf{V}_k + \delta_k)}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \end{array} \right\|^2 \\ &= \sum_{i=1}^{n} \|\mathbf{h}_i(\mathbf{V}_k + \delta_k)\|^2 \\ &= (\mathbf{V}_k + \delta_k)^T\mathbf{H}(\mathbf{V}_k + \delta_k) \end{aligned} \tag{4}$$

where

$$\mathbf{H} = \sum_{i=1}^{n} \mathbf{h}_i^T\mathbf{h}_i, \quad \mathbf{h}_i = \left[\begin{array}{c} \dfrac{\mathbf{K}_1\mathbf{T}_i - \hat{u}_i\mathbf{K}_3\mathbf{T}_i}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \\ \dfrac{\mathbf{K}_2\mathbf{T}_i - \hat{v}_i\mathbf{K}_3\mathbf{T}_i}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \end{array} \right] \tag{5}$$

Obviously, letting the derivative of the objective function $g(\delta_k)$ to be zero and solving the corresponding linear equation can generate the solution that minimize $g(\delta_k)$. Thus δ_k could be achieved by solving the following linear equation:

$$\bigtriangledown g(\delta_k) = 2\mathbf{H}\delta_k + 2\mathbf{H}\mathbf{V}_k = 0 \tag{6}$$

Or equivalently:

$$\mathbf{H}\delta_k = -\mathbf{H}\mathbf{V}_k \tag{7}$$

This L_2-norm approximation process can be summarized as: we i) use the tracking result for previous frame $\mathbf{V}^{t-1}$ as an initial guess $\mathbf{V}_0$ for current frame, and set $k = 0$; ii) evaluate $\mathbf{H}$ using Eq. (5), and calculate δ_k by solving (7); iii) set $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$, then set $k = k+1$ and go to step ii). This iteration continues until $\delta_k \to 0$ which means $g(\delta_k)$ approaches $f(\mathbf{V}_k)$.

We should note that $\mathbf{H}$ is a symmetric matrix of size $3n_v \times 3n_v$, and n_v of the eigenvalues of $\mathbf{H}$ are very close to zero no matter how many correspondences are used, which is similar with the result observed in [10]. This indicates that ambiguities always arise using only 3D-to-2D correspondences under the monocular perspective projection, and other constraints should be introduced to regularize the mesh shape.

2.3 Shape Regularization

Since the surface is inextensible, the most general shape constraints should be designed to retain the original length of each mesh edge, as:

$$\|\mathbf{v}_p - \mathbf{v}_q\| = l_r, \tag{8}$$

where l_r is the original length of the edge linking vertices $\mathbf{v}_p$ and $\mathbf{v}_q$. Since $\mathbf{v}_p - \mathbf{v}_q$ is a linear transformation of $\mathbf{V}$, we denote:

$$\mathbf{v}_p - \mathbf{v}_q = \mathbf{E}_r\mathbf{V}$$

where $\mathbf{E}_r$ is a transformation matrix. Then the constraints in Eq. (8) can be expressed as:

$$\|\mathbf{E}_r\mathbf{V}\| = l_r, \quad r = 1, \ldots, m, \tag{9}$$

where m is the number of mesh edges.

Suppose we are in the k-th iteration and try to update point $\mathbf{V}_k$ to point $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$. The constraints in (9) in this case become:

$$\|\mathbf{E}_r(\mathbf{V}_k + \delta_k)\| = l_r, \quad r = 1, \ldots, m,$$

which can be expressed as:

$$2\mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k + \delta_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k = l_r^2 - \mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\mathbf{V}_k, \quad r = 1, \ldots, m \tag{10}$$

Now if we remove the second term on the left-hand side of Eq. (10), we have:

$$2\mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\delta_k \approx l_r^2 - \mathbf{V}_k^T\mathbf{E}_r^T\mathbf{E}_r\mathbf{V}_k, \quad r = 1, \ldots, m \tag{11}$$

Note that Eq. (11) is a *local* linear approximation of the quadratic equations in Eq. (10), and (10) and (11) are asymptotically equivalent to each other as $\delta_k \to 0$.

The m linear equality constraints in (11) can be put together as:

$$\mathbf{F}_k\delta_k = \mathbf{g}_k, \tag{12}$$

where

$$\mathbf{F}_k = \begin{bmatrix} 2\mathbf{V}_k^T\mathbf{E}_1^T\mathbf{E}_1 \\ \vdots \\ 2\mathbf{V}_k^T\mathbf{E}_m^T\mathbf{E}_m \end{bmatrix}, \quad \mathbf{g}_k = \begin{bmatrix} l_1^2 - \mathbf{V}_k^T\mathbf{E}_1^T\mathbf{E}_1\mathbf{V}_k \\ \vdots \\ l_m^2 - \mathbf{V}_k^T\mathbf{E}_m^T\mathbf{E}_m\mathbf{V}_k \end{bmatrix} \tag{13}$$

Since (12) is a set of m linear equations with $3n_v$ unknowns ($3n_v > m$), there are infinitely many solutions for this equation. It is well known that all these solutions can be linearly parameterized using singular value decomposition (SVD) of the coefficient matrix $\mathbf{F}_k$. Since the m linear equations in (12) are independent, the rank of matrix $\mathbf{F}_k$ is m and its SVD is given by $\mathbf{F}_k = \mathbf{U}\Sigma\mathbf{S}$. All the solutions of $\mathbf{F}_k\delta_k = \mathbf{g}_k$ are then given by:

$$\delta_k = \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k, \tag{14}$$

where $\mathbf{F}_k^+$ denotes the Moore-Penrose pseudo inverse of $\mathbf{F}_k$, which can also be computed using the SVD of $\mathbf{F}_k$, $\mathbf{S}_k$ is a matrix of size $3n_v \times (3n_v - m)$ which consists of the last $(3n_v - m)$ columns of matrix $\mathbf{S}$, and φ_k is an arbitrary vector of dimension $3n_v - m$.

By substituting Eq. (14) into (4), we have a new objective function $t(\varphi_k)$, as:

$$\begin{aligned} t(\varphi_k) &= g(\mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k) \\ &= \sum_{i=1}^{n} \left\| \begin{array}{c} \dfrac{(\mathbf{K}_1\mathbf{T}_i - \hat{u}_i\mathbf{K}_3\mathbf{T}_i)(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \\ \dfrac{(\mathbf{K}_2\mathbf{T}_i - \hat{v}_i\mathbf{K}_3\mathbf{T}_i)(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)}{\mathbf{K}_3\mathbf{T}_i\mathbf{V}_k} \end{array} \right\|^2 \\ &= \sum_{i=1}^{n} \|\mathbf{h}_i(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)\|^2 \\ &= (\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k)^T\mathbf{H}(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k + \mathbf{S}_k\varphi_k) \end{aligned} \tag{15}$$

where $\mathbf{H}$ is defined in Eq. (5).

Letting the derivative of the objective function $t(\varphi_k)$ to be zero and solving the corresponding linear equation can generate the solution that minimize $t(\varphi_k)$. That is, φ_k is calculated by solving the following linear equation:

$$\bigtriangledown t(\varphi_k) = 2\mathbf{S}_k^T\mathbf{H}\mathbf{S}_k\varphi_k + 2\mathbf{S}_k^T\mathbf{H}(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k) = 0 \tag{16}$$

Or equivalently:

$$\mathbf{S}_k^T\mathbf{H}\mathbf{S}_k\varphi_k = -\mathbf{S}_k^T\mathbf{H}(\mathbf{V}_k + \mathbf{F}_k^+\mathbf{g}_k) \tag{17}$$

Now, the deformable surface tracking process can be summarized as: we i) use the tracking result for previous frame $\mathbf{V}^{t-1}$ as an initial guess $\mathbf{V}_0$ for current frame, and set $k = 0$; ii) evaluate $\mathbf{H}$ using Eq. (5), evaluate $\mathbf{F}_k$ and $\mathbf{g}_k$ using Eq. (13); iii) compute the SVD of $\mathbf{F}_k$ and then $\mathbf{F}_k^+$ and $\mathbf{S}_k$; iv) calculate φ_k by solving Eq. (17), and then obtain δ_k using (14); v) set $\mathbf{V}_{k+1} = \mathbf{V}_k + \delta_k$, then set $k = k + 1$ and go to step ii). This iteration continues until $\delta_k \to 0$. In practice, we stop the iteration when $\max_i |\delta_{k,i}| < \varepsilon$, where $\delta_{k,i}$ is the i-th element of δ_k, and ε is a prescribed threshold.

We should note that the sequence of $\mathbf{V}_k$ may not converge if large outliers exist. In this case, we remove the 3D-to-2D correspondence with the maximum reprojection error every 10 iterations until $\mathbf{V}_k$ converges.

3 Experimental Results

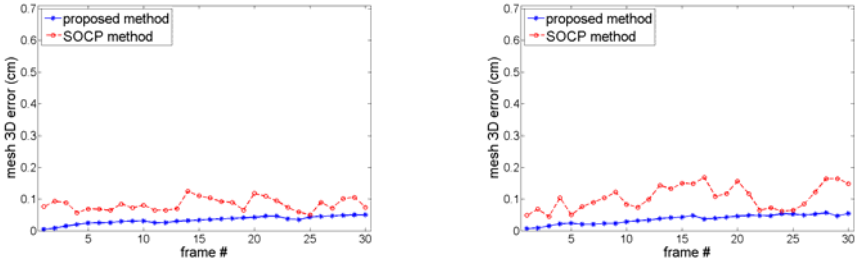
The performance of the proposed method was evaluated with both synthetic and real data. All the experiments are implemented under the Matlab environment on a 3GHz standard PC. In all the experiments, the threshold ε is set to 2 according to the grid search result.

3.1 Synthetic Data

We conducted three experiments to evaluate the performance of the proposed method on the synthetic data with gaussian noise, large inter-frame motions and outliers respectively. For comparison, Salzmann et al.'s SOCP method [1], which is considered to be a state-of-the-art approach, was also used for the synthetic data.

Experiment I. The first experiment evaluated the robustness of our method to image noise. We apply forces to a $8\text{cm}\times 11\text{cm}$ triangulated mesh and keep mesh edges to be their original lengths, which generates a 30-frame synthetic sequence. We randomly choose four 3D points in each facet of the mesh and project these points on an image plane using a perspective projection matrix, which gives us a set of 3D-to-2D point correspondences at each frame. Then we add gaussian noise with mean zero and standard deviations $\sigma = 1$ and $\sigma = 2$ to all the image point locations at each frame. Fig. 2 shows the average 3D distance between reconstructed mesh vertices and the ground-truth of each frame using two different methods.

The results show that our approach gives more stable and accurate results than the SOCP method. Furthermore, our method takes about 2.8 seconds to process one frame compared with 5.7 seconds of the SOCP method.



(a) $\sigma = 1$ (b) $\sigma = 2$

Fig. 2. Average 3D distance between reconstructed mesh vertices and the ground-truth using two different methods on the synthetic data with gaussian noise. (a) is the result when adding gaussian noise with mean zero and $\sigma = 1$. (b) is the result when adding gaussian noise with mean zero and $\sigma = 2$.

Experiment II. The second experiment evaluated the robustness of our method to large inter-frame motions. The synthetic sequence in Experiment I is used here. We remove 10 frames (from frame 6 to frame 15) from this sequence, which represents a large inter-frame motion. Fig. 3 shows the experimental results. Note that we didn't draw the results between frame 6 and frame 15 since they had been removed from the synthetic data in this experiment.

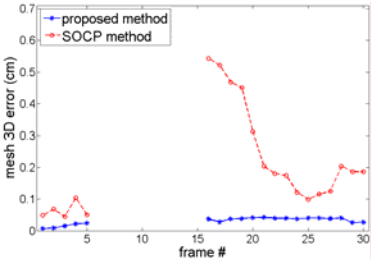


Fig. 3. Average 3D distance between reconstructed mesh vertices and the ground-truth using two different methods on the synthetic data with large inter-frame motions

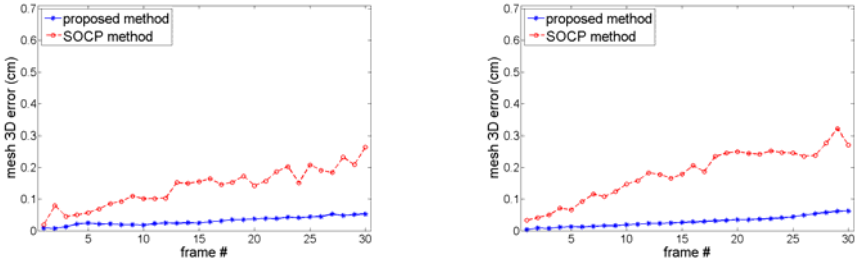
The results show that the SOCP method is unable to track the surface correctly when large inter-frame motions exist, and the reason is that this method introduces strong constraints to bound the vertex displacements from on frame to the next. Compared with the SOCP method, our approach correctly recovered the structure even though the surface changed dramatically between two frames.

Experiment III. The third experiment evaluated the robustness of our method to outliers. The synthetic sequence in Experiment I is used here. Besides the $\sigma = 2$ gaussian noise, at each frame, 30% and 60% of the image locations are corrupted by gaussian noise with mean zero and $\sigma = 10$, which can be regarded as outliers. Fig. 4 shows the tracking results using two different methods.

The results show that the proposed method performs more robust and stable than the SOCP method. Furthermore, our method takes about 6.6 seconds to process one frame, as opposed to 16.5 seconds of the SOCP method. The SOCP method utilizes L_∞-norm of reprojection errors and needs to remove all outliers, no matter large or small, before getting the final result. In contrast, our method utilizes the L_2-norm which is more roust than L_∞-norm in dealing with outliers, that is, only large outliers will be removed. As a result, the SOCP method spent more time on the outlier removal process compared with the proposed method.

3.2 Real Data

Our approach is qualitatively evaluated on real images. We use a paper sheet and a piece of cloth for real data experiments. We capture image sequences using a calibrated camera. The keypoints on the mesh and their barycentric coordinates are extracted from a reference image in which the surface is in front of the camera without deformations.



(a) $\sigma = 10$, 30% (b) $\sigma = 10$, 60%

Fig. 4. Average 3D distance between reconstructed mesh vertices and the ground-truth using two different methods on the synthetic data with outliers. (a) is the result when adding $\sigma = 10$ gaussian noise to 30% of the the image point locations besides the $\sigma = 2$ noise. (b) is the result when adding $\sigma = 10$ gaussian noise to 60% of the the image point locations besides the $\sigma = 2$ noise.

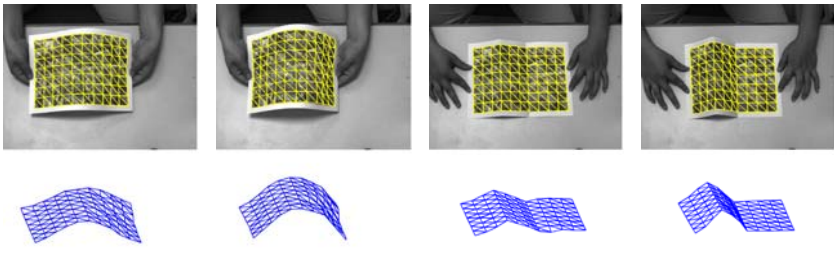


Fig. 5. Some tracking results of a paper sheet with smooth deformations and sharp folds. The first row are the original images with reprojected mesh. The second row are the reconstructed meshes seen from a different view.

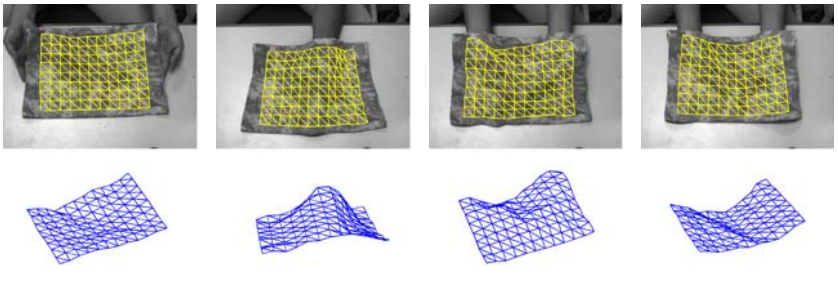


Fig. 6. Some tracking results of a piece of cloth

Then the 3D-to-2D keypoint correspondences are established between the reference image and the input one using SIFT [11]. Because SIFT is a wide-baseline matching method, mismatching happens frequently in the real data.

Some tracking results of the real image sequence are shown in Fig. 5 and Fig. 6. The results show that our approach can correctly recover 3D structures of surfaces with smooth, sharp and other complex deformations.

4 Conclusion

In this paper we present a method for the 3D shape recovery of a deformable surface from monocular image sequences. Different from state-of-the-art methods which utilize L_∞-norm of reprojection errors, our approach uses L_2-norm which performs more robust to outliers than the L_∞. Although using L_2-norm leads a non-convex optimization problem which is difficult to minimize, we design an iterative L_2-norm approximation process to approximate the non-convex objective function, in which only a linear system needs to be solved at each iteration. Furthermore, we introduce a shape regularization term into this iterative process in order to keep the inextensibility of the recovered mesh. Compared with previous methods, the proposed approach performs more robust to outliers and large inter-frame motions with high computational efficiency.

References

1. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. In: Proceedings of IEEE International Conference on Computer Vision (2007)
2. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2000)
3. Torresani, L., Hertzmann, A., Bregler, C.: Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 878–892 (2008)
4. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: Proceedings of IEEE International Conference on Computer Vision (2005)
5. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
6. Gay-Bellile, V., Bartoli, A., Sayd, P.: Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. In: Proceedings of IEEE International Conference on Computer Vision (2007)
7. Zhu, J., Hoi, S.C., Xu, Z., Lyu, M.R.: An effective approach to 3d deformable surface tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 766–779. Springer, Heidelberg (2008)
8. Kahl, F., Henrion, D.: Globally optimal estimates for geometric reconstruction problems. International Journal of Computer Vision 74(1), 3–15 (2007)
9. Kahl, F., Agarwal, S., Chandraker, M.K., Kriegman, D., Belongie, S.: Practical global optimization for multiview geometry. International Journal of Computer Vision 79(3), 271–284 (2008)
10. Salzmann, M., Lepetit, V., Fua, P.: Deformable surface tracking ambiguities. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

A Graph-Based Feature Combination Approach to Object Tracking

Quang Anh Nguyen[1,2], Antonio Robles-Kelly[1,2], and Jun Zhou[1,2]

[1] RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia
[2] National ICT Australia (NICTA)*, Locked Bag 8001, Canberra ACT 2601, Australia
{Quang.Nguyen,Antonio.Robles-Kelly,Jun.Zhou}@nicta.com.au

Abstract. In this paper, we present a feature combination approach to object tracking based upon graph embedding techniques. The method presented here abstracts the low complexity features used for purposes of tracking to a relational structure and employs graph-spectral methods to combine them. This gives rise to a feature combination scheme which minimises the mutual cross-correlation between features and is devoid of free parameters. It also allows an analytical solution making use of matrix factorisation techniques. The new target location is recovered making use of a weighted combination of target-centre shifts corresponding to each of the features under study, where the feature weights arise from a cost function governed by the embedding process. This treatment permits the update of the feature weights in an on-line fashion in a straightforward manner. We illustrate the performance of our method in real-world image sequences and compare our results to a number of alternatives.

1 Introduction

Object tracking is a classical problem in computer vision and pattern recognition. Existing approaches often employ low complexity local image descriptors and features to construct a model that can then be used to track the object. These features can be based upon the RGB values of the image under study, local texture descriptors and contrast operators [1]. The responses of the image brightness to Harr-like [2], Gaussian and Laplacian filters [3] have also been used for recognition and tracking.

Along these lines, modern appearance-based tracking frameworks such as the kernel-based methods [4], Kalman filter [5] and particle filter trackers [6] have attracted a great deal of attention from the computer vision community. The well known kernel-based algorithm [4] makes use of the mean-shift optimisation scheme [7] to search for a local maximum of feature similarity on the image lattice, without prior knowledge of the tracking environment. The Kalman filter [5] and the particle filter trackers [6] improve the tracking robustness by introducing probabilistic models for object and camera motion as well as state-space hypotheses.

However, it is somewhat surprising that the methods above do not combine multiple cues, but rather employ a fixed set of colour feature spaces such as RGB [4] or HSV [6].

* 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.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 224–235, 2010.

Hence they are prone to error in practical settings where the illumination conditions and object appearance vary significantly between subsequent frames. Stern and Efros [8] improve the tracking performance by adaptively swapping the tracking features across five pre-determined colour-space combinations. Nguyen and Smeulders [9] use a set of Gabor filters to transform the image intensities into texture information. Collins et al. [10] deploy the mean-shift tracker [4] on a feature pool of 49 log-likelihood images comprised by unique combinations of R,G and B values. In a related development, Han and Davis [11] combined two different colour spaces so as to construct 14 log likelihood images. Feature extraction is then achieved by performing PCA on the foreground and the local image background. Machine learning techniques such as Adaboost have also been employed to enhance multiple-feature trackers [12,13].

In this paper, we aim at presenting a feature combination approach to object tracking. Here, we make use of graphical model setting so as to abstract the features used in the tracking process into a graph. This leads to the use of techniques commonly employed in graph-spectral methods [14] to achieve maximum separation between the target and the scene background . Thus, here we provide a link between graphical models, graph embedding methods and tracking feature correlation. This treatment is devoid of free parameters and windowed sampling, while permitting low complexity features to be linearly combined analytically.

Moreover, the use of graph embedding techniques also leads to the recovery of a set of weights so as to evaluate the contribution of each feature to the target shift. This is reminiscent of boosting techniques [15], where a weak leaner is used for classification. In this way, our method can be viewed as a weighted linear combination of "weak" mean-shifts in each feature space which are combined into a "strong" global one. We also present an on-line updating scheme for the weights governing the tracking task. In practice, this is done based on the level of "confidence" on the target position and leads to the updating of the target model. Further, our approach can employ any arbitrary number of low complexity local image features and is not limited to colour cues.

The paper is organised as follows. Firstly, we introduce the basic concepts that will be used throughout the paper. We then turn our attention to the recovery of a global mean-shift from the contributions of each feature space. The on-line weight updating scheme is presented in Section 4. Finally, we elaborate on the algorithm in Section 5 and, in Section 6, we illustrate the robustness of the algorithm on a number of video sequences and compare our results to those delivered by alternatives.

2 Kernel-Based Tracking in Arbitrary Feature Spaces

As mentioned earlier, Kernel-based object tracking [4] makes use of the spatially-weighted histogram of the target region as input to a similarity function which the tracker aims at maximising via mean-shift iterations [16].

In order to characterise a target, one or more feature spaces must be determined so that a non-parametric power density function (PDF) such as M-bin histogram can be estimated. The ideal choice of feature space is the one that is distinctive to the target with respect to the surrounding background while being robust to noise and image

corruption. The principle of the kernel-based tracker is, however, not restricted to any particular feature space and, in a multiple feature setting, can be summarised as follows.

Let $\mathbf{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_{|\mathbf{\Phi}|}\}$ be the set of feature-spaces used for purposes of tracking. For the feature space ϕ_i, the new target position η_{ϕ_i} can be recovered making use of the two M-bin histograms $\mathbf{Q}_{\phi_i}$ and $\mathbf{P}_{\phi_i}$ corresponding to the target model and the search window, respectively. In particular,

$$\eta_{\phi_i} = \frac{\sum_{n=1}^{N} x_n w_n}{\sum_{i=1}^{N} w_n} \tag{1}$$

where w_n is the similarity weight for the n^{th} pixel x_n in the search window. For further detail on the equation above, we direct the reader to [4].

With the $|\mathbf{\Phi}|$ "weak" shifts $\{\eta_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ at hand, we can compute the "global" shift η as the weighted average of these "weak" shifts as $\eta = \sum_{i=1}^{|\mathbf{\Phi}|} \gamma_{\phi_i} \eta_{\phi_i}$ where γ_{ϕ_i} is the feature weight for the updated target-centre η_{ϕ_i} corresponding to the feature space ϕ_i.

3 Feature Combination via Graph Embedding

We now turn our attention to the recovery of the feature weight γ_{ϕ_i}. To this end, we cast the problem of feature combination into a graph-theoretic setting. In this manner, we aim at embedding the set of pairwise correlations between features in a metric space. To do this, we abstract the pairwise relationships between low complexity features into a relational structure and make use of graph-spectral methods, i.e. the eigenvalues and eigenvectors of the Laplacian matrix [17], so as to cast the feature weight γ_{ϕ_i} in an optimisation setting that leads to a Rayleigh Quotient. This can be viewed as the recovering of a graph embedding such that the correlation between features is minimum.

This embedding process commences by viewing the PDFs for the target foreground and its surrounding background as nodes on a weighted graph, whose edge-weights are given by their correlation in its geometric sense, i.e. the inner product of the pairwise PDFs. Viewed in this way, the Laplacian of the graph can be related to a Gram matrix of scalar products. This treatment, in turn, allows the use of matrix factorisation techniques to recover the coordinates for the embedding of the graph. Thus, the problem of finding the feature weight γ_{ϕ_i} turns into that of recovering the set of variables that maximises the pairwise distances between the features under consideration and, therefore, minimises their cross-correlation via the use of the eigenvalues and eigenvectors of a purposely-constructed matrix.

3.1 Feature Mapping

To commence, we require some formalism. Let $G = (V, E, W)$ denote a weighted graph with index-set V, edge-set $E = \{(u, v)|(u, v) \in V \times V\}$ and edge-weights $W : E \rightarrow [0, 1]$. Recall that, as mentioned earlier, the nodes of the graph are the PDFs for the target model and the scene background, i.e.$\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{P}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ respectively. As a result, we let the weight $W(u, v)$ associated with the edge connecting

the pair of nodes u and v corresponding to the i^{th} and j^{th} features in $\mathbf{\Phi}$ be given by the normalised cross-correlation

$$W(u,v) = \begin{cases} \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\mathbf{P}_{\phi_i}}{\|\mathbf{P}_{\phi_i}\|} \right\rangle & \text{if } i = j \\ \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\mathbf{Q}_{\phi_j}}{\|\mathbf{Q}_{\phi_j}\|} \right\rangle & \text{otherwise} \end{cases} \tag{2}$$

Note that W is a symmetric matrix of scalar products, in which the diagonal elements are given by the cross-correlation between the PDFs for the foreground and the background of the same feature, while the off-diagonal elements are the cross-correlation between the PDFs for the foreground for different features.

To take our analysis further, we proceed to define the squared distance between features on the graph. Here, we set the pairwise squared distance between a pair of nodes as their correlation value. This is akin to the approaches in pairwise grouping such that in [18]. We define

$$W(u,v) = \|\varphi(u) - \varphi(v)\|^2 \tag{3}$$

where $\varphi(u)$ is the embedding vector, i.e. the vector of coordinates for the feature ϕ_i corresponding to the node u in V. The squared distance can also be expressed in terms of a set of inner products as follows

$$W(u,v) = \langle \varphi(u), \varphi(u) \rangle + \langle \varphi(v), \varphi(v) \rangle - 2 \langle \varphi(u), \varphi(v) \rangle \tag{4}$$

This permits viewing the correlation between tracking features as pairwise distances in a metric space making use of the inner products.

3.2 Double Centering

To provide a link between the edge-weights $W(u,v)$ and the coordinate vectors $\varphi(u)$, we make use of double-centering [19]. In particular, this can be achieved by firstly relating the edge-weight matrix W to the Laplacian matrix $\mathcal{L}$ [14]. With the Laplacian matrix at hand, a double-centered matrix of scalar products $\mathbf{H} = \mathbf{J}\mathbf{J}^T$ can be computed. This operation introduces a linear dependency over the columns of the matrix $\mathbf{H}$ while preserving the symmetry of W.

This treatment is important because it allows us to view the double centered matrix $\mathbf{H}$ as a matrix of scalar products which can then be interpreted as the sums of squared, pairwise distances $\|\varphi(u) - \varphi(v)\|^2$ introduced in Equation 3. Furthermore, it can be shown that the matrix $\mathbf{H}$ is, in fact, the double-centered graph Laplacian [19]. As a result, the element of the matrix $\mathbf{H}$ corresponding to the nodes $u, v \in V$ is given by

$$\mathbf{H}(u,v) = -\frac{1}{2}\left[\mathcal{L}(u,v)^2 - \frac{1}{|V|}\sum_{u\in V}\mathcal{L}(u,v)^2 - \frac{1}{|V|}\sum_{v\in V}\mathcal{L}(u,v)^2 + \frac{1}{|V|^2}\sum_{u,v\in V}\mathcal{L}(u,v)^2\right] \tag{5}$$

The graph Laplacian $\mathcal{L}$ is defined as $\mathcal{L} = \mathbf{D}^{-1/2}(\mathbf{D} - W)\mathbf{D}^{-1/2}$ where $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D}$=$diag(deg(1), deg(2), \ldots, deg(|V|))$ and $deg(u)$=$\sum_{v\in V} W(u,v)$ is the degree of the node $u \in V$.

Let ξ_l be the l^{th} eigenvector of $\mathbf{H}$ scaled so its sum of squares is equal to the corresponding eigenvalue λ_l. Since $\mathbf{H}\xi_l = \lambda_l\xi_l$ and $(\mathbf{J}\mathbf{J}^T)\xi_l = \mathbf{H}\xi_l$, it follows that the squared distance between a pair of nodes in Equation 3 can be now written as

$$\| \varphi(u) - \varphi(v) \|^2 = \sum_{l=1}^{|V|} \lambda_l(\xi_l(u) - \xi_l(v))^2 = \mathbf{H}(u,u) + \mathbf{H}(v,v) - 2\mathbf{H}(u,v) \quad (6)$$

3.3 Minimising Feature Correlation

With these ingredients, we can introduce the variables $\pi(u)$ such that the weighted correlations between low complexity features are minimum. We do this by making use of the quantity

$$\epsilon = \sum_{u,v\in V} \left\|\pi(u)\varphi(u) - \pi(v)\varphi(v)\right\|^2 \quad (7)$$

which we aim at minimising. The cost function above can also be interpreted as the sum of squared weighted cross-correlations between the PDFs used for purposes of tracking. Thus, we can use Equation 6 and, after some algebra, we write

$$\epsilon = \sum_{u,v\in V} \left(\pi(u)^2\mathbf{H}(u,u) + \pi(v)^2\mathbf{H}(v,v) - 2\pi(u)\pi(v)\mathbf{H}(u,v)\right) \quad (8)$$

Note that, Equation 8 can be divided into two sets of terms. The first of these corresponds to the diagonal matrix of $\mathbf{H}$. The other set accounts for the off-diagonal elements of H. Rearranging terms, we get

$$\epsilon = 2|V| \sum_{u\in V} \pi(u)^2\mathbf{H}(u,u) - \sum_{\substack{u,v\in V \\ u=v}} 2\pi(u)^2\mathbf{H}(u,u) - \sum_{\substack{u,v\in V \\ u\neq v}} 2\pi(u)\pi(v)\mathbf{H}(u,v) \quad (9)$$

where we use the following facts

$$\sum_{u,v\in V} \pi(u)^2\mathbf{H}(u,u) = |V| \sum_{u\in V} \pi(u)^2\mathbf{H}(u,u) \text{ and } \sum_{u,v\in V} \pi(u)^2\mathbf{H}(u,u) = \sum_{u,v\in V} \pi(v)^2\mathbf{H}(v,v)$$

Moreover, Equation 9 can be reduced to

$$\epsilon = -\sum_{\substack{u,v\in V \\ u\neq v}} 2\pi(u)\pi(v)\mathbf{H}(u,v) \quad (10)$$

which can be written in compact form by defining a matrix $\hat{\mathbf{H}}$ which comprises the off-diagonal elements of $\mathbf{H}$ as follows

$$\hat{\mathbf{H}}(u,v) = \begin{cases} \mathbf{H}(u,v) & \text{if } u \neq v \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

This yields $\epsilon = -2\mathbf{\Pi}^T\hat{\mathbf{H}}\mathbf{\Pi}$ where $\mathbf{\Pi} = [\pi(1), \pi(2), \cdots, \pi(|V|)]^T$ is a column vector of order $|V|$. Note that the expression above is the numerator of a Rayleigh Quotient

whereas the omitted denominator, $\mathbf{\Pi}^T\mathbf{\Pi}$, is a normalisation constant. Thus, minimising ϵ is equivalent to maximising $\mathbf{\Pi}^T\hat{\mathbf{H}}\mathbf{\Pi}$ and, therefore, $\mathbf{\Pi}^* = \underset{\mathbf{\Pi}}{\operatorname{argmin}}\{\epsilon\}$ is given by the leading eigenvector of $\hat{\mathbf{H}}$ which corresponds to the eigenvalue whose rank is the largest.

The vector $\mathbf{\Pi}^*$, hence, is the minimiser of the squared distances between the nodes in the graph, i.e. the correlation between features. As a result, the set of feature weights $\{\gamma_{\phi_i}\}_{\phi i\in\mathbf{\Phi}}$ corresponding to the "weak" shifts is given by

$$\gamma_{\phi_i} = \frac{\pi(u)}{\sum_{u\in V}\pi(u)} \tag{12}$$

where the i^{th} feature ϕ_i corresponds to the node u in V.

4 On-Line Feature Weight Updating

As kernel-based trackers [4,10,11] rely on the M-bin histograms of the model to determine the target location via the mean-shift optimisation scheme, the validity of these histograms is extremely important for robust tracking. In [10], these M-bin histograms are modified after every frame by randomly selecting pixels from the target foreground so as to modify the tracking models across the feature spaces. Despite effective, this "mixing" method does not discriminate between pixels and, hence, is susceptible to mislocalisation due to histogram bias.

Here, we present an on-line feature weight updating method based upon the cross-correlation between the histograms of the current target model and that corresponding to the recovered target-centre after each mean-shift application. This technique is based upon the weighted cross correlation between histograms and, thus, is devoid of pixel-sample selection and injection. Moreover, we calculate the total cross-correlation in a similar manner to that in Section 3.

To commence, let $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i\in\mathbf{\Phi}}$ be the set of the M-bin histograms obtained from the new target position and ϱ_{ϕ_i} be the cross-correlation between the two histograms $\mathbf{Q}_{\phi_i}$ and $\hat{\mathbf{Q}}_{\phi_i}$, i.e. $\varrho_{\phi_i} = \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\hat{\mathbf{Q}}_{\phi_i}}{\|\hat{\mathbf{Q}}_{\phi_i}\|} \right\rangle$. The total cross-correlation between the two sets of histograms, $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i\in\mathbf{\Phi}}$ for the new target position and $\{\mathbf{Q}_{\phi_i}\}_{\phi_i\in\mathbf{\Phi}}$ for the current model, can be computed as a linear combination of the weighted feature cross-correlation ϱ_{ϕ_i} as

$$\mathit{\Gamma} = \sum_{\phi_i\in\mathbf{\Phi}} \gamma_{\phi_i}\varrho_{\phi_i} \tag{13}$$

where $\{\gamma_{\phi_i}\}_{\phi_i\in\mathbf{\Phi}}$ is the set of feature weights derived from Section 3. This treatment, in turn, allows us to set decision bounds for the updating operation. We do this by updating the model M-bin histograms only when the condition $0 \leq \kappa_0 < \mathit{\Gamma} < \kappa_1 \leq 1$ is satisfied, where κ_0 and κ_1 are constants. This hinges in the confidence of the tracking operation by following the notion that, if the total correlation between the new target-centre histograms and that of the target model is close to unity, there is no need to update since the two sets are sufficiently "close". On the contrary, if the total correlation is too low, then updating would "corrupt" the model. Updating is hence, appropriate when the

Algorithm 1. Training

Data: Selected region of the target model.
begin
Sample N_1 pixels from the foreground, N_2 pixels from the background.
Compute the set of M-bin histograms $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{P}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ across the feature spaces $\mathbf{\Phi}$.
Compute W as in Equation 2
Compute $\mathbf{H}$ as in Equation 5 and $\hat{\mathbf{H}}$ as in Equation 11
Compute $\mathbf{\Pi}^*$ as the leading eigenvector ξ_1 of $\hat{\mathbf{H}}$
Compute the feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using $\mathbf{\Pi}^*$ as in Equation 12
Save $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$
end

correlation is not so low so as to introduce noise corruption but not as high as to be a computational burden without improving tracking accuracy.

When update operations are deemed necessary, the histogram set $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ for the target model is updated making use of a mixture model of the form

$$\mathbf{Q}'_{\phi_i} = P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i})\hat{\mathbf{Q}}_{\phi_i} + \left(1 - P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i})\right)\mathbf{Q}_{\phi_i} \tag{14}$$

This can be viewed as a "blending" operation between the two histograms. It is, indeed, a two-class expectation for the two PDFs $\hat{\mathbf{Q}}_{\phi_i}$ and $\mathbf{Q}_{\phi_i}$, whose prior is given by the probability of the new target position given the feature cross-correlations ϱ_{ϕ_i}.

5 Algorithm Description

With the developments presented in the previous sections, the tracking algorithm can be divided into two stages. The first stage is the training phase, in which the user is required to select the target to track. The samples inside the selected region are then used to compute a set of PDFs corresponding to the feature spaces under study. In a similar manner, the area around the target is also sampled to create a set of background PDFs. In our implementation, for the sake of efficiency, we perform background sampling in the area of twice the size of the target. With the two sets of foreground and background PDFs at hand, we compute the corresponding cross-correlation weight matrix W. Subsequently, the double-centering matrix $\mathbf{H}$ is determined, followed by its off-diagonal matrix $\hat{\mathbf{H}}$. The set of feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ is then recovered from the leading eigenvector of $\hat{\mathbf{H}}$.

In the second stage, the tracking vehicle is the mean-shift tracker presented in [4]. After each new target position, the total cross-correlation Γ is then calculated to determine if the set of model histograms needs to be updated, i.e. $\kappa_0 < \Gamma < \kappa_1$. This implies that the feature weights and the target-model feature histograms will only be updated if the tracking operation is reliable, i.e. with a $\Gamma > \kappa_0$, while keeping computational cost low by avoiding updating operations when the candidate and the model are virtually the same, i.e. with a $\Gamma < \kappa_1$.

Algorithm 2. Tracking

Data: $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$, $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and target centre y
begin
  for idx=*StartFrame to EndFrame* **do**
    while *true* **do**
      Compute the set of M-bin histograms $\{\mathbf{P}_{\phi_\mathbf{i}}\}_{\phi_i \in \mathbf{\Phi}}$ for the searching window
      Compute the target centre $\{\eta_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ of each mean-shift as in Equation 1
      Compute the new target centre $\eta = \sum_{\phi_i \in \mathbf{\Phi}} \gamma_{\phi_i} \eta_{\phi_i}$
      if $\|\eta - y\| \leq \varepsilon$ **then**
        $idx = idx + 1$
        break
      else
        Update the target centre $y = \eta$
      end
    end
    Compute $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ at the new target centre
    Compute $\varrho_{\phi_i} = \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\hat{\mathbf{Q}}_{\phi_i}}{\|\hat{\mathbf{Q}}_{\phi_i}\|} \right\rangle$
    Compute $\Gamma = \sum_{\phi_i \in \mathbf{\Phi}} \gamma_{\phi_i} \varrho_{\phi_i}$
    if $\kappa_0 < \Gamma < \kappa_1$ **then**
      Compute $P\left(\hat{\mathbf{Q}}_{\phi_i} | \varrho_{\phi_i}\right)$
      Update $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ to $\{\mathbf{Q}'_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using Equation 14
      Compute the new feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using the updated $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ as in Algorithm 1
    end
  end
end

6 Experiments

In this section, we illustrate the robustness of our algorithm by presenting results on two image sequences from the PETS-ECCV 2004 dataset[1]. Note that further sequences can be found in the supplemental material accompanying this paper. In the first sequence, the target moves from a bright area in the scene to a shady region, meets another person and then walks away. The second sequence shows a group of four people moving across the scene with some body-overlapping as they approach the camera. For each of these sequences, the tracking target is manually selected by the user at the initial frame.

We have compared our results to those yielded by two competing algorithms. These are the on-line Variance Ratio-based (VR-based) method proposed by Collins et al. [10] and the on-line PCA-based method by Han and Davis [11]. Note that these methods [10,11] have significant improvement in performance over the random weights. We have also implemented two sets of features. The first set consists of 49 linear combinations of R,G,B as described in [10]. We call this set the 49-feature set. The second set is a mix of gradient, contrast and texture features including brightness, normalised RGB, Local

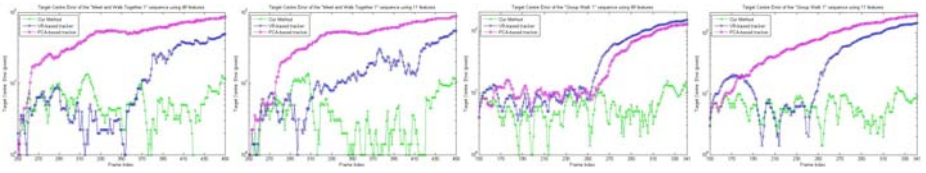
[1] PETS dataset can be accessed from http://www.cvg.rdg.ac.uk/slides/pets.html

Fig. 1. Results for the "Meet and Walk Together 1" sequence at frames 250 and 380. From top-to-bottom: results yielded by our algorithm using the 49-feature set (first and second columns) and the 11-feature set (third and forth columns), the on-line VR-based tracker [10] using the 49-feature set and the 11-feature set, and the on-line PCA-based tracker [11] using the 49-feature set and the 11-feature set.

Binary Patterns (LBPs) and six Haar-like features [2], which we call the 11-feature set. The Harr-like features include vertical and horizontal 2,3 and 4-rectangle features.

In our implementation of the VR-based tracker, we select the set of five log-likelihood images that yield the highest Variance Ratio as the tracking features for the current frame. For the PCA-based tracker, the eigenvectors associated with the eigenvalues whose normalised sum is greater than 0.7 are used so as to reduce the dimensionality of the log-likelihood images. As suggested in [11], a Gaussian filter is also implemented in order to reduce the amount of unwanted noise in the likelihood image corresponding to the leading eigenvalue. For our tracker, we consider the conditional probability for the update operations to be normally distributed, i.e. $P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i}) \sim N(\mu, \sigma)$. Moreover, we set $\mu = (\kappa_0 + \kappa_1)/2$ and $\sigma = (\kappa_1 - \kappa_0)/2$. This treatment allows the set of M-bin histograms for the target model to be updated based upon their individual correlations given the upper and lower bounds set for the update operation as a whole. We set the constants which govern the model update operations to $\kappa_0 = 0.7$ and $\kappa_1 = 0.9$.

In Figure 1, we present the sample results for frames 250 and 380 of the PETS-ECCV 2004 "Meet and Walk Together 1" sequence. In this sequence, the target appearance varies remarkably as it moves from the bright area into the shade between frames 290 and 310. As a result, the target model is subjected to significant change. Moreover, the target remains close to the other person from frame 330 onwards, which serves as a confounding factor that further complicates the tracking task. Despite these difficulties, the feature combination approach presented here allows the tracker to follow the target throughout the scene. This applies to both of the feature sets under consideration. The VR-based tracker [10], on the other hand, loses the target as the subject approaches the other person between frames 380 and 420. The PCA-based approach [11], however,



(a) (b) (c) (d)

Fig. 2. Target Center Error for our method, the on-line VR-based tracker [10], and the on-line PCA-based tracker [11]. (a)(b): "Meet and Walk Together 1" sequence using the 49-feature set and the 11-feature set, respectively; (c)(d): "Group Walk 1" sequence using the 49-feature set and the 11-feature set, respectively.

cannot adapt to the significant change in illumination and subsequently fails in tracking the target from frame 290 until the end of the footage.

We present a more quantitative analysis of the tracker performance in Figure 2(a) and (b). In these figures, we have plotted the target centre error as a function of frame index with respect to the ground truth provided with the PETS-ECCV 2004 dataset. For the sake of clarity, the error in the figure is shown in a logarithmic scale. Note that our tracker has the lowest mean target centre errors of 5.63 ± 2.77 pixels and 4.40 ± 3.44 pixels for the 49-feature set and the 11-feature set, respectively. The VR-based tracker [10], has a mislocalisation mean of 15.40 ± 15.44 pixels and 16.22 ± 13.69 pixels, respectively. The PCA-based tracker [11], being unable to track the target after frame 290, has a mean centre error of 49.86 ± 22.34 pixels and 48.27 ± 24.84 pixels. This is consistent with the behaviour described above.

We now turn our attention to the contribution of each feature to the global "strong" shift across the sequence. During the sequence, there are 22 updates during the footage. The first 20 updates occur between frames 250 and 310, in which the target moves across the bright area to the shady region in the scene. As a result, the appearance of the target varies significantly. During this frame range, the features such as the vertical and horizontal 3-rectangle Harr-like are assigned the highest weights, with an overall contribution of approximately 60%. In contrast, colour-based features such as the brightness and the normalised RGB channels are given much lower weights, with a contribution of less than 10%. The last few updates occur after frame 320, corresponding to the frames where the target moves completely inside the shady area. These adjustments reduce the weight of the Harr-like features and increase the contribution of the image brightness.

Moving on to the second of our experimental vehicles, Figure 3 shows the results for frames 250 and 280 of the PETS-ECCV 2004 "Group Walk 1" video sequence. The sequence records a group of four people moving across the scene, of which we track the female target. In this footage, there is no significant illumination change as in the previous sequence. However, as the group approaches the camera, their bodies overlap one another before exiting the scene. The similarity in their outfit colour further complicates the tracking task.

In this sequence, the VR-based tracker [10] performs well in the first 100 frames with both feature sets. However, it quickly loses the target once the target is partially occluded by another member of the group. This results in target centre errors of up to 46.47 ± 49.50 pixels and 46.75 ± 47.52 pixels, as shown in Figure 2 (c) and (d).

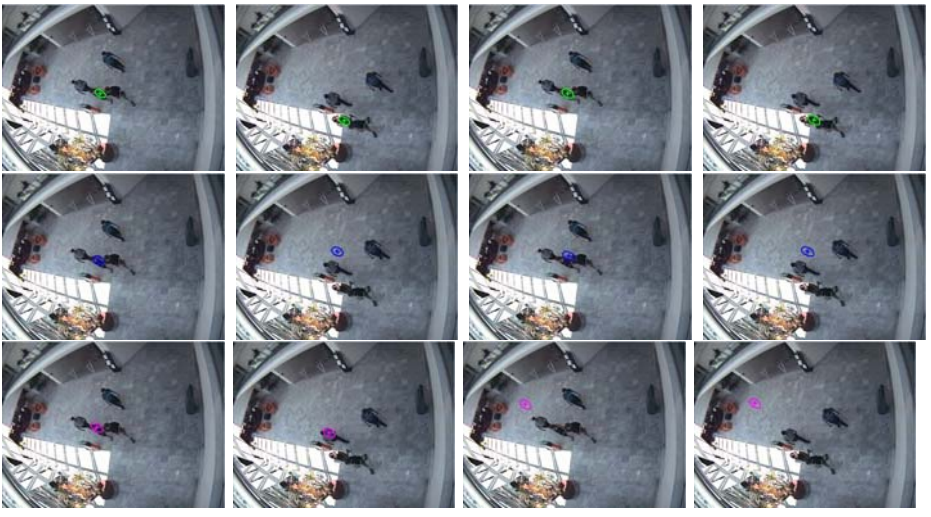


Fig. 3. Results for the "Group Walk 1"sequence at frames 250 and 280. From top-to-bottom: results yielded by our algorithm using the 49-feature set (first and second columns) and the 11-feature set (third and forth columns), the on-line VR-based tracker [10] using the 49-feature set and the 11-feature set, and the on-line PCA-based tracker [11] using the 49-feature set and the 11-feature set.

The performance of the PCA-based tracker [11] has high variation across the sequence. In particular, the PCA-tracker shows a similar performance to that of the VR-based tracker when the 49-feature set is used. It also loses the target at the frames where the subject bodies overlap, being unable to recovery afterwards. However, in the 11-feature set case, the PCA-tracker only manages to track the target in the first 20 frames. As a result, the error measurements are significant, 38.42 ± 41.00 pixels and 86.65 ± 60.16 pixels for the 49-feature set and the 11-feature set, respectively. For our tracker, the model integrity is preserved as a consequence of the use of the total correlation as a measure of tracking confidence. Our tracker successfully follows the target throughout the scene with low target centre-errors, i.e. 6.09 ± 3.03 pixels and 6.25 ± 2.31 pixels for the 49-feature set and the 11-feature set, respectively.

On the contribution of each feature to the global "strong" shift, there are 37 updates throughout the footage. These mainly occur when the subject bodies occlude one another. Nonetheless, the vertical 3-rectangle Harr-like feature is dominant across the sequence. From our experiments we also notice that the normalised RGB colour channels are not as discriminant as the other features in the set. This can be attributed to the fact that the clothing colour of the subjects in the scene does not separate the target from the rest of the crowd.

7 Conclusion

In this paper, we have presented a feature combination approach for object tracking. We have shown how the target-centre may be recovered from a weighted linear combination of "weak" mean-shifts. This feature combination method is based upon graph

embedding techniques. Thus, it provides a principled link between feature combination, graph-spectral methods and graphical models. The method performs on-line updating based upon the correlation between the target current model and that of the new target position at the current frame. The updating scheme presented here is governed by the reliability of the tracking process. As a result, our method can cope with confusing backgrounds, unexpected fast movements and temporary occlusions by taking advantage of the information drawn from multiple feature spaces corresponding to a number of visual cues. The approach is quite general in nature and can employ other features elsewhere in the literature. We have also compared our results to those delivered by alternative methods.

References

1. Nascimento, J.C., Marques, J.S.: Robust shape tracking in the presence of cluttered background. IEEE Transactions on Multimedia 6(6), 852–861 (2004)
2. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision 57(2), 137–154 (2002)
3. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. In: European Conf. on Computer Vision., vol. 3, pp. 255–271 (2002)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(5), 564–577 (2003)
5. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 780–785 (1997)
6. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
7. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
8. Stern, H., Efros, B.: Adaptive color space switching for face tracking in multi-colored lighting environments. In: Int. Conf. on Automatic Face and Gesture Recognition, p. 249 (2002)
9. Nguyen, H., Smeulders, A.: Tracking aspects of the foreground against the background. In: European Conf. on Computer Vision., vol. 2, pp. 446–456 (2004)
10. Collins, R., Liu, Y., Leordeanu, M.: On-line selection of discriminative tracking features. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
11. Han, B., Davis, L.: Object tracking by adaptive feature extraction. In: International Conf. on Image Processing, vol. 3, pp. 1501–1504 (2004)
12. Avidan, S.: Ensemble tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 494–501 (2005)
13. Grabner, H., Bischof, H.: On-line boosting and vision. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 260–267 (2006)
14. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
15. Freund, Y.: Boosting a weak learning algorithm by majority. In: Proceedings of the Workshop on Computational Learning Theory, pp. 202–216 (1990)
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
17. Chavel, I.: Riemannian Geometry: A Modern Introduction. Cambridge University Press, Cambridge (1995)
18. Robles-Kelly, A.: Segmentation via graph-spectral methods and riemannian geometry. In: International Conf. on Computer Analysis of Images and Patterns, pp. 661–668 (2005)
19. Borg, I., Groenen, P.: Modern Multidimensional Scaling, Theory and Applications. Springer Series in Statistics. Springer, Heidelberg (1997)

A Smarter Particle Filter

Xiaoqin Zhang[1], Weiming Hu[1], and Steve Maybank[2]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Beijing, China
{xqzhang,wmhu}@nlpr.ia.ac.cn
[2] School of Computer Science and Information Systems, Birkbeck College, London, UK
sjmaybank@dcs.bbk.ac.uk

Abstract. Particle filtering is an effective sequential Monte Carlo approach to solve the recursive Bayesian filtering problem in non-linear and non-Gaussian systems. The algorithm is based on importance sampling. However, in the literature, the proper choice of the proposal distribution for importance sampling remains a tough task and has not been resolved yet. Inspired by the animal swarm intelligence in the evolutionary computing, we propose a swarm intelligence based particle filter algorithm. Unlike the independent particles in the conventional particle filter, the particles in our algorithm cooperate with each other and evolve according to the *cognitive effect* and *social effect* in analogy with the cooperative and social aspects of animal populations. Furthermore, the theoretical analysis shows that our algorithm is essentially a conventional particle filter with a hierarchial importance sampling process which is guided by the swarm intelligence extracted from the particle configuration, and thus greatly overcome the sample impoverishment problem suffered by particle filters. We compare the proposed approach with several nonlinear filters in the following tasks: state estimation, and visual tracking. The experiments demonstrate the effectiveness and promise of our approach.

1 Introduction

Particle filters have been extensively studied in the computer vision and pattern recognition community due to its crucial value in numerous applications including visual tracking, robot localization, machine learning, and signal processing.

Essentially, particle filter is a sequential Monte Carlo approach to solve the recursive Bayesian filtering problem, which combines the powerful Monte Carlo sampling technique with Bayesian inference. It relaxes the linearity and Gaussianity constraints of the Kalman filter and provides a tractable solution to non-linear and non-Gaussian problems. The basic idea of particle filtering is to use a number of independent random variables called particles, sampled from a proposal distribution, to represent the posterior probability, and to update the posterior by involving the new observations. The particles is properly propagated and weighted recursively according to the Bayesian rule. Although particle filtering has achieved a considerable success in the analysis of sequential time series, it is faced with a fatal problem-sample impoverishment due to its suboptimal sampling mechanism, based on a proposal distribution. When the proposal distribution is concentrated in the tail of the observation distribution the performance of

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 236–246, 2010.

the particle filter is very poor since most particles have low weights, thereby leading to the well-known sample impoverishment problem.

Recently PSO (particle swarm optimization) [1,2,3,4,5], a new population based stochastic optimization technique, has received more and more attention because of its considerable success. Unlike the independent particles in the particle filter, the particles in PSO interact locally with one another and with their environment in analogy with the cooperative and social aspects of animal populations, for example as found in birds flocking. Starting from a diffuse population, now called a swarm, individuals, now termed particles, tend to move in the state space and eventually cluster in regions where optimal state is located. The advantages of this mechanism are, on one hand, the robustness and sophistication of the obtained group behavior and, on the other hand, the simplicity and low cost of the computation associated with each particle.

Inspired by the forgoing discussions, we propose a swarm intelligence based particle filter algorithm, in which the particles are viewed as intelligent individuals, e.g. birds, and evolve through communicating and cooperating with each other. Meanwhile, we also conduct a theoretical analysis from a 'Bayesian filtering' perspective, and find that the proposed algorithm is essentially a conventional particle filter with a hierarchial importance sampling process. The hierarchial importance sampling process which consists of two stages: 1) a coarse sampling from the state transition distribution $p(x_t|x_{t-1})$, 2) a fine sampling carried out by the PSO iterations which are based on the 'cognitive' and 'social' aspects of particle populations. In this way, the newest observations are gradually taken into consideration to approximate the sampling results from the optimal proposal distribution $p(x_t|x_{t-1}, y_t)$ [6], and thereby overcome the sample impoverishment problem suffered by convectional particle filters.

This paper is arranged as follows. The standard particle filter and its limitation are presented in Section 2. The proposed annealed Gaussian based particle swarm optimization is introduced in Section 3. Section 4 gives a detailed description of the smarter particle filter and its theoretical analysis. Experimental results are shown in Section 5, and Section 6 is devoted to conclusion.

2 Particle Filter and Its Limitation

To make this paper self-contained, we first briefly review the conventional particle filter, which is described in more detail in [7], and then summarize its major limitation.

2.1 Particle Filter

The particle filter is an on-line Bayesian inference process for estimating the unknown state x_t at time t from sequential observations $y_{1:t}$ perturbed by noise. A dynamic state-space form employed in the Bayesian inference framework is shown as follows [7],

$$\text{state transition model } x_t = f_t(x_{t-1}, \epsilon_t) \leftrightarrow p(x_t|x_{t-1}) \tag{1}$$

$$\text{observation model } y_t = h_t(x_t, \nu_t) \leftrightarrow p(y_t|x_t) \tag{2}$$

where x_t, y_t represent system state and observation, ϵ_t, ν_t are the system noise and observation noise. $f_t(.,.)$ and $h_t(.,.)$ are the state transition and observation models,

which are determined by probability distributions $p(x_t|x_{t-1})$ and $p(y_t|x_t)$ respectively. The Bayesian inference process is achieved by

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{3}$$

where the prior $p(x_t|y_{1:t-1})$ is the propagation of the previous posterior along the temporal axis,

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{4}$$

When the state transition and observation models are nonlinear and non-Gaussian, the above integration is intractable and one has to resort to numerical approximations such as particle filters. The basic idea of particle filter is to use a number particles $\{x_t^i\}_{i=1}^N$, sampled directly from the state space, to approximate the posterior distribution. Thus the posterior can be formulated as $p(x_t|y_{1:t}) = \frac{1}{N}\sum_{i=1}^N \delta(x_t - x_t^i)$, where $\delta(\cdot)$ is the Dirac function. Since it is usually impossible to sample from the true posterior, an easy-to-implement distribution, the so-called *proposal distribution* denoted by $q(\cdot)$ is employed, hence $x_t^i \sim q(x_t|x_{t-1}^i, y_{1:t})$, $(i = 1, \cdots, N)$, then each particle's weight is set to

$$w_t^i \propto \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t|x_{t-1}^i, y_{1:t})}. \tag{5}$$

Finally, the posterior probability distribution is approximated as $p(x_t|y_{1:t}) = \sum_{i=1}^N w_t^i$ $\delta(x_t - x_t^i)$. After the importance sampling step, a re-sampling step is adopted to ensure the efficiency of the particles' evolution. To summarize, the detail process of particle filter is presented in Algorithm 1.

Algorithm 1. Particle Filter

1. Initialization: for $n = 1, \cdots, N$, sample $x_0^{(n)} \sim p(x_0)$, $w_0^{(n)} = 1/N$.
2. For time steps $t = 1, 2, \cdots$
3. Importance Sampling: for $n = 1, \cdots, N$, draw samples from the importance proposal distribution as follows:
$$\tilde{x}_t^{(n)} \sim q(x_t|x_{t-1}^{(n)}, y_{1:t})$$
4. Weight update: evaluate the importance weights with Equation (5).
5. Normalize the importance weights:
$$\tilde{w}_t^{(n)} = \frac{w_t^{(n)}}{\sum_{i=1}^N w_t^{(i)}}$$
6. Output the statistics of the particles: MMSE or MAP estimate.
7. Resampling: generate N new particles $x_t^{(n)}$ from the set $\{\tilde{x}_t^{(n)}\}_{n=1}^N$ according to the importance weights $\{\tilde{w}_t^{(n)}\}$.
8. Repeat Steps 3 to 7.

2.2 Limitation

The proposal distribution $q(\cdot)$ is critically important for a successful particle filter because it concerns putting the sampling particles in the useful areas where the posterior

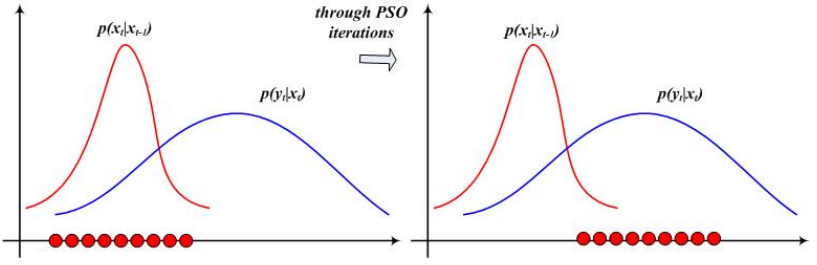


Fig. 1. An illustration of importance sampling (left: sample from $p(x_t|x_{t-1})$, right: after PSO iterations)

is significant. In practice, the state transition distribution $p(x_t|x_{t-1})$ is usually taken as the proposal distribution for its simplicity. However, this proposal distribution contains little information about the current observations, consequently resulting to a inefficient sampling.

As shown in Fig.1(a), when the transition model is situated in the tail of the observation distribution, then the weight of most particles are low, thereby leading to the sample impoverishment problem.

3 Annealed Gaussian Based PSO

3.1 Traditional PSO

Particle swarm optimization [1], is a population based stochastic optimization technique, which is inspired by the social behavior of bird flocking. In detail, a PSO algorithm is initialized with a group of random particles $\{x^{i,0}\}_{i=1}^{N}$ (N is the number of particles). Each particle $x^{i,0}$ has a corresponding fitness value which is evaluated by a fitness model $f(x^{i,0})$, and has a relevant velocity $v^{i,0}$ which is a function of the best state found by that particle (p^i, for individual best), and of the best state found so far among all particles (g, for global best). Given these two best values, each particle updates its velocity and state with following equations in the nth iteration,

$$v^{i,n+1} = w^n v^{i,n} + \varphi_1 u_1 (p^i - x^{i,n}) + \varphi_2 u_2 (g - x^{i,n}) \tag{6}$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \tag{7}$$

where w^n is the inertial weight, the φ_1, φ_2 are acceleration constants, and $u_1, u_2 \in (0, 1)$ are uniformly distributed random numbers. The inertial weight w is usually a monotonically decreasing function of the iteration n. For example, given a user-specified maximum weight w_{max} and a minimum weight w_{min}, one way to update w is as follows:

$$w^{n+1} = w^n - dw,\ dw = (w_{max} - w_{min})/T \tag{8}$$

where T is the maximum iteration number. In Eq.(6), the three different parts represent *inertial velocity*, *cognitive effect* and *social effect* respectively. After the nth iteration, the fitness value of each particle is evaluated by a predefined fitness model as follows.

$$f(x^{i,n+1}) = p(y^{i,n+1}|x^{i,n+1}) \tag{9}$$

where $y^{i,n+1}$ is the observation corresponding to the state $x^{i,n+1}$. Then the individual best and global best of the particles are updated in the following equations:

$$p^i = \begin{cases} x^{i,n+1}, & \textbf{if } f(x^{i,n+1}) > f(p^i) \\ p^i, & \textbf{else} \end{cases} \tag{10}$$

$$g = \arg\max_{p^i} f(p^i) \tag{11}$$

In this way, the particles search for the optima through the above iterations until the fitness value of g reaches a certain threshold or the maximum iteration number is encountered.

3.2 Annealed Gaussian Based PSO

In the above version of PSO algorithm, there are several parameters to be tuned: inertial weights w^n, acceleration constants φ_1, φ_2. There is a lack of a mechanism for controlling of these parameters, which fosters the danger of swarm explosion and divergence especially in high dimensions. Therefore, we propose an annealed Gaussian based particle swarm optimization (AGPSO) algorithm, where the particles and their velocities are updated in the following way,

$$v^{i,n+1} = |r_1|(p^i - x^{i,n}) + |r_2|(g - x^{i,n}) + \eta \tag{12}$$

$$x^{i,n+1} = x^{i,n} + v^{i,n+1} \tag{13}$$

where r_1, r_2 are random numbers sampled from the Gaussian probability distribution $\mathcal{N}(0,1)$, and η is zero-mean Gaussian perturbation noise to avoid trapping in local optima whose covariance matrix is changed in an adaptive simulated annealing way [8]:

$$\Sigma_\eta^n = \Sigma e^{-cn} \tag{14}$$

where Σ is the covariance matrix of the predefined transition distribution, c is an annealing constant, and n is the iteration number. Compared with the traditional PSO, it has two major merits: a) a big reduction in the number of parameters–there is a single annealing parameter, b) it converges much faster than traditional PSO (see Section 5.1).

4 Swarm Intelligence Based Particle Filter

4.1 Motivation

In [6], it is shown that the 'optimal' importance proposal distribution is $p(x_t|x_{t-1}^i, y_t)$ in the sense of minimizing the variance of the importance weights. However, in practice, it is impossible to use $p(x_t|x_{t-1}^i, y_t)$ as the proposal distribution in the non-linear and non-Gaussian cases, since it is difficult to sample from $p(x_t|x_{t-1}^i, y_t)$ and to evaluate $p(y_t|x_{t-1}^i) = \int p(y_t|x_t)p(x_t|x_{t-1}^i)dx_t$. So the question is, how to incorporate the current observation y_t into the transition distribution $p(x_t|x_{t-1})$ to form an effective proposal distribution at a reasonable computation cost.

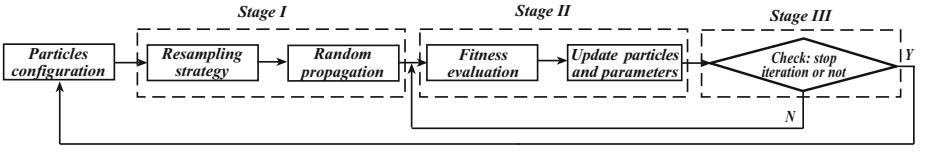


Fig. 2. Overview of the proposed algorithm

4.2 The Smarter Particle Filter

From the description of Section 3, we can see that the PSO iterations can naturally take the observation into consideration, since the particles cooperate and evolve according to their fitness values which are updated by their corresponding observations. Inspired by this property of the PSO, we propose a swarm intelligence based particle filter, in which the particles are firstly propagated by the state transition model, and then corporately evolve according to the PSO iterations.

To give a clear view, the flowchart of the swarm intelligence based particle filter is shown in Fig.2. First, the individual best of particles from the previous time $t-1$ are resampled and randomly propagated by state transition model to enhance their diversities. Then, by moving the particle swarm towards the particle with the best fitness value, PSO drives all particles towards high likelihood regions. Finally, when the fitness value of g_t reaches a certain threshold or the maximum iteration number is encountered, the optimized sampling process is stopped. The global best g_t or the mean of individual best p_t^i is output as the maximum a posterior (MAP) estimate or minimum mean square error (MMSE) estimate. The details of the proposed algorithm are as follows.

1. **Input:** the N individual best particles $\{p_{t-1}^i\}_{i=1}^N$ at time $t-1$;
2. Resample the above particles according to their fitness value, resulting to a new particle set $\{\tilde{p}_{t-1}^i\}_{i=1}^N$;
3. Randomly propagate the particle set to enhance their diversities according to the following transition model
$$x_t^{i,0} \sim p(x_t|\tilde{p}_{t-1}^i)$$
4. **for** $n = 0, 1, 2, \ldots, T$ **do**
5. Carry out the PSO iteration based on Equations (12),(13)
$$v_t^{i,n+1} = |r_1|(p_t^i - x_t^{i,n}) + |r_2|(g_t - x_t^{i,n}) + \eta$$
$$x_t^{i,n+1} = x_t^{i,n} + v_t^{i,n+1}$$
6. Evaluate the fitness values
$$f(x_t^{i,n+1}) = p(y_t^{i,n+1}|x_t^{i,n+1})$$
where $y_t^{i,n+1}$ is the observation corresponding to $x_t^{i,n+1}$;
7. Update the two best particles and the covariance matrix
$$p_t^i = \begin{cases} x_t^{i,n+1}, & \textbf{if } f(x_t^{i,n+1}) > f(p_t^i) \\ p_t^i, & \textbf{else} \end{cases}, \quad g_t = \arg\max_{p_t^i} f(p_t^i)$$
$$\Sigma_\eta^{n+1} = \Sigma e^{-c(n+1)}$$
8. Check the convergence criterion;
9. If satisfied, **break**;
10. **end for**
11. **Output:** the global best g_t or the mean of $\{p_t^i\}_{i=1}^N$;

4.3 Theoretical Analysis from Bayesian Filtering View

In this part, we conduct a theoretical analysis of our algorithm from a Bayesian filtering view, and show why our algorithm improves on the particle filter.

Hierarchical Importance Sampling In our algorithm as described in Section 4.2, we take a two-stage sampling strategy to generate samples that approximate to the 'optimal' proposal distribution: first, the particles are sampled from the state transition distribution $p(x_t|x_{t-1})$; second, the sampled particles evolve through the PSO iterations to obtain the final importance sampling.

From the particle filtering view, we can see that our strategy is essentially a hierarchical importance sampling. In the coarse importance sampling stage, the particles are firstly sampled from the state transition distribution as in conventional particle filters to enhance their diversity.

$$x_t^{i,0} \sim p(x_t|\tilde{p}_{t-1}^i) \tag{15}$$

In the fine importance sampling stage, the particles evolve through PSO iterations, and are updated according to the newest observations. In fact, this is essentially a latent multi-layer importance sampling process with an implicit proposal distribution. Suppose $x_t \in \mathbb{R}^d$ be d-dimensional state, let's focus on one PSO iteration in Section 4.2, suppose $x_t \in \mathbb{R}^d$ is a d-dimensional state, the distribution of the lth element in the vector $|r_1|(p_t^i - x_t^{i,n})$ is as follows:

$$|r_1|(p_t^i - x_t^{i,n})_l, \sim \begin{cases} 2\mathcal{N}(0, (p_t^i - x_t^{i,n})_l^2)\ [0, +\infty), \ \textbf{if}\ \ (p_t^i - x_t^{i,n})_l \geq 0 \\ 2\mathcal{N}(0, (p_t^i - x_t^{i,n})_l^2)\ (-\infty, 0), \ \textbf{else} \end{cases}$$

where $l = 1, \cdots, d$, so the distribution of $|r_1|(p_t^i - x_t^{i,n})$ is

$$|r_1|(p_t^i - x_t^{i,n}) \sim R_1 = 2\mathcal{N}(0, \Sigma_1),\ \Sigma_1 = \begin{pmatrix} (p_t^i - x_t^{i,n})_1^2 & \mathbf{0} \\ & \ddots \\ \mathbf{0} & (p_t^i - x_t^{i,n})_d^2 \end{pmatrix}$$

Similarly available,

$$|r_2|(g_t - x_t^{i,n}) \sim R_2 = 2\mathcal{N}(0, \Sigma_2),\ \Sigma_2 = \begin{pmatrix} (g_t - x_t^{i,n})_1^2 & \mathbf{0} \\ & \ddots \\ \mathbf{0} & (g_t - x_t^{i,n})_d^2 \end{pmatrix}$$

Together with $\eta \sim R_3 = \mathcal{N}(0, \Sigma_\eta)$, the implicit proposal distribution behind a PSO iteration is $R = R_1 * R_2 * R_3$[1] with a $x_t^{i,n}$ translation. Here $*$ stands for convolution operator.

In this way, the PSO iterations can naturally take the current observation y_t into consideration, since $\{p_t^i\}_{i=1}^N$ and g_t are updated to their observations. Therefore, with

[1] Since the analytical form of R is not available, we called it latent sampling process.

coarse importance sampling stage from the state transition distribution $p(x_t|\tilde{p}^i_{t-1})$, the hierarchical sampling process can approximate to the optimal sampling from $p(x_t|x^i_{t-1}, y_t)$.

As shown in Fig.1, when the transition distribution is situated in the tail of the observation likelihood, the particles directly drawn from this distribution do not cover a significant region of the likelihood, and thus the importance weights of most particles are low, resulting to unfavorable performance. In contrast, through hierarchial sampling process in our algorithm, the particles are moved towards the region where the likelihood of observation has larger values, and are finally relocated to the dominant modes of the likelihood, demonstrating the effectiveness of our sampling strategy.

5 Experimental Results

We compare the performance of our algorithm to several non-linear filters on two estimation problems: 1) a synthetic state estimation problem; 2) real world visual tracking problem. All of the experiments are carried out on a CPU Pentium IV 3.2GHz PC with 512M memory[2].

5.1 State Estimation

The algorithm is firstly tested on a non-linear state estimation problem, which is described as benchmark in many papers [9]. Consider the following nonlinear state transition model given by

$$x_t = 1 + \sin(w\pi(t-1)) + \phi_1 x_{t-1} + v_{t-1}, \quad x_t \in \mathbb{R} \tag{16}$$

where v_{t-1} is a Gamma $\mathcal{G}a(3,2)$ random variable modeling the process noise, and $w = 4e-2$ and $\phi_1 = 0.5$ are scalar parameters. A non-stationary observation model is as follows

$$y_t = \begin{cases} \phi_2 x_t^2 + n_t, & t \leq 30 \\ \phi_3 x_t - 2 + n_t, & t > 30 \end{cases} \tag{17}$$

where $\phi_2 = 0.2, \phi_3 = 0.5$, and the observation noise n_t is drawn from a Gaussian distribution $\mathcal{N}(0, 0.00001)$. Given only the noisy observation y_t, several filters are used to estimate the underlying state sequence x_t for $t = 1 \cdots 60$. Here, we compare our algorithm (with AGPSO) with conventional particle filter [7], extended Kalman based particle filter [10], unscented particle filter [9], auxiliary particle filter [11], and our algorithm (with traditional PSO)[3]. For each algorithm, a proposal distribution is chosen as shown in Table 1. The parameters in APSO and PSOPF are set as follows: $\Sigma = 0.8, c = 2, \varphi_1 = \varphi_2 = 1, w_{max} = 0.8, w_{min} = 0.1, T = 20$. Fig.3 gives an illustration of the estimates generated from a single run of the different filters. Compared with other nonlinear filters, our algorithm is more robust to the outlier, where the observation is severely contaminated by the noise. Since the result of a single run is a random variable, the experiment is repeated 100 times with re-initialization to generate

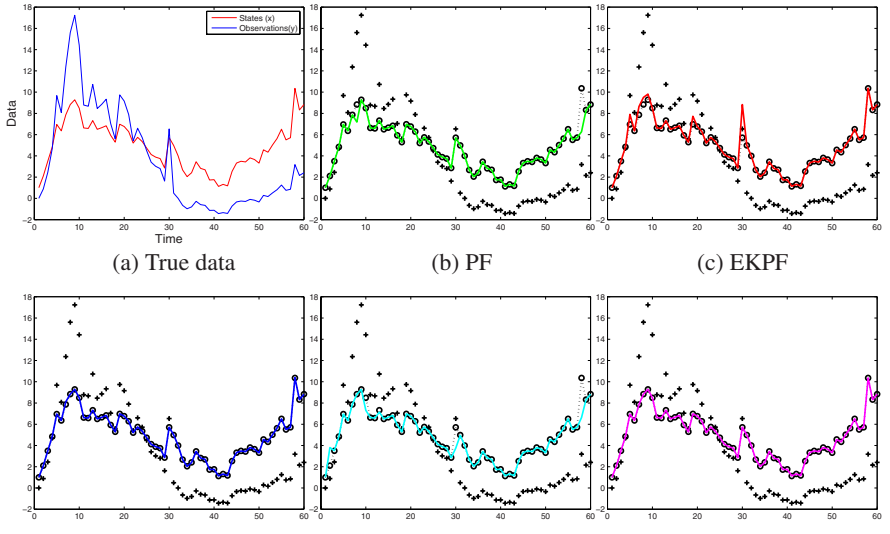
[2] The data and code used in these experiments are available by writing to the authors.

[3] We call these filters AGPSOPF, PF, EKPF, UPF, APF, PSOPF respectively for short in the following parts.

Table 1. Experimental results of state estimation

Algorithm	Proposal	MSE mean	MSE var	Time(s)
Particle filter (PF)	$p(x_t\|x_{t-1})$	0.42225	0.045589	3.6939
Extended Kalman particle filter (EKPF)	$N(\bar{x}_t, P_t)$	0.31129	0.015167	13.014
Unscented particle filter (UPF)	$N(\bar{x}_t, P_t)$	0.06977	0.024894	26.2815
Auxiliary particle filter (APF)	$p(x_t\|x_{t-1})$	0.55196	0.037047	7.1835
Our algorithm (with PSO)	$p(x_t\|x_{t-1})$	0.13019	0.044086	10.2087
Our algorithm (with AGPSO)	$p(x_t\|x_{t-1})$	0.060502	0.06852	6.8005

statistical averages. Table 1 summarizes the performance of all the different filters in the following aspects: the means, variances of the mean-square-error (MSE) of the state estimates and the average execute time for one run. It is obvious that the average accuracy of our algorithm is better than generic PF, EKPF, APF and comparable to that of UPF. However, the real-time performance of our algorithm is much better than UPF as Table 1 shows. Meanwhile, we can see that AGPSOPF can achieve a much faster convergence rate than PSOPF. This is because the velocity part employed in Eq.(6) carries little information, while the annealing part in our PSO iterations enhances the diversity of the particle set and its adaptive effect enables a fast convergence rate. In summary, the total performance of our algorithm prevails over that of other nonlinear filters.



(a) True data (b) PF (c) EKPF

(d) UPF (e) APF (f) Our algorithm

Fig. 3. An illustration of a single run of different filters

5.2 Visual Tracking

In this part, we apply these filters (except EKPF and PSOPF) to a rapid motion tracking task to further demonstrate the effectiveness of the sampling strategy in our algorithm.

Fig. 4. Tracking performances of a human face with rapid motion (green: PF, blue: UPF, cyan: APF, magenta: our algorithm)

This video sequence[4] contains a human face with a rapid motion (see Fig.4). In tracking application, $p(x_t|x_{t-1})$ is used to model the object motion, so when $p(x_t|x_{t-1})$ is not coincident with the actual motion, the sampling directly from $p(x_t|x_{t-1})$ will not be efficient. Therefore, although this sequence seems simple, its rapid and arbitrary motion is a challenge for the different improvements of sampling strategy.

In our implementation, we adopt an incremental learned subspace based appearance model [12] for observation evaluation, and we consider only translational motion $x = (t_x, t_y)$ for simplicity, since our goal is to test the sampling efficiency of all the non-linear filters. Here, $p(x_t|x_{t-1})$ is set to a Gaussian distribution with a covariance matrix $\Sigma = diag(8^2, 8^2)$, and the annealing const is also set to 0.3, and the particle number is set to 200. As shown in Fig.4, the PF based tracker and APF based tracker soon fail to track the object, because the particles directly sampled from the state transition distribution $\mathcal{N}(x_{t-1}, \Sigma_s)$ can not catch the rapid motion of the object, and thus the weights of most particles are low, leading to the tracking failure. More particles and an enlargement for the diagonal elements of the covariance matrix would improve its performance, but this strategy involves more noises and a heavy computational load, and it may trap in the curse of dimensionality when the dimension of the state increases. While the UPF based tracker can follow the object throughout the sequence, the localization accuracy is unsatisfactory. In comparison, our algorithm, which evolves the particles by the swarm intelligence based importance sampling, never loses the target and achieves the most accurate results. Furthermore, we have conducted a quantitative evaluation of these algorithms, and have a comparison in the following aspects: frames of successful tracking, RMSE (root mean square error) between the estimated position and the labeled groundtruth, and average tracking time of each frame. In Table 3, our algorithm outperforms the other filters based trackers in accuracy with a reasonable sacrifice of speed, which witnesses the effectiveness our sampling strategy.

Table 2. Quantitative results of the tracking performance

Algorithm	Frames Tracked	RMSE of Position (by pixels)	Average Tracking Time (by seconds)
PF	5/31	33.4580	0.051
UPF	31/31	3.5097	80.785
APF	4/31	38.7260	0.098
AGPSO	31/31	2.0112	0.731

[4] The sequence is available at http://vision.stanford.edu/ birch/headtracker/seq/.

6 Conclusion

In this paper, we propose a swarm intelligence based particle filter to overcome the sample impoverishment problem. Unlike the independent particles in the convectional particle filters, the particles in our algorithm cooperate each other and evolve according to the *cognitive effect* and *social effect* in analogy with the cooperative and social aspects of animal populations. We conduct a theoretical analysis in a Bayesian filtering view, and find that our algorithm is essentially a convectional particle filter with a hierarchial importance sampling process which is guided by the swarm intelligence extracted from particle configuration. The experimental results demonstrate the effectiveness and promise of our approach.

Acknowledgment

This work is partly supported by NSFC (Grant No. 60825204, 60672040, 60705003) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453, 2009AA01Z318).

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
2. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in amultidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
3. Wachowiak, M., Smolikova, R., Zheng, Y., Zurada, J., Elmaghraby, A.: An approach to multimodal biomedical image registration utilizing particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 289–301 (2004)
4. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
5. Zhang, X., Hu, W., Li, W., Qu, W., Maybank, S.: Multi-object tracking via species based particle swarm optimization. In: Proceedings of InternationalWorkshop on Visual Surveillance (2009)
6. Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing 10(3), 197–208 (2000)
7. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing 50(2), 174–188 (2002)
8. Ingber, L.: Simulated annealing: Practice versus theory. Journal of Mathematical and Computer Modeling 18(1), 29–57 (1993)
9. Merwe, R., Doucet, A., Freitas, N., Wan, E.: The unscented particle filter. In: Advances in Neural Information Processing Systems (2001)
10. Freitas, D., Niranjan, M., Gee, A., Doucet, A.: Sequential monte carlo methods to train neural network models. Neural Computation 12(4), 955–993 (2000)
11. Pitt, M., Shephard, N.: Filtering via simulation: Auxiliary particle filters. Journal of the American Statistical Association 94(446), 590–591 (1999)
12. Lim, J., Ross, D., Lin, R., Yang, M.: Incremental learning for visual tracking. In: Advances in Neural Information Processing Systems, vol. 17, pp. 793–800 (2005)

Robust Real-Time Multiple Target Tracking

Nicolai von Hoyningen-Huene and Michael Beetz

Intelligent Autonomous Systems Group,
Technische Universität München,
Boltzmannstr. 3,
85748 Garching, Germany
{hoyninge,beetz}@cs.tum.edu

Abstract. We propose a novel efficient algorithm for robust tracking of a fixed number of targets in real-time with low failure rate. The method is an instance of Sequential Importance Resampling filters approximating the posterior of complete target configurations as a mixture of Gaussians. Using predicted target positions by Kalman filters, data associations are sampled for each measurement sweep according to their likelihood allowing to constrain the number of associations per target. Updated target configurations are weighted for resampling pursuant to their explanatory power for former positions and measurements. Fixed-lag of the resulting positions increases the tracking quality while smart resampling and memoization decrease the computational demand. We present both, qualitative and quantitative experimental results on two demanding real-world applications with occluded and highly confusable targets, demonstrating the robustness and real-time performance of our approach outperforming current state-of-the-art.

1 Introduction

Low cost and high availability of digital cameras offer opportunities in traditional surveillance tasks and open up new fields like automatic sports analysis or studies of social insect behavior. While the available video footage grows constantly, these data need to be examined. Robust automatic tracking of multiple targets in video is a key feature to assist or avoid expensive and tedious human interactions.

We present a multiple target tracking algorithm that can follow more than twenty similar targets robustly over long sequences in real-time. The proposed method constitutes a Rao-Blackwellized Resampling Particle filter with fixed-lag estimates. The posterior of target positions given the observed measurements is approximated as a mixture of Gaussians approving the use of Kalman filters. The data association problem is solved by sampling according to the likelihood of an assignment for one measurement sweep. Multiple measurements can be assigned to the same target following a Poisson distribution. While smart resampling and memoization allows for real-time capability, fixed time delay for the estimates offers an increase in robustness.

We applied our method to demanding sequences in the soccer and insects domain as they contain a high volume of similar and confusable targets with

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 247–256, 2010.

natural motion. Our algorithm exhibits higher quality in tracking at less computational time than the current state-of-the-art multiple target tracking method by Khan et al. [1]. Lower failure rate can be achieved by disallowing merged measurements and restricting multiple measurements to a Poisson distribution, which better matches reality. Performance gain is due to the direct sampling of associations instead of running a Markov chain, better exploiting recyclability and avoiding uninformative computations like burn-in steps.

After a survey of related work in the field of multiple target tracking, we detail our new Rao-Blackwellized Resampling Particle filter in section 3. Section 4 depicts the experimental results for soccer and ant tracking with comparison to the state-of-the-art. We finish in section 5 with our conclusions.

2 Related Work

The problem of tracking is to recursively estimate an unknown state based on limited observations. Tracking algorithms follow predominantly a Bayesian approach approximating the posterior probability density function (pdf) of the target states given all measurements up to that time, an initial target distribution and the process of how positions evolve over time (motion model) and how measurements inform about target states (sensor model).

In single target tracking two approaches are widely used. The Kalman filter [2] constrains the target state distribution to a Gaussian, consists of a predict and an update step and has shown to be the optimal estimator for linear motion and sensor model. Several suboptimal extensions as e.g. the Extended and the Unscented Kalman filter have been proposed for nonlinear motion and/or sensor models and additional constraints. The second approach known as Particle filter or sequential Monte Carlo method (SMC) approximates arbitrary probability density functions (pdf) on discrete points (particles) only (see [3,4]) yielding a fast tracking method also for nonlinear motion and sensor models.

Multiple-target tracking algorithms differ from single target tracking by the problem of associating each measurement with an appropriate target which is known as data association. Multiple target tracking approaches can be categorized by their handling of the data associations problem.

The Nearest-Neighbor Data Association (NN) assigns each measurement to the closest target mostly based on the Mahalanobis distance (e.g. [5]). The Joint Probabilistic Data Association filter (JPDAF) forms a sub optimal Bayesian algorithm that approximates the posterior distributions of the targets as separate Gaussians for each target, that is assigned to all measurements with weights depending on the predicted association probability (see [6,2]).

Multiple hypothesis tracking (MHT) [6] builds a (mostly pruned) tree of all possible association sequences of each measurement with close targets. The restriction to single associations only as well as the use of Kalman filters and the Hungarian method to find the k best global associations allow computation in polynomial time, but inhibit to handle multiple or merged associations. The Probabilistic MHT (PMHT) [7] does not attempt to enumerate all possible

combinations of feasible data association links, but uses a probabilistic structure derived using expectation-maximization.

Markov Chain Monte Carlo (MCMC) methods sample the data associations based on an importance density by starting with an initial association and proposing local modifications of it (e.g. associate, dissociate or swap) that are accepted with a special acceptance ratio. The sampling performs a Markov Chain over associations in a Bayesian graph where the transition probabilities are chosen so that the stationary distribution of the chain converges to the density of the data associations. The Markov Chain is usually run for a burn-in time for initialization of real sampling, the relative frequency of the sampled associations form the desired pdf. Khan et al. proposed in [1,8] a real-time Rao-Blackwellized MCMC-based particle filter allowing also sampling of split and merged measurement associations. Counterintuitively for a particle filter the MCMC approach can not easily be parallelized maintaining the correct sampling behavior, although work has been published recently by [9].

The Rao-Blackwellized Monte Carlo data association (RBMCDA) approach by Särkkä [10,11] sequentially samples one association after another estimating target positions as a mixture of Gaussians and handling dependencies between assignments of each single measurement by data association priors. The assumption of independence of the order of data associations in one sweep is made. RBRPF [12] is an extension of RBMCDA introducing smart resampling and memoization, that lead to real-time tracking in the first place, and relaxation of the association independence assumption.

3 Rao-Blackwellized Resampling Particle Filter with Fixed-Lag

Following the Bayesian approach our tracking method approximates the posterior probability density function (pdf) $p\left(x_k|z_{1:k}\right)$ of the target positions x_k at time k given all measurements $z_{1:k}$ seen so far. A particle filter for complete player configurations constitutes the base of our algorithm. The pdf is approximated only at S discrete points x_k^i with weights w_k^i called weighted particles:

$$p\left(x_k|z_{1:k}\right) \approx \sum_{i=1}^{S} w_k^i \delta\left(x_k - x_k^i\right). \tag{1}$$

Each particle consists of Gaussians for all N target states with mean m and covariance V:

$$x_k^i = \left\{\mathcal{N}\left(x_{j,k}^i; m_{j,k}^i, V_{j,k}\right)\right\}\ j = 1, \ldots, N. \tag{2}$$

The main loop of the algorithm is depicted in fig. 1 following the Sample Importance Resampling (SIR) framework described in [4] with a combined step of an early resampling and the drawing of new particles. These steps are merged due to the discrete (but still exponential) number of possible data associations which change the nature of sampling. To save computational time, every particle is predicted once with a given motion model f (also called system model)

$$\hat{x}_k^i = f_k\left(x_{k-1}^i, \Gamma_{k-1}\right) \tag{3}$$

with i.i.d. process noise Γ_{k-1}. Each target state is predicted individually under the assumption that their motion is independent. The prediction is solved analytically which is known as Rao-Blackwellization, which is possible owing to the Gaussian nature of the target states. The most probable associations given predicted positions and measurements are sampled several times according to the weight of their former particle plus a constant minimum number o. An assignment $J_{k,r}^i(j,l)$ of a specific target j to a measurement l is drawn using the importance density

$$p\left(J_{k,r}^i(j,l)\right) = \frac{p\left(z_{l,k}|\hat{x}_{j,k}^i\right)}{p\left(J_{k,r}^i(l,\varnothing)\right) + \sum_j p\left(z_{l,k}|\hat{x}_{j,k}^i\right)} \tag{4}$$

with $J_{k,r}^i(l,\varnothing)$ denoting the measurement to be clutter.

Measurements and target states are linked by the sensor model h_k^r (also called measurement model)

$$z_k = h_k^r\left(\hat{x}_k^i, R_k\right), \tag{5}$$

with i.i.d. measurement noise R_k. The function h_k^r depends on $J_{k,r}^i$ and relates assigned target states to the measurements z_k. If h_k^r is a linear function (written as a matrix H_k^r) and target state and measurements are Gaussian, individual assignment probabilities can be evaluated analytically as

$$p\left(z_{l,k}|\hat{x}_{j,k}^i\right) \sim \mathcal{N}\left(z_{l,k}; H_{j,k}^r \hat{x}_{j,k}^i, H_{j,k}^r V_{j,k}^i {H_{j,k}^r}^T + R_{l,k}\right) \tag{6}$$

with $R_{l,k}$ denoting the covariance of the measurement $z_{l,k}$. The probability for a measurement to be clutter depends on the application, but can mostly be approximated to be uniformly distributed over the sensor space $\mathcal{M}$

$$p\left(J_{k,r}^i(l,\varnothing)\right) \sim |\mathcal{M}|^{-1}. \tag{7}$$

The probability for an assignment can also be influenced by additional constraints like the matching of untracked properties (e.g. color and appearance) and the probability for multiple assignments in one sweep. Multiple measurements for one target are sampled according to a Poisson distribution with $\lambda = p_{sd}$. If a target is tossed not to have an additional assignment, the probability for that assignment is zero and it is therefore not taken into account in the sum of the denominator in eq. 4 for that iteration. Before each sampling from one specific particle all measurements are shuffled randomly to prevent an unwanted prior due to the order of measurements (especially for multiple assignments).

During the sampling, data associations for one particle are checked for identity to skip unnecessary weight and update computations. Each drawn data association $J_{k,r}^i$ for one particle i results in a new particle u with the target states updated according to a weighted sum of predicted and observed states by their uncertainties:

$$m^u_{j,k} = \hat{m}^i_{j,k} + V^u_{j,k} \left(h^r_k\right)^{-1} \left(\sum_{J^i_{k,r}(l,j)} R^{-1}_{l,k} \left(z_{l,k} - h^r_k \left(\hat{m}^i_{j,k} \right) \right) \right) \tag{8}$$

and covariances updated as

$$V^u_{j,k} = \left(\hat{V}^{-1}_{j,k} + \left(h^r_k\right)^{-1} \left(\sum_{J^i_{k,r}(l,j)} R^{-1}_l \right) \right)^{-1} . \tag{9}$$

The inverse of the linear sensor model h^r_k is usually not linear for measurements providing partial information about the state only, but can be solved by copying the missing information from the target state into the measurement. For measurements providing only positional data but target states containing velocity information, eq. 5 can be written as

$$(x_z, y_z) = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix} (x, y, \dot{x}, \dot{y})^T \text{ or } (x_z, y_z, \dot{x}, \dot{y}) = I\,(x, y, \dot{x}, \dot{y})^T \tag{10}$$

with the second (augmented) form being invertible.

The weights of the newly sampled particles u are evaluated as the probability for the former prediction and all assigned measurements given the newly sampled positions including a detection probability p_d for all a assigned targets

$$w^u_k \propto p\left(\hat{x}^i_k | x^u_k\right) p^a_d \left(1 - p_d\right)^{N-a} \prod_{J^i_{k,r}(l,j)} p\left(z_{l,k} | x^u_{j,k}\right) \prod_{J^i_{k,r}(l,\varnothing)} p\left(J^i_{k,r}\left(l, \varnothing\right)\right). \tag{11}$$

The weight is multiplied by the number of times this data association was sampled.

This approach differs from [11], where Särkkä et al. set the weights according to the probability of the associations that were used for sampling, but is similar to [1] where residuals between updated positions and predictions as well as

```
program RBRPF
   input
        {x^i_{k-1}, w^i_{k-1}}_{i=1}^{N_{k-1}}   particles for time k − 1
        z_k                                      measurements at time k
   output
        {x^j_k, w^j_k}_{j=1}^{N_k}               particles for time k
 BEGIN
   FOR i = 1 : N_{k-1}
       x^j_k ~ DRAW-N-RESAMPLE[z_k, x^i_{k-1}]
       Calculate w^j_k according to 11
   END FOR
   Normalize weights: w^i_k = w^i_k ( Σ_{i=1}^{N_s} w^i_k )^{-1}
 END
```

Fig. 1. Main Loop of the Proposed Rao-Blackwellized Resampling Particle Filter

measurements are used. It helps to avoid the hospitality problem where multiple measurements are preferred over only one measurement of high accuracy because the weight for such an association is higher due to smaller resulting covariances in the denominator of the Gaussians.

An estimate of the target states x_k is found by selecting the particle with maximum weight. In the case of fixed-lag estimation, target states are evaluated as particles which descendant is the particle with maximum weight for the fixed time distance δ in the future resulting in higher robustness due to the exploitation of more informations by the cost of a delay.

4 Experimental Results

We conducted two experiments on real world tracking problems with a fixed number of about twenty targets. Target states have been modeled as 2-dimensional position and velocity $m^i_{j,k} = (px, py, \dot{p}x, \dot{p}y)^T$. The motion model was chosen as the discretized Wiener velocity model $A_{\Delta t}$ (see [2]) for time difference Δt between $k-1$ and k as a linear motion model:

$$\hat{m}^i_{j,k} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} m^i_{j,k}, \qquad V'_k = A_{\Delta t} V_{k-1} A^T_{\Delta t} + \begin{pmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{pmatrix} \tilde{q} \tag{12}$$

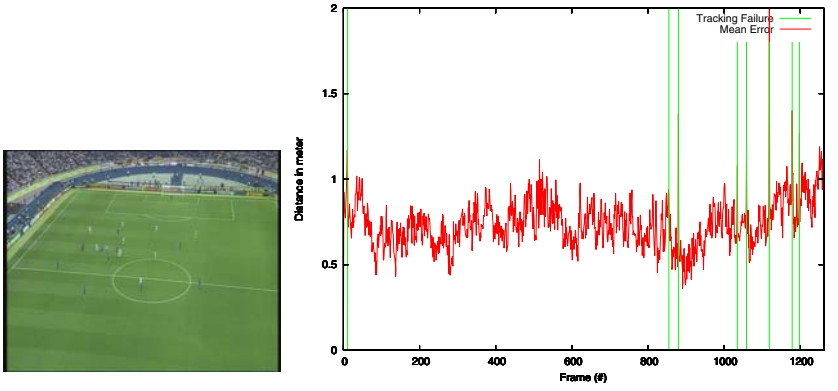
with power spectral density $\tilde{q}$ as a constant factor.

We used the additional number of samplings $o = 10$ through all experiments. Initial positions were given manually.

4.1 Tracking Soccer Players

Identity tracking in sports is an interesting and demanding testing bed for multiple target tracking algorithms due to frequent occlusions of similar targets. The tracker was evaluated as part of the ASPOGAMOsystem [13]. We provide a video sequence of the beginning of the 2006 world championship's final consisting of 1262 frames shot with 25Hz. We tracked all soccer players and the main referee (23 targets) captured by a nonstatic pan-tilt-zoom camera used for TV broadcasting. An example image is depicted in fig. 2(a). Homographies for each frame have been computed using [14]. Groundtruth was collected by manually marking each target position in the video image and transforming it to world coordinates for the whole scene. Mostly all players are visible in the sequence, but goalkeepers and wing players are sometimes not shown due to zooming. The sequence constitutes a demanding test for any multiple target tracking method because of uncertain, missed and occluded targets captured by a moving camera.

The automatic player detection was done following [15] by segmenting possible player regions via thresholding the local variance of the grayscale video image while skipping the field lines. These regions were matched by color templates to



(a) Final of WC2006 (b) Average Distance of Players to Groundtruth

Fig. 2. The Soccer Groundtruth Sequence with 1262 Frames

suppress outliers and to robustly estimate the players center of gravity. The center point was assumed to be at 0.9m above the ground to allow the computation of real-world coordinates by transforming the projected point with the inverse homography. One sweep corresponds to measurements at one frame of the video. The measurement covariances depend on the estimation of the camera parameters and differ also inside one frame depending on the distance to the camera. They are in the range of $[0.17, 12.5]$ in goal and $[0.15, 2.53]$ in sideline direction.

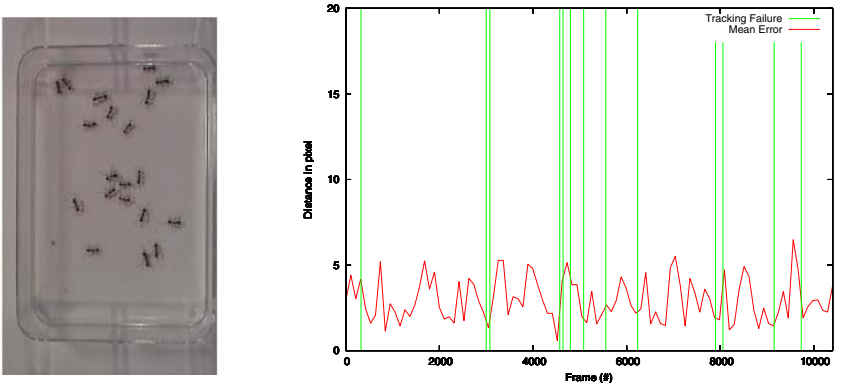
We used the following parameters for the constant velocity model: $\Delta t = 0.04$ (due to 25fps), $\tilde{q} = 0.0008$ (due to max acceleration of humans) and set $p_{sd} = 0.3$, $p_d = 0.3$. Covariances were inited with $V_0 = 0.001 I_{4N}$. Tracking was done with $S = 50$ in real-world coordinates; all positions and covariances are specified in meters.

Failures were counted when a target deviated from the ground truth position by more than 5.0 meters. After a failure, only the failed target was reinitialized to the ground truth position and tracking was resumed. Our method without delay failed 8 times on the soccer sequence with 1262 frames. A delay could not reduce the number of failures further. The mean distance to the groundtruth is depicted in fig. 2(b). The smart resampling resulted in 38.91 ± 13.92 effectively used particles. Tracking without player detection needed $15.05ms \pm 5.077$ per frame resulting in a mean frame rate of 66.4 fps.

4.2 Tracking Ants

In [1] Khan et al. tested their proposed MCMC tracker on a challenging ground truth sequence of twenty ants in a small container. The image data and groundtruth are available online at *http://www.kinetrack.org*.

The ants that should be tracked to gain insights in social behavior of insects are about 1cm long and move as quickly as 3cm per second frequently interacting



(a) 20 Interacting Ants (b) Average Distance of Ants to Groundtruth

Fig. 3. Insects Domain with Frequent Interactions and Similar Target Appearance

with up to five or more ants in close proximity. The test sequence presents a substantial challenge for any multitarget tracking algorithm and was selected for comparison purpose. An image is depicted in fig. 3(a). The sequence consists of 10,400 frames recorded at a resolution of 720×480 pixels at 30 Hz. We used the same simple thresholding procedure of the blurred and sownsampled video as [1] to obtain the measurements.

The number of failures detected on the ground truth sequence for the MCMC tracker with different number of particles and our tracker without and with smoothing are shown in table 1. Failures were counted when a target deviated from the ground truth position by more than 60 pixels. After a failure, all of the targets were reinitialized to the ground truth position and tracking was resumed. We used the same parameters as given in [1]. Measurements contain simple 2D positions $z_l = [x, y]^T$ without velocity. Target motion was modeled using a constant velocity model as mentioned above with time step $\Delta t = 0.033$ and $\tilde{q} = 32$. The initial covariance was set to $V_0 = 32I_{4N}$ and the measurement noise was $R = 32I_{4N}$. All positions and covariances are specified in pixels.

We measured the run time as the average frame rate in frames per second (fps) including image processing time on a 2.2 GHz Dual-core PC and also on a Pentium 4-M 1.6 GHz for better comparability. With current standard hardware our method is able to track the twenty ants faster than real-time (40 fps) with low failure rate. The smoothing reduces the number of failures even further while keeping the frame rate at 40 fps. Our algorithm exhibits higher quality in tracking needing about one half of the computational time than the current state-of-the-art tracker in [1]. Instead of [1] we do not allow merged measurements as these result mostly from the fore target occluding the back and may mislead the tracker. Also we restrict the number of detections of one target by a Poisson distribution with p_{sd}, yielding less (possibly wrong) associations. The speed-up is achieved as we directly sample the associations instead of running a Markov chain, allowing a better constriction to necessary computations by memoization

Table 1. Experimental Results for Tracking Ants through 10,400 Frames

Algorithm	P4-M 1.6GHz	P4-M 3Ghz	Dual Core 2.5Ghz	Fail
MCMC [1] $S = 1$	-	23.03 ± 0.87 fps	-	24
MCMC [1] $S = 6$	-	8.75 ± 0.55 fps	-	21
RBRPF $S = 6$	8.38 ± 1.5 fps	-	40.68 ± 1.0 fps	19
RBRSPF $S = 6$, $\delta = 4$	8.39 ± 1.5 fps	-	40.76 ± 1.0 fps	13

without the need for uninformative burn-in steps. The average distance over all ants is depicted in fig. 3(b). Analogous pictures have been published for the MCMC tracker in [8]. The distance for tracking without smoothing differs only minor to the ones with delay, which is based on the use of Kalman filters to predict and update target states in both approaches. The mean for the average tracking error is with 3.16 pixels low respecting a systematic error caused by the downsampling to forth of the original resolution only.

We also conducted experiments on a second ant dataset of [1] where ants were moving on two glass layers. Khan et al. provide 16 video sequences that were preprocessed in the same way as above to extract measurements from video. The MCMC approach could track through 12 of the 16 demanding sequences successfully with parameters $\Delta t = 0.1$, $V_0 = 32I_{4N}$, $\Gamma = 4I_{4N}$ and $\Sigma_{ii} = 150I_{4N}$ but failed on sequences 5, 8, 12 and 14. Our approach could also handle 12 of the 16 sequences using $p_d = 0.8$, $p_{sd} = 0.14$ but failed on 3, 8, 12 and 16. With $p_{sd} = 0.4$ our method also tracked through sequence 16 successfully. All sequences include longer partial or full occlusions or sudden changes in direction and velocity which makes it a hard task for every tracker assuming a constant velocity model. In average about 40fps could be achieved on the Core 2 Duo with 2.5 GHz emphasizing the real-time capability of RBRPF.

5 Conclusions

We presented the Rao-Blackwellized Resampling Particle filter with Fixed-Lag as a novel multiple target tracking algorithm. The method exhibits real-time performance by exploiting the properties of Gaussians through Rao-Blackwellization and the discreteness together with rareness of probable data associations through smart resampling. Robustness of tracking is increased by retrieving target estimates after a fixed-lag and therefore utilizing more informations. Demanding real-world experiments with frequent interactions and highly similar targets demonstrate the capabilities of our approach, that outperformed the state-of-the-art MCMC method in robustness as well as computational time.

References

1. Khan, Z., Balch, T., Dellaert, F.: MCMC Data Association and Sparse Factorization Updating for Real Time Multitarget Tracking with Merged and Multiple Measurements. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(12), 1960–1972 (2006)

2. Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
3. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. Int. J. Computer Vision 29(1), 5–28 (1998)
4. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. on Signal Processing 50(2) (February 2002)
5. Cai, Y., de Freitas, N., Little, J.J.: Robust Visual Tracking for Multiple Targets. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 107–118. Springer, Heidelberg (2006)
6. Bar-Shalom, Y., Li, X.-R.: Multitarget-Multisensor Tracking: Principles and Techniques. YBS (1995)
7. Streit, R.L.: The PMHT and related applications of mixture densities. In: Proc. Intl. Conf. on Information Fusion (July 2006)
8. Khan, Z., Balch, T., Dellaert, F.: MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. IEEE Trans. on Pattern Analysis and Machine Intelligence (2005)
9. Bardet, F., Chateau, T., Ramadasan, D.: Real-Time Multi-Object Tracking with Few Particles. In: VISAPP, INSTICC, pp. 456–463 (2009)
10. Särkkä, S., Vehtari, A., Lampinen, J.: Rao-Blackwellized Monte Carlo data association for multiple target tracking. In: Proc. of Intl Conf. on Information Fusion, Stockholm, vol. 7 (2004)
11. Särkkä, S., Vehtari, A., Lampinen, J.: Rao-Blackwellized Particle Filter for Multiple Target Tracking. Information Fusion Journal 8, 2–15 (2007)
12. Hoyningen-Huene, N.v., Beetz, M.: Rao-Blackwellized Resampling Particle Filter for Real-Time Player Tracking in Sports. In: VISAPP, INSTICC, pp. 464–471 (2009)
13. Beetz, M., von Hoyningen-Huene, N., Kirchlechner, B., Gedikli, S., Siles, F., Durus, M., Lames, M.: ASPOGAMO: Automated Sports Games Analysis Models. International Journal of Computer Science in Sport 8(1) (2009)
14. Gedikli, S.: Continual and Robust Estimation of Camera Parameters in Broadcasted Sport Games. PhD thesis, TU München (2008)
15. Beetz, M., Gedikli, S., Bandouch, J., Kirchlechner, B., von Hoyningen-Huene, N., Perzylo, A.: Visually Tracking Football Games Based on TV Broadcasts. In: Proc. of Intl. Joint Conf. on Artificial Intelligence, IJCAI (2007)

Dynamic Kernel-Based Progressive Particle Filter for 3D Human Motion Tracking

Shih-Yao Lin[1] and I-Cheng Chang[2]

[1] Graduate of Institute of Networking and Multimedia, National Taiwan University, Taiwan
[2] Dept. of Computer Science and Information Engineering, National Dong Hwa University, Hualein, Taiwan
d97944010@csie.ntu.edu.tw

Abstract. This paper presents a novel tracking algorithm, the dynamic kernel-based progressive particle filter (DKPPF), for markless 3D human body tracking. An articulated human body contains considerable degrees of freedom to be estimated. The proposed algorithm aims to reduce the computational complexity and improve the accuracy. The DKPPF decomposes the high dimensional parameter space into three low dimensional spaces and hierarchically searches the posture coefficients. Moreover, it applies multiple predictions and a mean shift tracker to estimate the human posture iteratively. A dynamic kernel model is proposed to automatically adjust the kernel bandwidth of mean shift trackers according to the probability distribution of the posture states. The kernel model is capable of improving the accuracy of the tracking result. The experimental examples show that the proposed approach can effectively improve the accuracy and expedite the computation.

1 Introduction

Model-based tracking of human bodies has been an active area of research that has received a significant amount of attention recently. The applications of the research include video surveillance, computer game design, and medical analysis. Unfortunately, it is difficult to track the motion of a full body without applying any sensors or devices. The principal difficultly arises from the considerable degrees of freedom of the human motion that have to be estimated, leading to a high computational cost.

A number of methods have been proposed to track the 3D body motion parameters using model-based approaches. Particle filtering [1] is one of the most popular algorithms for dealing with the task, because the particle filter contains multiple predictions and recovers the lost tracks. Because of the high dimensionality of the DOFs of the human body, however, the bottleneck of the particle filter tracking algorithm is the high computational burden. Other research proposed a modified particle filter or used a particle filter with a mathematical training model. Deutcher [2] proposed an anneal particle filter (APF) that uses a multiple searching layer to improve the accuracy and reduce the number of particles. Agarwal [4][5] presented a learning-based approach, which calculated two regression models, the Relevance Vector Machine (RVM) regression and Support Vector Machine (SVM) regression, for recovering the human motion parameters with monocular image sequences. Lin[3] proposed an effective

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 257–266, 2010.

algorithm for 3D model-based human motion tracking, the progressive particle filter. This method 1) decreased the computational cost of the classical particle filter by applying hierarchical searching and 2) improved the accuracy by embedding mean shift trackers into each particle toward high probability locations. Urtasun[6] recovered 3D human poses by using Gaussian Process Dynamical Models (GPDMs), which learn the human pose and motion prior to recovery. Navarantnam[7] employed the HMM to train the temporal coherence of body motion for recovering the pose and finding the smooth trajectory of articulation.

This paper employs the progressive particle filter [3] to reduce the computational cost and improve the accuracy of 3D human motion tracking. The progressive algorithm is based on a particle filter and integrates the hierarchical searching approach and mean shift algorithm. Hierarchical searching is an effective seeking method for reducing the computational loading due to the considerable DOFs of full body motion. Mean shift trackers, which are embedded in each particle, improve the accuracy via an iterative mode seeking process. The progressive particle filter needs only a few particles to predict the joint angle using the hierarchical searching approach, and the mean shift tracker makes each particle shift to its own local maximum to improve the accuracy. We further propose a dynamic kernel model to adaptively adjust the kernel scale of each mean shift tracker. The dynamic kernel model considers the probability in the previous state and dynamically modifies the kernel scale to improve the searching speed of each mean shift procedure. The experimental results show that the proposed approach successfully reduces the iteration time and maintains the accuracy.

The rest of this paper is organized as follows. Section 2 introduces the 3D human model. Section 3 describes the observation likelihood function. Section 4 briefly introduces the classical particle filter and the proposed approach - the dynamic kernel progressive particle filter. Section 5 presents the experimental results. Finally, conclusions are drawn in Section 6.

2 3D Human Model

The proposed 3D human model is shown in Fig. 1. It contains 15 components, including a head, torso, neck, and pairs of upper arms, lower arms, thighs, legs, and feet. Each of these body parts is made of deformable flesh to simulate the target body and improve the precision of the tracking procedure.

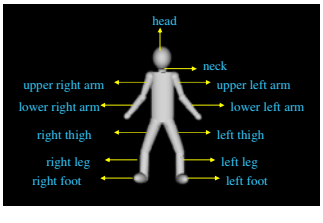


Fig. 1. 3D articulated human body model with 30 DOFs

3 Observation Likelihood Estimation

A single input feature is usually insufficient for tracking the objects. Therefore, the paper combines above four different kinds of features $f = \{f_A, f_E, f_C, f_S\}$ to improve the tracking accuracy. The notation f_A denotes the silhouette map, f_E represents edge distance map, f_C is the contour distance map, and f_S indicates the skin color map. Fig. 2 shows the four different features.

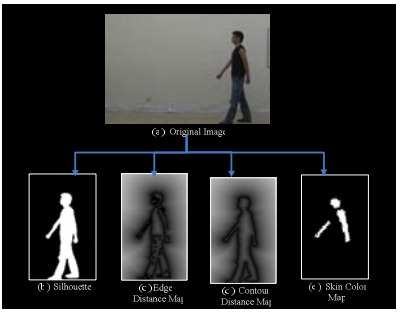


Fig. 2. Feature Extraction: (a) original image O, (b) silhouette map f_A, (c) edge distance map f_E, (c) contour distance map f_C, and (d) skin color map f_S

The observation likelihood function estimates the similarity between the features of the input video f and the predicted pose of the human model $f^M = \{f_A^M, f_E^M, f_C^M, f_S^M\}$. The observation likelihood function is estimated as:

$$\begin{aligned} &p(z \mid x) \propto L(f, f^M) \\ &= \exp(-[\omega_A \cdot M_A(f_A, f_A^M) + \omega_E \cdot M_E(f_E, f_E^M) + \omega_C \cdot M_C(f_C, f_C^M) + \omega_S \cdot M_S(f_S, f_S^M)]) \end{aligned} \tag{1}$$

where $M_A(f_A, f_A^M)$, $M_E(f_E, f_E^M)$, $M_C(f_C, f_C^M)$, and $M_S(f_S, f_S^M)$ are four different similarity measure functions and ω_A, ω_E, ω_C, and ω_S are the weights of each measure function. $M_A(f_A, f_A^M)$ and $M_S(f_S, f_S^M)$ calculate the normalized sum pixels of the absolute difference between silhouette/skin map of the input video and silhouette/skin map of the predicted human model, respectively. The measure functions are shown as follows:

(1) Silhouette similarity measure function

$$M_A(f_A, f_A^M) = \sum_{i=1}^{N_A} abs(f_{A(i)} - f_{A(i)}^M) / N_A \tag{2}$$

where N_A is the number of pixels in f_A.

(2) Skin color similarity measure function

$$M_E(f_E, f_E^M) = \sum_{i=1}^{N_E} dist(f_{E(i)} - f_{E(i)}^M) / N_E \tag{3}$$

where N_s is the number of pixels in f_S.

$M_E(f_E, f_E^M)$ and $M_C(f_C, f_C^M)$ evaluate the distance between the input edge/contour map of the input video and the edge/contour map of the human model. $M_E(f_E, f_E^M)$ and $M_C(f_C, f_C^M)$ are shown as follows:

(3) Edge similarity measure function

$$M_E(f_E, f_E^M) = \sum_{i=1}^{N_E} dist(f_{E(i)} - f_{E(i)}^M) / N_E \tag{4}$$

where N_E is the number of pixels in f_E.

(4) Contour similarity measure function

$$M_C(f_C, f_C^M) = \sum_{i=1}^{N_C} dist(f_{C(i)} - f_{C(i)}^M) / N_C \tag{5}$$

where N_C is the number of pixels in f_C.

4 Tracking Approach

4.1 Particle Filter

The tracking of full body motion can be treated as a Bayesian state estimation problem. Particle filtering [1] is a favorable technique for human motion tracking due to the fact that it provides multiple hypotheses for complex human motion. The posterior Bayesian formulation of the particle filter is defined as:

$$p(x_t \mid Z_t) \propto p(z_t \mid x_t) \cdot p(x_t \mid Z_{t-1}) \tag{6}$$

where x_t denotes the state vector at time t and z_t expresses the observation. The history of observations from 1 to t is indicated as $Z_t = \{z_1, \ldots, z_t\}$. The pdf $p(x_t \mid Z_{t-1})$ is the prediction probability distribution, which is available at time t-1 and can be expressed as:

$$p(x_t \mid Z_{t-1}) = \int p(x_t \mid x_{t-1}) \cdot p(x_{t-1} \mid Z_{t-1}) d_{x_{t-1}} \tag{7}$$

The particle filter provides multiple predictions, which usually apply non-linear and non-Gaussian tracking processes. Unfortunately, the performance of the particle filter usually depends on the number of particles. Because 3D human motion has a large number of degrees of freedom, the tracking procedure demands a large number of particles for estimation. As the number of particles increases, both the accuracy and the computational complexity also increase.

4.2 Progressive Particle Filter

The progressive particle filter [3] tailors a standard particle filter to suit the full body tracking task. The task integrates the techniques of the hierarchical searching approach and iterative mean shift mode seeking. The progressive particle filter decomposes the high dimensional space of full body tracking into several spaces with a lower dimensionality. The tracking procedure tracks each decomposed space and is thus more effective than procedures that require searching the entire high dimensional space. Hierarchical searching can reduce the computational cost because it focuses on each sub-process. The algorithm embeds mean shift trackers into particles in each lower space procedure, and the mean shift trackers apply the iterative mode seeking technique to each particle in order to progress toward nearby high probability locations. After the mean shift mode seeking procedure is completed, each particle converges into each optimal location. In general, most of the particles are gathered in the same target position.

The task contains three principal processes as shown below:

(1) Global motion process:
The system roughly samples the 3D human model's possible location and orientation according to the transition probability $p(x_t^g, x_t^u, x_t^l \mid x_{t-1}^g, x_{t-1}^u, x_{t-1}^l, z_t^g, z_t^u)$. It applies the global mean shift trackers to push each particle until that particle reaches its own nearby local maximum with iterative mean shift searching. The mean shift vector function $M(\cdot)$ can be expressed as:

$$M(\overline{x}_{t-1}) = \frac{\sum_{i=1}^{n_t} K(s_t^{(i)} - \overline{x}_{t-1}) w(s_t^{(i)}) s_t^{(i)}}{\sum_{i=1}^{n_x} K(s_t^{(i)} - \overline{x}_{t-1}) w(s_t^{(i)})} \tag{8}$$

where $s_t^{(i)}$ is the *ith* particle, $\overline{x}_{t-1}$ is the mean position at time *t*, and $w(.)$ is the weighting function.

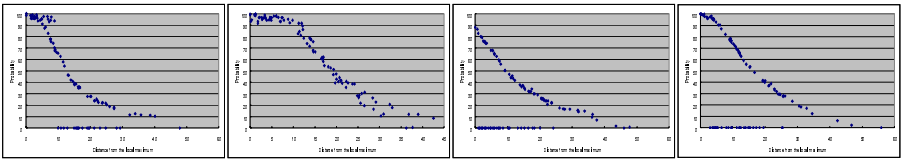
(2) Upper extremity process:
Once full body motion state is determined, the upper extremity process keeps searching for the angles of the upper arms and thighs. The upper extremity particle filter propagates a few possible upper extremity postures with the transition probability $p(x_t^u \mid x_t^g, x_{t-1}^u, x_{t-1}^l)$ and weights each particle by estimating the similarity between the predicted posture and the observed features. The process then applies the upper extremity mean shift trackers to move each particle into the most similar posture.

(3) Lower extremity process:
The lower extremity particle filter re-samples the particles with probability $p(x_t^l \mid x_t^g, x_t^u, x_{t-1}^l)$ for roughly predicting the potential lower extremity pose and weights each particle according to the similarity between the sampled posture and the observed lower extremity. The mean shift trackers move the lower extremity to the most similar posture. This process increases tracking accuracy by shifting each particle to its corresponding mode. Furthermore, it can also reduce the number of particles required during motion tracking. The transition density of the progressive particle filter is defined as:

$$p(x_t \mid x_{t-1}, z_t) \propto p(z_t^g \mid x_t^g)\, p(x_t^g \mid x_{t-1}^g, x_{t-1}^u, x_{t-1}^l)\, p(z_t^u \mid x_t^u, x_t^l)\, p(x_t^u \mid x_t^g, x_{t-1}^u, x_{t-1}^l)\, p(x_t^l \mid x_t^g, x_t^u, x_{t-1}^l) \tag{9}$$

4.3 Dynamic Kernel Model

Fig. 3 provides general motion statistics describing the relationship between the distance from the mean state for each particle and the relatively similar probability. The study finds that the distribution is similar to a Gaussian density distribution. Therefore, if the probability of the particle is high, the re-sampling range should be narrow because the center of the searching range is already approaching the nearest local maximum. On the contrary, if the probability value is low, then the re-sampling range should be large because the current location is far from the local maximum.



(a) (b) (c) (d)

Fig. 3. The distribution between the distance from the local maximum and the probability for each particle in the waving hand sequence: (a) frame 15, (b) frame 30, (c) frame 45, (d) frame 60

We propose a dynamic kernel model (DKM) K_D, which is formulated as:

$$K_D\left(sgm\left(s_{t-1}^{(i)}, x_{t-1}\right), x\right) = \frac{1}{\sqrt{2\pi}\cdot\sigma_t^{(i)}} \exp\left(-\frac{\|x\|}{2\sigma_t^{(i)}}\right),\ i = 1 \sim N \tag{10}$$

where σ_t^i can be expressed by the sigma function $Sgm(.)$.

$$\sigma_t^i = Sgm(s_{t-1}^i, x_{t-1}) = \left(\frac{s_{t-1}^{(i)} - x_{t-1}}{\sigma_{t-1}^{(i)}}\right).\hat{\sigma} \tag{11}$$

We investigated the training data and applied the dynamic variance function (12) to estimate the standard sigma value $\hat{\sigma}$ of the training data.

$$\hat{\sigma} = \sum_{m=1}^{M} \sqrt{\left(\frac{\sum (s_t^{(i)} - x_t)^2}{N * M}\right)},\ i = 1 \sim N \tag{12}$$

where M is the number of training data; M is equal to 100 in our experiment. The notation x_t refers to the mean value in state t, and N is the number of particles in each state. In this case, the sigma value is between 15 and 25 as shown in Fig. 4. Hence, the system employs the average sigma value of 17.3975 as $\hat{\sigma}$ in our system.

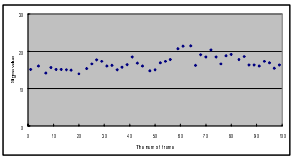


Fig. 4. The relationship between the distance from the local maximum and the state sequence

4.4 Dynamic Kernel Based Progressive Particle Filter

The iteration time for the progressive particle filter depends on the kernel bandwidth h of the mean shift trackers. Thus, it may require many iterations to shift each particle toward each nearby mode when the kernel bandwidth is too narrow and the particle far from the highest probability position. In addition, this method expends redundant computational load when the kernel bandwidth is too wide and the particle close to the mode. The reason for this lies in the fact that the mean shift tracker still needs to propagate redundant particles to predict the low probability positions and calculate their probabilities.

The dynamic kernel-based progressive particle filter (DKPF) can embedded the DKM into each mean shift tracker of the progressive particle filter to dynamically adjust the scale of the mean shift kernel bandwidth by the previous-state probability of each particle.

When the particle is located at a low probability position in the DKFP, the mean shift tracker expands the kernel bandwidth with the DKM. Thus, the modified mean shift tracker shifts the particle toward the higher probability space quickly. If the particle is already located at a high probability position, the particle only needs to move a short distance to reach the mode. The algorithm applies the DKM with the probability of the particle in the previous iteration time to narrow the bandwidth of mean shift tracker and thus accelerate the convergence speed of the mean shift procedure. A diagram of the DKPPF is shown below:

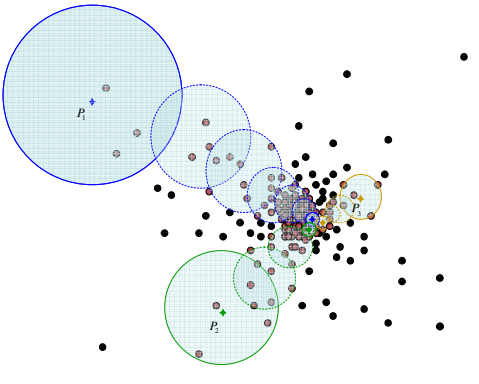


Fig. 5. The mean shift procedure of the DKPPF

with the particles P_1, P_2, and P_3 with mean shift kernel bandwidths h_1, h_2, and h_3. The kernel bandwidth lengths of these mean shift trackers are distributed as $h_1 > h_2 > h_3$ because the probability order of these particles follows the rule $P_1 < P_2 < P_3$. As shown in Fig.5, the mean shift trackers dynamically adjust the kernel bandwidth by the DKM with the probability of the particle. Particle P_1 is far from the mode, so the kernel scale range needs to be enlarged to shift P_1 from the low probability space toward a high density position using the mean shift procedure. The kernel bandwidth of P_3 should adjust to a lesser degree because P_3 is located at a high density position close to the mode. The small kernel bandwidth can expedite the searching time necessary in the mean shift process as it identifies the relevant ranges.

The DKPPF is trained in advance using video data to obtain the trained sigma value $\hat{\sigma}$. The kernel estimation function of the DKPPF can be rewritten as:

$$\begin{aligned}\hat{f}(\bar{x}) &= \hat{p}(x_t^{(i)} \mid Z_t, \{s_{t-1}^{(j)}\}_{j=1}^{n}) = \frac{1}{nh_t^{(i)}} \sum_{j=1}^{n} K_D\left(\frac{s_t^{(j)} - x_t^{(n)}}{h_t^{(i)}}\right) w_t^{(i)} \\ &= \frac{1}{n \cdot sgm(s_{t-1}^{(j)}, x_{t-1}^{(i)})} \sum_{j=1}^{n} K_D\left(\frac{s_t^{(j)} - x_t^{(i)}}{sgm(s_{t-1}^{(j)}, x_{t-1}^{(i)})}\right)\end{aligned} \tag{13}$$

where i is the particle number and t denotes each mean shift procedure iteration. and the term $sgm(\cdot)$ can be expressed as:

$$Sgm(\{s_{t-1}^{(j)}\}_{j=1}^{n}, x_t^{(i)}) = \left(\frac{\sum_{j=1}^{n} s_{t-1}^{(j)} - x_{t-1}}{h_{t-1}^{(i)}}\right) \cdot \hat{\sigma} \tag{14}$$

The mean shift vector for each particle can be calculated as:

$$M_r(x_{t-1}^{(i)} = s_{t-1}^{(i)}) = \frac{\sum_{i=1}^{n} K_D(x_t^{(i)} - s_t^{(j)}) w(s_t^{(j)}) s_t^{(j)}}{\sum_{i=1}^{n} K_D(x_t^{(i)} - s_t^{(j)}) w(s_t^{(j)})} \tag{15}$$

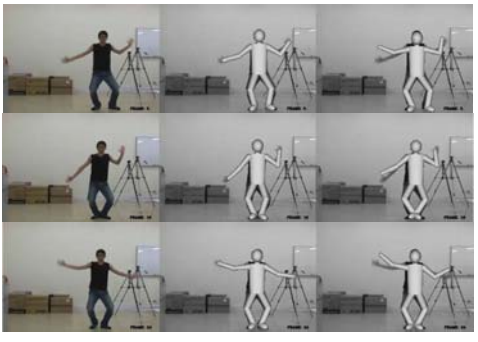
The DKPPF not only performs multiple predictions to overcome the non-linear and non-Gaussian motion of the complex human action but also uses a hierarchical searching strategy to decompose the high-DOF space into a low-DOF space and thereby decrease the particle number and reduce the computational cost. The embedded DKM further reduces the iterative time of each tracker to improve searching efficiency.

5 Experimental Results

Two experiments were performed to evaluate the performance of the proposed algorithm. The first experiment presents the crab step, which demonstrates the horizontal motion of the human body. The experiment compares the tracking results between the standard particle filter, progressive particle filter, and DKPPF. In this experiment, the standard particle filter used the hierarchical searching technique with 100 particles in

each body part. The DKPPF also used this number of particles. The results demonstrate that the DKPPF is more accurate than the standard particle filter. In this experiment, the training sigma was 18.1.

The experiment uses a different number of particles for each approach and compares the accuracy and performance of the algorithms. Table 1 demonstrates that both the DKPPF and PPF obtain better accuracy results than the standard particle filter. The DKPPF requires fewer iterations than the PPF.



(a) (b) (c)

Fig. 6. Crab step experimental results: (a) original video sequence, (b) tracking result with the DKPF, and (c) standard particle filter tracking results

Table 1. The Comparison between the standard particle filter (PF), progressive particle filter (PPF), and dynamic kernel-based particle filter (DKPPF)

	Number of Particles	Average Error (degree)	Average Iteration
PF	25	13.39	
	50	10.91	
	100	10.61	
	200	7.14	
PPF	10	8.54	8.11
	15	7.60	6.67
	20	7.61	5.89
	25	7.46	6.11
DKPPF	10	7.86	9.14
	15	6.95	6.21
	20	6.68	4.05
	25	7.34	5.11

The second experiment presents the tracking of a front jumping motion (Fig. 7). The aim was to track the global vertical motion. The skin color of the face region offers one precise method by which to track the vertical motion. This experiment also shows that the proposed approach can obtain exact tracking results even when the motion of the target occurs very rapidly.

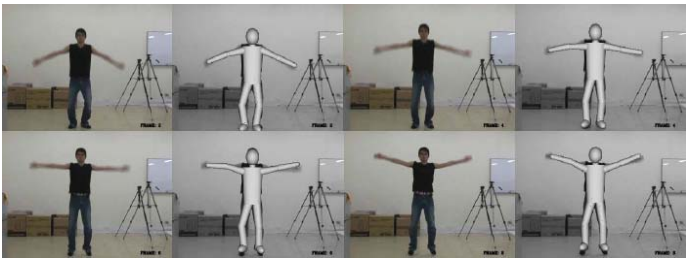


Fig. 7. Front jump tracking results

6 Conclusion

This paper proposes an effective algorithm, the kernel-based progressive particle filter, for 3D human motion tracking. The progressive particle filter integrates the hierarchical searching method to reduce the computational cost as well as embedded mean shift trackers on each particle to improve the accuracy. The paper improves the progressive particle filter with the dynamic kernel model to reduce the number of iterations for each mean shift tracker. The dynamic kernel model adjusts the scale of the mean shift tracker bandwidth to effectively improve the mode-seeking procedure. Experimental results show that the progressive particle filter is more accurate than the traditional particle filter despite a reduction in the number of particles used by the particle filter. In summary, the kernel-based progressive particle filter improves upon the progressive particle filter by reducing number of iterations necessary in the mean shift-searching procedure.

References

1. Isard, M., Blake, A.: CONDENSATION-Conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–98 (1998)
2. Deutscher, J., Blake, A., Reid, L.: Articulated Body Motion Capture by Annealed Particle Filtering. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 126–133 (2000)
3. Lin, S.-Y., Chang, I.-C.: 3D Human Motion Tracking Using Progressive Particle Filter. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part II. LNCS, vol. 5359, pp. 833–842. Springer, Heidelberg (2008)
4. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. IEEE Transaction on Patten Analysis and Machine Intelligence 28(1), 44–45 (2006)
5. Agarwal, A., Triggs, B.: Learning to track 3D human motion from silhouettes. In: ACM International Conference Proceeding Series Proceedings of the twenty-first international conference on Machine learning, vol. 69 (2004)
6. Urtasun, R., Fleet, D.J., Fua, P.: 3D People Tracking with Gaussian Process Dynamic Models. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 238–245 (2006)
7. Navarantnam, R., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Hierarchical Part-Based Human Body Pose Estimation. In: Proc. British Machine Vision Conf., vol. 1, pp. 479–488 (2005)

Bayesian 3D Human Body Pose Tracking from Depth Image Sequences

Youding Zhu and Kikuo Fujimura

Honda Research Institute USA,
Mountain View, California
zhu.81@osu.edu, kfujimura@hra.com

Abstract. This paper addresses the problem of accurate and robust tracking of 3D human body pose from depth image sequences. Recovering the large number of degrees of freedom in human body movements from depth image sequence is challenging due to the need to resolve depth ambiguity caused by self-occlusions and difficulty to recover from tracking failure. Human body poses could be estimated with a high accuracy based on local optimization using dense correspondences between 3D depth data and the vertices in an articulated human model. However, it cannot recover from tracking failure. This paper presents a method to reconstruct human pose by detecting and tracking human body anatomical landmarks (key-points) from depth images. The proposed method is robust and recovers from tracking failure when a body part is re-detected. However, its pose estimation accuracy depends solely on image-based localization accuracy of key-points. To address these limitations, we present a flexible Bayesian method for integrating pose estimation results obtained by methods based on key-points and local optimization. Experimental results are shown and performance comparison is presented to demonstrate the effectiveness of the proposed method.

Keywords: Depth image, dense correspondences, key-point detection, constrained inverse kinematics, robust 3D human pose tracking.

1 Introduction

For the past decades, 3D human body pose tracking from video inputs has been an active research field motivated by various applications including human computer interaction. The major challenges of recovering the large number of degrees of freedom in human body movements from image sequence are the difficulties to resolve the various ambiguities in the projection of human motion onto the image plane and the diversity of visual appearance caused by clothing and varying illumination.

The recent introduced time-of-flight (TOF) based imaging devices have captured the attention of researchers due to the potential to resolve depth ambiguity [1,2,3,4]. 3D pose tracking usually is difficult to obtain robustly by using a single optical camera. In particular, methods based on silhouette information often fail to track 3D poses where there are self-occlusions. Although non-silhouette

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 267–278, 2010.

based methods [5,6] have been proposed to track poses with self-occluded limbs, their robustness depends much on illumination conditions, body texture, and perhaps extensive training in case of learning based methods. Depth data, as in Figure 1, provides a valuable cue in resolving the depth ambiguity problem. Other advantages of TOF cameras include their portability, relatively good depth resolution comparing with stereo cameras.

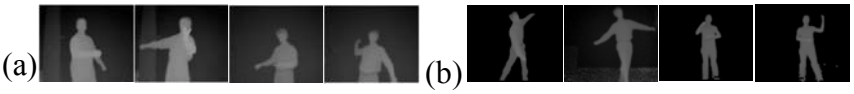


Fig. 1. Depth data (a) Example upper body posture that are to be tracked;(b) Example whole body posture that are to be tracked

Most existing approaches to track human body pose from depth video sequence [1,2,3,4] are related to the Iterative Closest Point (ICP) approach [7]. These approaches are able to track the human body pose with a high accuracy because dense correspondences are used for pose optimization. However, these approaches based on local optimization are vulnerable to tracking failure when body parts get close to each other and cannot recover from tracking failure afterwards. Knoop et al [2] show that they can achieve more accurate pose tracking by integrating hand/face tracking. However, it becomes a challenging task to have a 2D hand/face tracker that works well for various complicated motion, and they do not elaborate on how the robustness of 2D feature tracker could affect their 3D pose estimation. Zhu et al [4] use coarse body identification to reduce the ambiguity during dense correspondences search. However, it has difficulties to detect arms when they re-appear.

Recovering from pose tracking failure is indeed an important component for a robust pose tracking algorithm. Considering example postures shown in Figure 1, on the one hand, a visible arm could get close to the torso so that depth resolution is not high enough to detect the arm. Also it is possible that a visible limb could be occluded temporarily by another limb. On the other hand, a missing limb can reappear later. A robust tracking algorithm must deal with intermittent occlusions to prevent tracking failures.

For many existing pose tracking methods, tracking long sequences will result in tracking failure which cannot be easily recovered. This paper presents a key-point based method to reconstruct poses from anatomical landmarks detected and tracked from depth image analysis. Key-point based method is robust and can recover from tracking failure when a body part is re-detected and tracked. However, its pose estimation accuracy depends solely on the image-based localization accuracy of key-points. To address these limitations, we present a Bayesian method to integrate pose estimation results from methods using local optimization and key-point detection.

The rest of the paper is organized as follows. Section 2 depicts the human model used in this paper, and the background on pose estimation with constrained inverse kinematics. Our Bayesian method for accurate and robust pose tracking is presented in Section 3. Methods using key-points and local optimization are described in Section 3.1 and 3.2, respectively. Experimental results are shown in Section 4; and Section 5 concludes the paper.

2 Human Body Model and Pose Estimation with Constraint Inverse Kinematics

The human body model is represented as a hierarchy of joint link models with a skin mesh attached to it as in Lewis et al [8]. The human model in Figure 2(a) includes 28 dofs for whole body, and 20 dofs for upper body. During pose estimation, one of natural constraints is to enforce joint limits. For example, by enforcing elbow joint limits, we could avoid generating the backward bending arms as in the Figure 2(b).

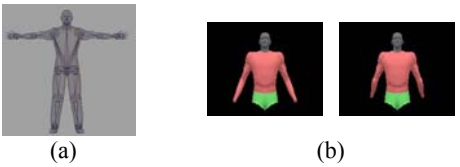


Fig. 2. Human body model (a) Hierarchical joint link model with 28 dofs; (b) Elbow joint limit constraints for natural pose tracking

Fig. 3. Model marker points (a) from key-point detection; (b) from dense ICP correspondences (each yellow vector represents a correspondence pair)

Let q_0 be the initial model pose, V be the set of model marker points, P be the set of observed points from the sensor. Let $\hat{q} = ConstaintIK(q_0, V, P)$ denote the constrained inverse kinematics as:

$$\hat{q} = q_0 + sJ^*(P - V) \tag{1}$$

$$J^* = W_1^{-1} J^T (JW_1^{-1} J^T + W_2)^{-1} \tag{2}$$

where s is a scalar to adjust the step size of inverse kinematics, W_1 and W_2 are defined as for singularity avoidance and joint limit avoidance. In this study, we use the joint limit avoidance method described by Zhu et al [9].

The model marker points include the set of model vertices as shown in Figure 3. In Figure 3(a), the model marker points are located at the human anatomical landmarks, and observed points are detected through low-level depth image analysis as in Section 3.1. On the contrary, in Figure 3(b), model marker points are sampled randomly from the model vertices, and observed points are found during the ICP correspondence searching as in Section 3.2.

3 Robust 3D Pose Tracking with Bayesian Method

Let q_t be the model pose parameters, inlcuding all of degrees of freedom of human model, at time t, and $p(q_t|I_1, I_2, \cdots, I_t)$ be the probability distribution of pose parameters given all observed images $\{I_1, I_2, \cdots, I_t\}$, then Bayesian tracking is formulated as:

$$\begin{aligned} p(q_t|I_1, I_2, \cdots, I_t) &\propto p(I_t|q_t)p(q_t|I_1, I_2, \cdots, I_{t-1}) \\ &= p(I_t|q_t) \int_{q_{t-1}} p(q_t|x_{t-1})p(q_{t-1}|I_1, I_2, \cdots, I_{t-1})dq_{t-1} \end{aligned} \tag{3}$$

Assuming that we can approximate the observation distribution as:

$$p(I_t|q_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \tag{4}$$

Let human dynamics have Gaussian noise $N(0, W)$, the temporal propagation is given by:

$$p(q_t|I_1, I_2, \cdots, I_{t-1}) = \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; f(\mu_j^{t-1}), \Lambda_j^{t-1} + W) \tag{5}$$

where $f(\mu_j^{t-1})$ is any appropriate pose dynamic process.

Using the above Bayesian tracking equation, we can represent the posterior as:

$$p(q_t|I_1, I_2, \cdots, I_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; f(\mu_j^{t-1}), \Lambda_j^{t-1} + W) \tag{6}$$

As we can see, this will increase the Gaussian components for the posterior distribution exponentially along the updating of time. Instead, we approximate this with M component Gaussian distribution:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{j=1}^{M} \pi_j^t N(q_t; \hat{\mu}_j^t, \hat{\Lambda}_j^t) \tag{7}$$

Since we represent the posterior distribution as a sum of Gaussian, there are available methods to perform density approximation. One simple way is that

we keep the dominant modes in the posterior distribution. Researchers [5,10] also suggest to pick modes from likelihood function and combine them with compatible ones from the predicted priors. Some authors [11] also pick the modes from likelihood function and re-weight with predicted prior.

The detailed illustration of this Bayesian inference method to pose tracking is shown in Figure 4, where we are able to integrate three sources of information: key-point detection from low-level image analysis, local pose optimization with ICP, and temporal prediction information if that is available. We describe these components in the following sections.

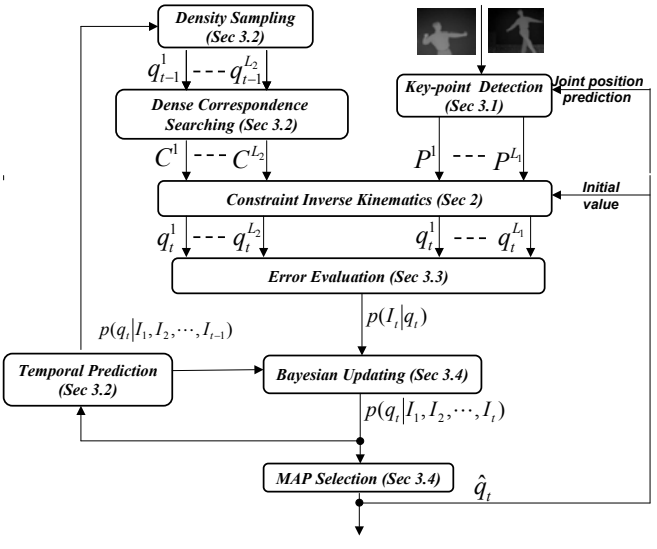


Fig. 4. Robust pose estimation with Bayesian tracking framework

3.1 Key-Point Detection from Depth Image Sequence for Pose Tracking

In order to have a robust pose tracker, one of the crucial processing steps is to localize each visible limb. We present a method to detect, label and track body parts using depth images as shown in Figure 5. To detect major body parts such as the head, torso, and waist, we make use of a deformable template which we refer to as the HNT template which consists of a head, neck, and trunk. The trunk is further decomposed into a torso and waist. They are represented by a circle, trapezoid, rectangle, and another trapezoid, respectively as in Figure 5, 6 shown in red. To localize the HNT template, our algorithm takes a background-subtracted depth image I as input and deform the HNT template to produce the optimal template configuration by minimizing the discrepancy between the deformed HNT template and the background-subtracted depth image.

Once the head, neck, and trunk are detected, limbs (two arms and two legs) are to be detected as shown in Figure 6. For example, we can detect a upper

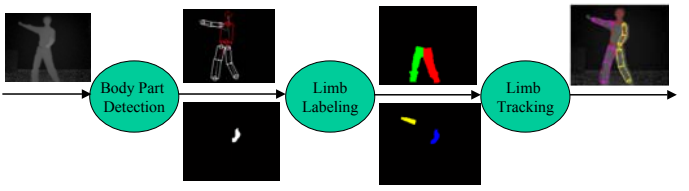


Fig. 5. Body part detection, labeling and tracking

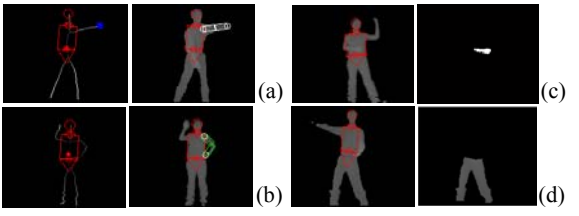


Fig. 6. HNT template localization (shown in red) and limb detection: (a) Open arm detection; (b) Looped arm detection; (c) Arm detection that is in front of torso; (d) Lower limb detection

body limb that is open, or that forms a loop, or that is in front of torso based on depth image analysis. We can detect lower limbs by finding all pixels that are lower than the waist.

After the limbs are detected, we perform a labeling step in order to differentiate the left and right limbs as well as to determine the limb occlusion status. We use the following steps to label detected arms (same steps applied to leg labeling) based on the arm occlusion status at the last frame. When both arms are visible from the last frame, let us define H_{LA} and H_{RA} to be the histogram of depth values for the left and right arms respectively, and we label the current detected limb pixels as left or right arm based on its geometric and appearance distance to the tracked arms.

$$P(L_x^t = \mathrm{LA} | X_{\mathrm{LA}}^t, H_{\mathrm{LA}}^t, X_{\mathrm{RA}}^t, H_{\mathrm{RA}}^t) = \frac{e^{-\gamma d_{\mathrm{LA}}(x)} H_{\mathrm{LA}}(I_x)}{e^{-\gamma d_{\mathrm{LA}}(x)} H_{\mathrm{LA}}(I_x) + e^{-\gamma d_{\mathrm{RA}}(x)} H_{\mathrm{RA}}(I_x)} \tag{8}$$

where $d_{\mathrm{LA}}(x)$ is the distance from pixel x to the left arm:

$$d_{\mathrm{LA}(x)} = \begin{cases} 0 & \text{if } x \text{ is inside left arm} \\ d(x, \mathrm{LA}) & \text{otherwise} \end{cases} \tag{9}$$

where $d(x, \mathrm{LA})$ is the minimal distance from x to edges of the left arm. $d_{\mathrm{RA}}(x)$ is defined similarly. In short, a pixel x has a high probability of belonging to LA, if x is sufficiently close to where LA was in the previous frame. While two arms are overlapping in the image, x has a high probability of belonging to LA if it has a depth value that is close to depth values represented by the left arm in the previous frame.

When only one arm is visible from the last frame, we compute the geomretic distance from the detected arm pixels to the tracked arm, and decide the label based on the maximal arm movement distance between successive frames. When both arms are not visible from the last frame, we label the detected arm based on its spatial distribution relative to the torso center line, where the left arm is located to the left of torso center line.

Finally, when the observed number of pixels for a limb is less than the threshold, we declare that the limb is occluded. For each visible limb, we preform a local optimizaiton to align the 2-D scaled prismatic model [12] to the detected limbs.

The key-points corresponding to the human anatomical landmarks as in Figure 3(a) are extacted from the deformed HNT template and 2-D scaled prismatic model. They are further used to generate 3D pose hypotheses based on constraint inverse kinematics. At any frame, we might only detect a subset of landmarks because of occlusion. Moreover, it is difficult to accurately localize elbow points for certain poses, and we can only obtain approximate elbow positions. It is known that methods based on inverse kinematics depend on starting pose values as well. Let $\hat{q}_{t-1}$ be the optimal pose estimation of the last frame and let q^0_{t-1} be the resting pose. We use the constrained inverse kinematics to generate three sets of joint hypotheses ($L_1 = 3$). q^1_t generates the pose for both the optimal estimation $\hat{q}_{t-1}$ and all feature points. However, q^2_t generates pose from the resting pose q^0_{t-1} and all feature points so that these hypotheses keep our estimation robust against possibly erroneous estimations from the last frame. q^3_t generates pose from the optimal estimation $\hat{q}_{t-1}$ without using elbow feature points so that it is robust against errors in elbow detection.

3.2 Temporal Prediction, Density Sampling and Dense Correspondence Searching for Pose Tracking

Since the motion to be tracked in this study is general and has high uncertainty, a common approach is to model the human pose temporal dynamics as zero velocity with a Gaussian noise $N(0, W)$. Therefore, we can approximate the temporal prediction prior as:

$$p(q_t|I_1, I_2, \cdots, I_{t-1}) = \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \tag{10}$$

Density sampling can be performed based on this temporal prediction prior distribution as this is a standard Gaussian mixture distribution.

Let q^i_{t-1} be one of samples from density sampling, Vs denote a set of sampled model vertices that is visible from camera, Cs denote the set of 3D depth points that is closest to Vs (as shown in Figure 3(b)), and q^i_t denote the pose from local pose optimization:

$$q^i_t = ConstraintIK(q^i_{t-1}, Vs, Cs) \tag{11}$$

We obtain visible model vertices Vs from the depth buffer technique of OpenGL rendering. Closest point set Cs is obtained through its grid acceleration data structure.

3.3 Tracking Error Evaluation

To evaluate the tracking quality, we use a tracking error measurement function that is based on the sum of the distances from sampled depth points to their corresponding closest model vertices. Without loss of generality, let us use Ps to denote the set of sampled depth points and Vs the set of visible model vertices that are closest to the Ps. Then, our tracking error measurement function can be defined as:

$$d^2(Ps, Vs(q_t)) = \sum_j \|Ps_j - Vs_j(q_t)\|^2 \tag{12}$$

With this tracking error measurement function, we can approximate the observation distribution as:

$$p(I_t|q_t) \propto exp\{-d^2(Ps, Vs(q_t))\} \tag{13}$$

We can further approximate the observation distribution by keeping only a few modes from the local optimization and constrained inverse kinematics on key-points. Let $\{\mu_k^t, k = 1, \cdots, k = K\}$ denote the set of modes, we can approximate the observation distribution as:

$$p(I_t|q_t) \approx \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \tag{14}$$

where, w_k^t can be estimated as:

$$\tilde{w}_k^t \approx exp\{-d^2(Ps, Vs(\mu_k^t))\}$$

$$w_k^t = \frac{\tilde{w}_k^t}{\sum_{k=1}^{K} \tilde{w}_k^t} \tag{15}$$

Λ_k^t can be estimated as:

$$\Lambda_k^t \approx (J_{Vs}^T J_{Vs})^{-1} \tag{16}$$

3.4 Bayesian Updating and MAP Selection

Given observation distribution $p(I_t|q_t)$ as Equation 14, and temporal prediction prior $p(q_t|I_1, I_2, \cdots, I_{t-1})$ as Equation 10, we obtain the posterior distribution as:

$$p(q_t|I_1, I_2, \cdots, I_t) = \sum_{k=1}^{K} w_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \tag{17}$$

In order to avoid the exponential increase of Gaussian components, without loss of generality, we first approximate it by the first M dominant observation modes as:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{k=1}^{M} \hat{w}_k^t N(q_t; \mu_k^t, \Lambda_k^t) \sum_{j=1}^{M} \pi_j^{t-1} N(q_t; \mu_j^{t-1}, \Lambda_j^{t-1} + W) \quad (18)$$

and then re-weight them with temporal prior:

$$p(q_t|I_1, I_2, \cdots, I_t) \approx \sum_{j=1}^{M} \pi_j^t N(q_t; \mu_j^t, \Lambda_j^t) \quad (19)$$

where weights π_j^t can be estimated as:

$$\tilde{w}_j^t = \hat{w}_k^t \sum_{j=1}^{M} \pi_j^{t-1} N(\mu_k^t; \mu_j^{t-1}, \Lambda_j^{t-1} + W)$$

$$\pi_j^t = \frac{\tilde{w}_j^t}{\sum_{j=1}^{M} \tilde{w}_j^t} \quad (20)$$

At any frame, the optimal pose estimation is exported as the mode in the posterior distribution $p(q_t|I_1, I_2, \cdots, I_t)$.

4 Experiments

The Bayesian pose tracking algorithm is implemented and tested on a set of depth image sequences captured from a single time-of-flight (TOF) range sensor [13]. The current implementation works well for body twists up to 40 degree rotation on either side of a front facing posture. Large twists and severe interaction between upper and lower body limbs remain as a challenge in the current implementation. Upper-body and whole-body tracking results are shown in Figures 7, 8, 9, and 10.

We summarize and compare its performance with ICP method and key-point based method as in Table 1. The ICP method utilizes general correspondences to estimate the pose, which does not require detection and tracking of key-points. Nevertheless, the ICP method could result in tracking failure for transient occlusions, and is not be able to recover from it. Furthermore, the ICP method could not be integrated with other information flexibly. The key-point based method is able to track through transient occlusion, and recover from tracking failures when the body parts are detected again. But, it is not able to take advantage of other information either. As seen, the Bayesian-based method is able to take advantage of both ICP and key-point based methods. It is able to track through transient occlusions, recover from tracking failure whenever body parts are detected again, and update the pose by performing local optimization

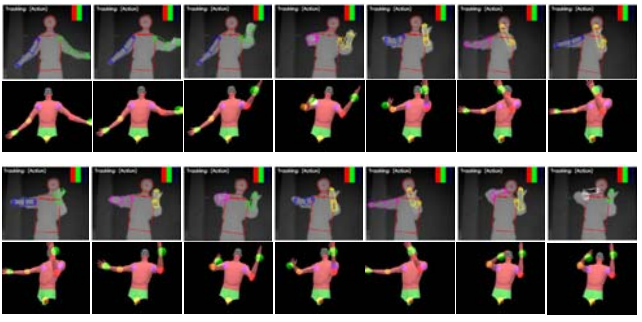


Fig. 7. Upper body pose tracking for violin playing motion. Rows 1 and 3: depth image sequence with the detected body parts. Rows 2 and 4: corresponding reconstructed pose.

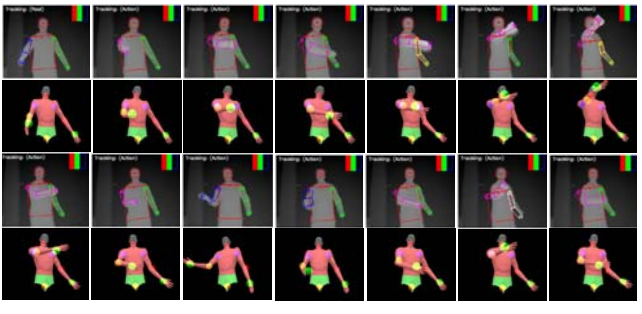


Fig. 8. Upper body pose tracking for frisbee throwing motion. Rows 1 and 3: depth image sequence with the detected body parts. Rows 2 and 4: corresponding reconstructed pose.

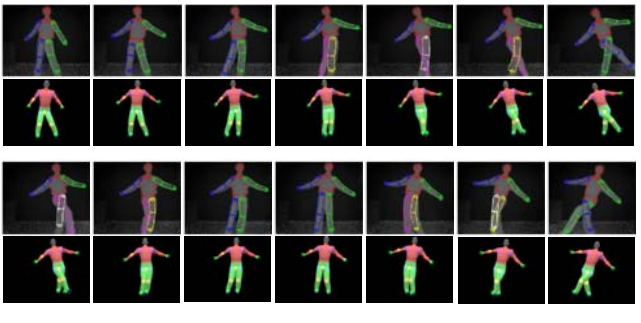


Fig. 9. Whole body pose tracking with self occlusions during leg crossing. Rows 1 and 3: depth image sequence with the detected body parts. Rows 2 and 4: corresponding reconstructed pose.

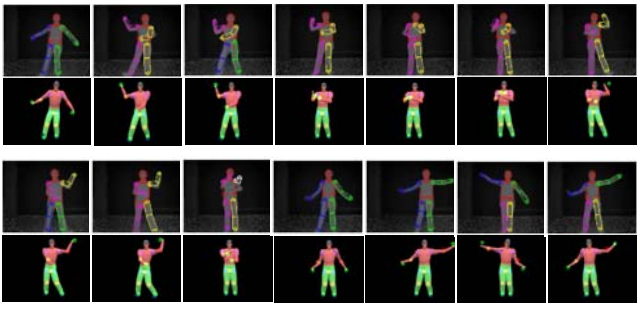


Fig. 10. Whole body pose tracking during a dancing sequence. Rows 1 and 3: depth image sequence with the detected body parts. Rows 2 and 4: corresponding reconstructed pose.

Table 1. Comparison between various human pose tracking approaches

Methods	Tracking through occlusion	Error-recovery	Tracking with missing key-points	Integration with other information	Speed
ICP-based	No	No	Yes	No	5∼9Hz
Key-point-based	Yes	Yes	No	No	3∼6Hz
Bayesian-based	Yes	Yes	Yes	Yes	0.1Hz

Table 2. A comparison of overall trajectory accuracy between key-point based method and Bayesian-based method

Methods	X trajectory accuracy	Y trajectory accuracy	Z trajectory accuracy
Key-point-based	80mm	84mm	93mm
Bayesian-based	73mm	78mm	87mm

without key-points. The Bayesian-based method has the potential to make use of other information flexibly whenever available, for example, pose prediction from machine learning approaches. Furthermore, the Bayesian-based method could achieve a higher accuracy for joint trajectories than key-point based methods because it could take advantage of ICP to refine the alignment between 3D model and point clouds, as shown in Table 2.

5 Conclusion

We have presented a Bayesian method to integrate pose estimation results from methods using key-points and local optimization. This demonstrates a potential approach to integrate pose estimation results from different modalities to improve the robustness and accuracy. The computational burden of the Bayesian-based methods is a major concern for interactive applications with our current implementation.

References

1. Grest, D., Woetzel, J., Koch, R.: Nonlinear body pose estimation from depth images. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 285–292. Springer, Heidelberg (2005)
2. Knoop, S., Vacek, S., Dillmann, R.: Sensor fusion for 3d human body tracking with an articulated 3d body model. In: ICRA, pp. 1686–1691 (2006)
3. Ziegler, J., Nickel, K., Stiefelhagen, R.: Tracking of the articulated upper body on multi-view stereo image sequences. CVPR 1, 774–781 (2006)
4. Zhu, Y., Fujimura, K.: Constrained optimization for human pose estimation from depth sequences. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 408–418. Springer, Heidelberg (2007)
5. Sminchisescu, C., Triggs, B.: Kinematic jump processes for monocular 3d human tracking. CVPR 1, 18–20 (2003)
6. Lee, M.W., Cohen, I.: Proposal maps driven mcmc for estimating human body pose in static images. CVPR 2, 334–341 (2004)
7. Besl, P., McKay, N.: A method for registration of 3-d shapes. PAMI 14(2), 239–256 (1992)
8. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformations: A unified approach to shape interpolation and skeleton-driven deoformation. In: SIGGRAPH, pp. 165–172 (2000)
9. Zhu, Y., Dariush, B., Fujimura, K.: Controlled human pose estimation from depth image streams. In: CVPR time-of-flight workshop (2008)
10. Cham, T.J., Rehg, J.: A multiple hypothesis approach to figure tracking. CVPR 2, 239–245 (1999)
11. Demirdjian, D., Taycher, L., Shakhnarovich, G., Grauman, K., Darrell, T.: Avoiding the 'streetlight effect': Tracking by exploring likelihood modes. ICCV 1, 357–364 (2005)
12. Morris, D., Rehg, J.: Singularity analysis for articulated object tracking. In: CVPR, pp. 189–196 (1998)
13. SwissRanger: online time-of-flight camera information from, http://www.mesa-imaging.ch/prodviews.php

Crowd Flow Characterization with Optimal Control Theory

Pierre Allain[1], Nicolas Courty[1], and Thomas Corpetti[2]

[1] Université Européenne de Bretagne – VALORIA / UBS
{Pierre.Allain,Nicolas.Courty}@univ-ubs.fr
[2] COSTEL, Rennes (France) / LIAMA, Beijing (China)
Thomas.Corpetti@irisa.fr

Abstract. Analyzing the crowd dynamics from video sequences is an open challenge in computer vision. Under a high crowd density assumption, we characterize the dynamics of the crowd flow by two related information: velocity and a disturbance potential which accounts for several elements likely to disturb the flow (the density of pedestrians, their interactions with the flow and the environment). The aim of this paper to simultaneously estimate from a sequence of crowded images those two quantities. While the velocity of the flow can be observed directly from the images with traditional techniques, this disturbance potential is far more trickier to estimate. We propose here to couple, through optimal control theory, a dynamical crowd evolution model with observations from the image sequence in order to estimate at the same time those two quantities from a video sequence. For this purpose, we derive a new and original continuum formulation of the crowd dynamics which appears to be well adapted to dense crowd video sequences. We demonstrate the efficiency of our approach on both synthetic and real crowd videos.

1 Introduction

Analyzing crowd video sequences has recently revealed to open specific and original problems in computer vision. Direct applications consider the design of safety systems for public confined or opened spaces. In this case, the goal of a surveillance system is to be able to give an information of the flow of persons at a given time in a given situation. From this information, one can infer useful statistics about dangerous areas such as bottlenecks or narrow passages. Automatic surveillance system can also trigger alarms whenever abnormal or dangerous situations are detected. It is also noticeable that such tools participate to our comprehension of crowd phenomena. As an example, Helbing and colleagues [1] have recently build a new theory on crowd dynamics based on an analysis of video recordings of the annual pilgrimage in Makkah. Such a study of the crowd behavior from real data enables to hold out critical locations (*i.e.* areas with high density and pressure) of a scene and can in addition contributes to the elaboration of accurate simulation models. Another original recent application is related to computer graphics and the production of digital effects: Courty and Corpetti [2] designed a data-driven crowd animation system based on the velocity fields acquired with optical flow techniques from an input crowd video sequence.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 279–290, 2010.

The coupling of crowd dynamics and real data exhibits very promising results and has opened a rich area of research. This paper is a contribution in that direction. We argue that the apparent motion information is intrinsically insufficient to characterize the dynamics of the flow since the lack of motion in the image can be interpreted as a null density or a large congestion area where people are likely to be injured. We define a substantially complete crowd flow analysis as the extraction from the sequence of *i)* time-consistent motion fields and *ii)* an associated disturbance potential. The motion field is a rich dynamical descriptor of the flow which can be related to the velocity of flow. The disturbance potential accounts for several physical quantities such as the density or the pressure in the flow. This information is crucial to extract sensible and potentially dangerous areas. Although an important number of approaches are available to measure the apparent velocity field from images sequences in various situations, the estimation of the disturbance potential is a critical problem and is still an open domain of research. This component is indeed tricky to observe directly from images. It is nevertheless intuitive that this potential influences the motion field: in a natural way, human beings tend to avoid over-concentrated or high-pressure areas, and their velocities are directly influenced by the surrounding person concentration.

The original contribution of this paper is to use recipes from optimal control theory [3] and variational assimilation [4], originally used in the context of meteorology, to define a new tool for the characterization of the crowd flow. Such techniques enable to estimate a (potentially high dimensional) system state driven with a dynamic model known up to some noise. A key advantage relies on the ability to measure unobserved parameters that control the dynamic model. As such, it is thoroughly adapted to the problem we are dealing with. The definition of a system based on variational assimilation especially requires *i)* a dynamic model related to the motion field and the disturbance potential and *ii)* an observation operator that links our data (images) to some components of system state (motion fields). Among others, we propose in this paper a new and crude physical model for crowd dynamics and apply it to estimate time-consistent informations of image sequences of human crowds.

The remainder of the paper is organized as follows: after the presentation of related work in the context of crowd flow analysis from video, we give an overview of our method (Section 3.1). Section 3.2 presents our modeling of the problem and implementation issues. Before concluding, Section 4 exposes our experimentations on both synthetic (with the associated ground truth) and real crowd sequences.

2 Related Work

Analysis of crowd video sequences are generally focused toward two distinct problems: the counting of people in the crowd and the detection of abnormal situations where accidents are likely to occur. The counting issue generally yields the questions of *i)* background subtraction and *ii)* feature tracking. Concerning the tracking, the choice of the features to extract is determinant. Typical methods are based on appearance

models [5,6] that exhibit different sensibilities to inter persons occlusions. The temporal and spatial consistency of the tracked features can be obtained through clustering methods [7]. In [8], Brostow and Cipolla successfully argue that only the apparent motion in image space is relevant to singularize individual in the crowd flow. When several hundreds of pedestrians are present in the crowd most of the conventional tracking methods (like Kalman filters or particle filtering [9,10]) fail, because the degradation of the visual features related to single individuals disturbs the analysis. Moreover, the large induced state space yields computationally too expensive problems. In those cases, the analysis of the crowd sequence may amount to the analysis of a crowd **flow** that have global properties and may be treated as a whole. Works related to this class usually tend to solve the different problems of event detection or changes in the flow rate [11,12]. The analysis usually takes as input the apparent motion in the image space (optical flow). In [12], unsupervised feature clustering is used to define normal motion patterns, and HMMs are used to detect particular situations. The method proposed by Ali and Shah in [11] allows to segment the crowd flow with regions of substantially different dynamics by examining the Lagrangian coherent structures in the flow. In some sense, this Eulerian perception of the crowds dynamics (that assumes that the crowd can have fluid-like properties) opposes to a Lagrangian view of the indivual tracking problem.

In our method, we propose to use a physical crowd model to guide the analysis knowledge. This physical model is new and can be related to the continuous formulations of crowd dynamics such as the one of Hughes [13]. In a recent work [14], Ali and Shah also use an *a priori* knowledge on the crowd dynamics by using a scene structured based force model that guides a tracking process. This work differs from ours in the sense that our goal is not the individual tracking of people in the scene, but rather a characterization of the entire crowd flow.

Given a dynamic model related to a phenomenon, different filtering processes have been developed to extract time-consistent parameters. Recently, Papadakis *et al.* have exploited the variational assimilation principle to successfully extract time-consistent and high-dimensional state spaces (dense motion fields, curves, physical parameters such as vorticity, ...) directly from images [15,16]. The variational assimilation methods perform in batch and allow to explicitly enforce a (more or less confident) dynamic model to the variables to recover. Their framework is expressed by means of an *adjoint formulation*: adjoint variables are introduced and enable to compute easily the gradient of the cost-function. The resulting algorithm consists of iterating a forward integration of the evolution model and a backward integration of the adjoint evolution model guided by a discrepancy measurement between the state variable and the available noisy observations. This efficient procedure authorizes to refine an initial condition (which can be low confident) as well as the deviations *wrt.* the dynamic model.

In this paper, we suggest to define a dynamic model for crowds that couples the velocity to the disturbance potential. Recalling that the main difficulty of crowd analysis concerns the estimation of this last potential, this model will be a support to estimate this quantity using variational assimilation. This is the scope of the next section.

3 Estimation by Coupling Observations/Model with Variational Assimilation

3.1 Overview of the Method

We recall here that our objective is to estimate at the same time the apparent motion of the crowd and its disturbance potential in the image sequence. An overall schema is given in Figure 1. We take as input the original images and two user-defined information: the eventual position of obstacles and some predefined destination areas in the image. These two information are combined to compute a potential function that conveys information on the optimal directions of displacements for the crowd. From the input images are also derived some initializations for our algorithm as well as the observations (that mainly consist in the apparent motion between image pairs). These are used in the assimilation process, that tries to match, through an iterative process, the observations and the evolution of the dynamical process. As a result, a complete sequence of velocity and disturbance potential are computed.

We present some background on variational assimilation in the next section (3.2), while our model, along with implementation issues, is thoroughly described in the ending part of this section (3.3).

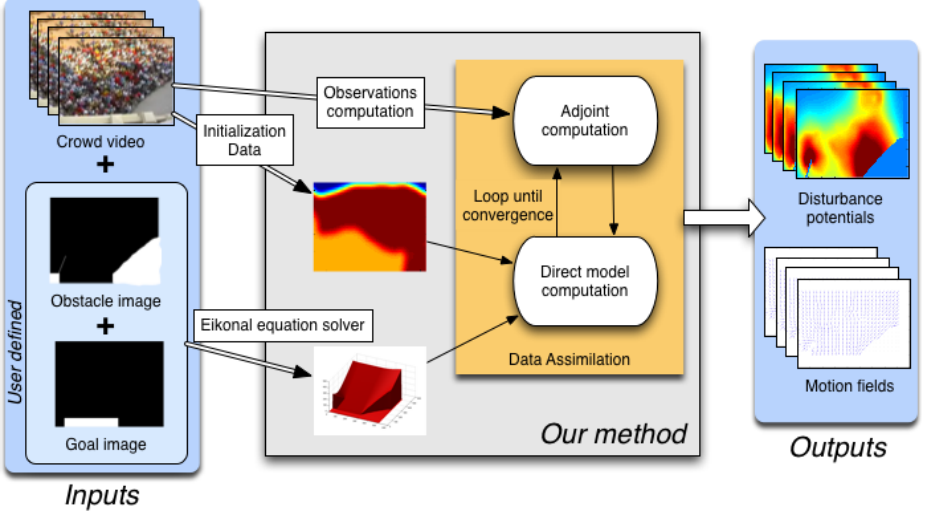


Fig. 1. Method overview

3.2 Variational Assimilation

In this section, the key points required for the comprehension of the variational assimilation are introduced. A complete and detailed presentation would be out of the scope of this paper. For details, we refer the reader to [3,4].

Our problem consists in recovering, from an initial condition, a system's state $\boldsymbol{\mathcal{X}}$ partially observed and driven with an approximately known dynamic. This formalizes as finding $\boldsymbol{\mathcal{X}}(\boldsymbol{x}, t)$, for any location $\boldsymbol{x}$ at time $t \in [t_0, t_f]$, that satisfies the system:

$$\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t}(\boldsymbol{x}, t) + \mathbb{M}(\boldsymbol{\mathcal{X}}(\boldsymbol{x}, t)) = \boldsymbol{\epsilon}_m(\boldsymbol{x}), \tag{1}$$

$$\boldsymbol{\mathcal{X}}(\boldsymbol{x}, t_0) = \boldsymbol{\mathcal{X}}_0(\boldsymbol{x}) + \boldsymbol{\epsilon}_n(\boldsymbol{x}), \tag{2}$$

$$\boldsymbol{\mathcal{Y}}(\boldsymbol{x}, t) = \mathbb{H}(\boldsymbol{\mathcal{X}}(\boldsymbol{x}, t)) + \boldsymbol{\epsilon}_o(\boldsymbol{x}, t), \tag{3}$$

where $\mathbb{M}$ is the non-linear operator relative to the dynamics, $\boldsymbol{\mathcal{X}}_0$ is the initial vector at time t_0 and $(\boldsymbol{\epsilon}_n, \boldsymbol{\epsilon}_m)$ are (unknown) variables relative to noise on the dynamics and the initial condition respectively. Besides, noisy measurements $\boldsymbol{\mathcal{Y}}$ of the unknown state are available through the non-linear operator $\mathbb{H}$ up to $\boldsymbol{\epsilon}_o$. To estimate the system's state a common methodology consists in defining a cost-function $\mathcal{J}$ based on the three previous relations to minimize. The evaluation of $\boldsymbol{\mathcal{X}}$ can be done by canceling the gradient of this cost function. Unfortunately, the estimation of such gradient is in practice unfeasible for large system's state since it requires to compute perturbations along all the components of $\boldsymbol{\mathcal{X}}$. A way to cope this difficulty, firstly proposed by Lions in [3], is to write *an adjoint formulation* of the problem. It can be shown that this yields the following algorithm:

1. Starting from $\tilde{\boldsymbol{\mathcal{X}}}(\boldsymbol{x}, t_0) = \boldsymbol{\mathcal{X}}_0(\boldsymbol{x})$, perform a *forward* integration: $\frac{\partial \tilde{\boldsymbol{\mathcal{X}}}}{\partial t} + \mathbb{M}(\tilde{\boldsymbol{\mathcal{X}}}) = 0$
2. $\tilde{\boldsymbol{\mathcal{X}}}$ being available, **find the adjoint variables** $\boldsymbol{\lambda}(\boldsymbol{x}, t)$ with the *backward* equation:

$$\boldsymbol{\lambda}(t_f) = 0 \;\; ; \;\; -\frac{\partial \boldsymbol{\lambda}}{\partial t}(t) + \left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger} \boldsymbol{\lambda}(t) = \left(\frac{\partial \mathbb{H}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger} R^{-1}(\boldsymbol{\mathcal{Y}} - \mathbb{H}(\tilde{\boldsymbol{\mathcal{X}}}))(t) \tag{4}$$

3. **Update the initial condition** : $d\boldsymbol{\mathcal{X}}(t_0) = B\boldsymbol{\lambda}(t_0) + d\boldsymbol{\mathcal{X}}(t_0)$;
4. $\boldsymbol{\lambda}$ being available, **find the state space** $d\boldsymbol{\mathcal{X}}(t)$ from $d\boldsymbol{\mathcal{X}}(t_0)$ with the *forward* integration

$$\frac{\partial d\boldsymbol{\mathcal{X}}}{\partial t}(t) + \left(\frac{\partial \mathbb{M}}{\partial \tilde{\boldsymbol{\mathcal{X}}}}\right) d\boldsymbol{\mathcal{X}}(t) = Q\boldsymbol{\lambda}(t) \tag{5}$$

5. **Update** : $\tilde{\boldsymbol{\mathcal{X}}} = \tilde{\boldsymbol{\mathcal{X}}} + d\boldsymbol{\mathcal{X}}$
6. **Loop** to step 2 until convergence

where the matrices B, Q, R are relative to the covariance of the errors $(\boldsymbol{\epsilon}_m, \boldsymbol{\epsilon}_n, \boldsymbol{\epsilon}_o)$, $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)$ and $\left(\frac{\partial \mathbb{H}}{\partial \boldsymbol{\mathcal{X}}}\right)$ are the *linear tangent operators* of $\mathbb{M}$ and $\mathbb{H}$ respectively[1] and $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger}$ and $\left(\frac{\partial \mathbb{H}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger}$ their *adjoint operators*[2]. Intuitively, the adjoints variables $\boldsymbol{\lambda}$ contain information about the discrepancy between the observations and the dynamic model. They are computed from a current solution $\tilde{\boldsymbol{\mathcal{X}}}$ with the backward integration (4) that implicates both observations and dynamical operators. This deviation information between data/model is then used to refine the initial condition (step 3) and to recover the system state through an imperfect dynamic model where errors are $Q\boldsymbol{\lambda}$ (step 4). Note that if the dynamic is supposed to be perfect (like in many physical applications), the associated covariance Q is null and the algorithm only refines the initial condition.

[1] The linear tangent of an operator $\mathbb{A}$ is the Gâteaux derivative : $\lim_{\beta \to 0} \frac{\mathbb{A}(X+\beta\theta)-\mathbb{A}(X)}{\beta}$.

[2] The adjoint $\mathbb{A}^{\dagger}$ of a linear operator $\mathbb{A}$ on a space $\mathcal{D}$ is such as $\forall x_1, x_2 \in \mathcal{D}$, $< \mathbb{A}x_1, x_2 > = < x_1, \mathbb{A}^{\dagger}x_2 >$.

From the previous algorithm, a complete assimilation system is then defined with *i)* a dynamic model $\mathbb{M}$; *ii)* an observation operator $\mathbb{H}$; *iii)* an initial condition and *iv)* the error covariance matrices B, Q and R. The next section defines all these components for our problem.

3.3 Dynamic Model, Observations and Covariance

Proposed dynamic model for crowd behavior. The aim of this part is to design a simple dynamic model for crowds that will be used for the assimilation. The system's state $\boldsymbol{\mathcal{X}}$ is composed of the two components of interest that are the velocity field $\boldsymbol{v} = (u, v)^T$ and of the disturbance potential of the crowd D ($\boldsymbol{\mathcal{X}} = (u, v, D)^T = (\boldsymbol{v}, D)^T$). Let us define a model for the velocity evolution.

Velocity modeling. In order to get a prior knowledge of the displacement of the crowd, we assume that all human share the same goal and that the topology (obstacles) of the analyzed scene is available. In a first place our methodology is thus restricted to image sequences exhibiting one main flow of pedestrians. Reasonably assuming that each pedestrian aims at minimizing their travel time to their objectives, the optimal direction at a given location can be modeled as the gradient of a potential function Φ defined over the whole domain D. This potential is the solution of the classical Eikonal equation which has among others been widely used in the context of path planing [17,18]. For a given scene, we then derive an optimal field $\boldsymbol{V} = (U, V)^T = \nabla\Phi$ of the pedestrians that corresponds to the theoretical normalized direction of a pedestrian without any constraint. If now the pedestrians evolve in a crowded environment, we assume that if their velocity differs from the optimal direction, this is due to a disturbance into the scene (density, pressure, ...). Therefore, we propose the following dynamical model:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \alpha\Big(\boldsymbol{V}(\boldsymbol{x}, t) \underbrace{-\beta\nabla D(\boldsymbol{x}, t)}_{\text{disturbance repulsion}}\Big) \tag{6}$$

where α and β are two constant coefficients that depend on the global speed of the scene.

Disturbance potential modeling. As for the disturbance potential modeling, we simply assume that this scalar quantity is transported by the motion field and is also eventually diffused along time. This corresponds to a simple physical equation of transport of a scalar. It then obeys to a classical advection-diffusion relation:

$$\frac{\partial D(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla D(\boldsymbol{x}, t) = \delta\Delta D(\boldsymbol{x}, t). \tag{7}$$

where δ is a small diffusing parameter. Finally, the complete dynamical system of $\boldsymbol{\mathcal{X}} = (\boldsymbol{v}, D)^T$ reads (with $(\bullet) = (\boldsymbol{x}, t)$):

$$\begin{bmatrix} \boldsymbol{v}(\bullet) \\ \frac{\partial D(\bullet)}{\partial t} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \alpha\beta\nabla \\ 0 & \boldsymbol{v}(\bullet)\cdot\nabla - \beta\Delta \end{bmatrix}}_{\mathbb{M}(\boldsymbol{\mathcal{X}})} \begin{bmatrix} \boldsymbol{v}(\bullet) \\ D(\bullet) \end{bmatrix} = \begin{bmatrix} \alpha\boldsymbol{V}(\bullet) \\ 0 \end{bmatrix} + \boldsymbol{\epsilon_{\mathbf{m}}} \tag{8}$$

To suppress the obstacle influence in the computation of the gradient ∇, we have used non-symmetric finite-difference in their neighborhood. Concerning the Laplacian operator Δ related to the diffusion in (7), we have applied an anisotropic operator that do not diffuse into the obstacles.This dynamic model $\mathbb{M}$ is non-linear due to the advection term $\boldsymbol{v}(\bullet)\cdot\nabla$ that depends on the density. In practice, at a given iteration n, the velocity $\boldsymbol{v}$ used for the advection is the one obtained at iteration $n-1$ so that the operator is linear. The associated tangent linear $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)$ is then itself. The analytical expression of the adjoint $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger}$ is more tricky to obtain but in our implementation, we have used the fact that its discrete version is the transpose of the discrete version of $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)$ [19].

Let us now turn to the observations of the state variables.

Observations: velocity based on optical-flow. As mentioned above, only the motion fields $\boldsymbol{v}$ can be accurately observed from the images, the disturbance potential being a tedious quantity to estimate. Starting from the well-known optical flow constraint equation (ofce), one can assume, to cope with the aperture problem, that the unknown optic flow vector at a location $\boldsymbol{x}$ is constant within some neighborhood of size n [20]. The motion field respects then:

$$K_n * \underbrace{\left(\frac{\partial I(\boldsymbol{x},t)}{\partial t} + \nabla I(\boldsymbol{x},t)\cdot\boldsymbol{v}(\boldsymbol{x},t)\right)}_{dI/dt} \approx 0, \tag{9}$$

where I stands for the luminance function and K_n is a Gaussian kernel of standard deviation n. From the previous relation, the observation system $\boldsymbol{\mathcal{Y}}(\boldsymbol{x},t)=\mathbb{H}(\boldsymbol{x},t)\boldsymbol{\mathcal{X}}(\boldsymbol{x},t)+\boldsymbol{\epsilon}_o$ can be defined with (noting $I_\bullet = \partial I/\partial\bullet$):

$$\boldsymbol{\mathcal{Y}}(\boldsymbol{x},t) = K_n * I_t(\boldsymbol{x},t) \text{ and } \mathbb{H}(\boldsymbol{x},t) = \left[-K_n * I_x(\boldsymbol{x},t), -K_n * I_y(\boldsymbol{x},t), 0\right]. \tag{10}$$

This observation operator involves only the motion field. This means that the correction on the disturbance potential will uniquely be achieved by relying on motion observations. From a computational point of view, this operator is linear. The associated tangent linear and adjoints are then derived in the same way than previously.

Covariances and initialisations. For the initialization, we only need to get the disturbance potential since the corresponding initial velocity field is obtain from (6). The choice of this density depends on the scene to be analyzed. In our experiments, it was roughly set manually and filtered with a Gaussian kernel. Noting that the assimilation process refines this initialization, this latter can be only issued from a coarse and manual estimation.

The covariance matrix of the initial condition B and the covariance matrix of the dynamic model parameter Q have been fixed to constant diagonal matrices (no spatial prior on the validity of the model and the initial density are available). Concerning the observation covariance R, we have used $R = R_{max} + (R_{min} - R_{max})(1 - \exp(-\|\nabla I\|/\sigma^2))$. This states that when the image brightness does not contain gradients, the usual ofce is not valid and the covariance is maximal. At the opposite, when high gradients appears, the ofce is confident and R is low.

4 Experimentations and Discussion

In this section we present some experimental results obtained with our method. We first begin by giving some comparison elements in a synthesized case. We then present the results obtained on a real crowd video. This part is concluded by a discussion on the proposed method.

For all the following experimentations, the value used the optimal control system were $B = 0.5I_d$, $Q = 0.1I_d$, $R_{min} = 0.5$, $R_{max} = 5$, $\sigma = 9$.

4.1 Validation

Ground truth generation. Our goal here is to compare the obtained results to some ground truth. In order to get a flexible validation pipeline, we used synthesized crowd scenes. Designing our simulation framework was done with the following constraints: *i)* a totally different crowd simulation model than the one used for assimilation *ii)* the video should present realistic details in terms of visual appearance and fine pedestrians motion (such as arms and legs balancing). It is then possible to confront the virtual ground truth to the results of the assimilation process.

(a) $t = 3s$ (b) $t = 15s$ (c) $t = 23s$

Fig. 2. Synthesized crowd scene

The virtual sequence has been generated using an agent-based crowd simulation model slightly derived from Helbing's model [21], and virtual characters including walking motions acquired through motion capture (see Figure 2). The density maps $\rho(\boldsymbol{x}, t)$ and the velocity fields $\mathbf{U}(\boldsymbol{x}, t)$ are computed from the agent model to the grid using a Gaussian kernel regularisation (we used $\sigma = 0.5$), and will be considered as the truth.

Dynamic model results. The integration of the proposed simulated dynamic model provides important information. First, one can see (Figure 4) that it is able, through the disturbance potential, to locate the places where the pedestrians are assumed to be effectively the most disturbed. But in counterpart, velocities tend to quickly decrease over the flow, and do not match to the supposed freedom of move once the obstacle overtaken.

The assimilation, Figure 5, improves the results. As shown on D maps, the values fit much better the truth density maps, which is an important part of the disturbance potential, in time and space, and particulary after the pedestrians have passed the obstacle. The velocities norms also fit well to the truth, and the correlation between high disturbance and low speed is clearly effective. Only the back part of the obstacle remains uncorrectly managed.

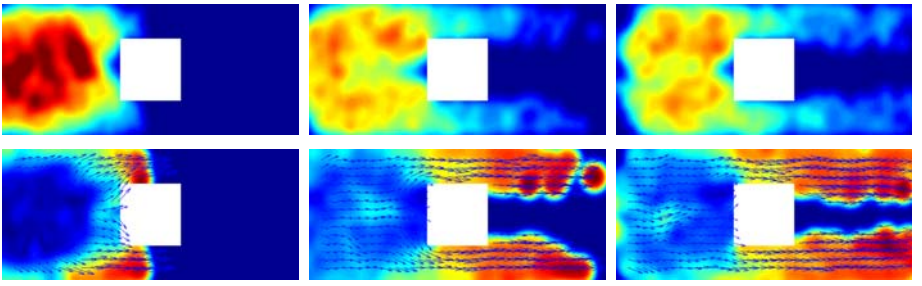


Fig. 3. Ground truth (agent-based model) - Upper line: Density maps (0 - dark blue, to 5 - dark red) $pedestrians/m^2$- Lower line: Velocity fields and norm (0 to 1 m/s)

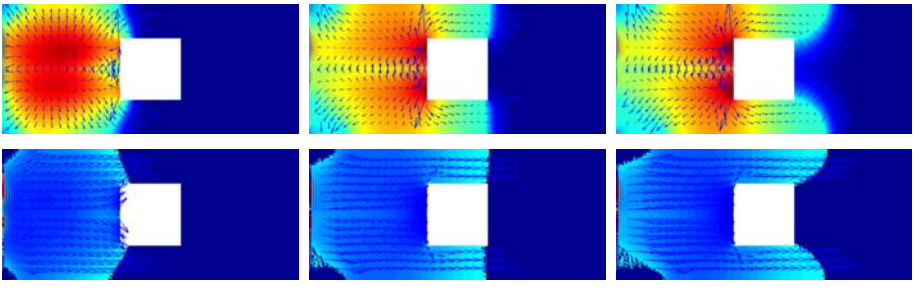


Fig. 4. Simulated dynamic model - Upper line: Disturbance (D) maps (0 to 5) with associated gradient (that informs about potential repulsion) - Lower line: Velocity fields and norm

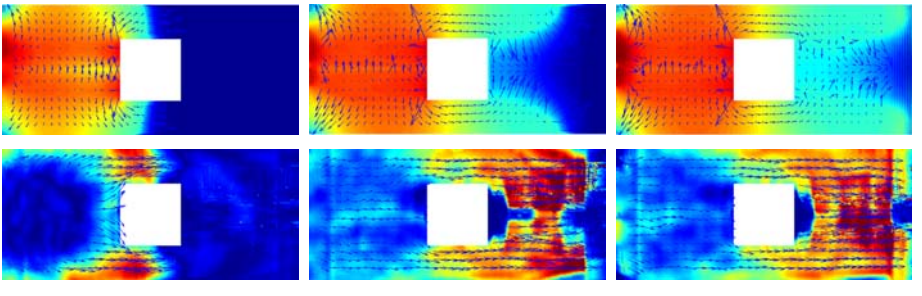


Fig. 5. Assimilated dynamic model - Upper line: Disturbance (D) maps (0 to 5) with associated gradient - Lower line: Velocity fields and norm (0 to 1)

RMS comparison. Because of its strong link with pedestrian disturbance, it is relevant to compare the norm of the velocity to the ground truth. In this purpose, we used the standard RMS function. One can see Figure 6, that the assimilation of D greatly improved the results as compared to the simple model simulation. The global error is almost lowered by a factor 2. However, the model, as said before, is not well suited to the back part of the obstacle.

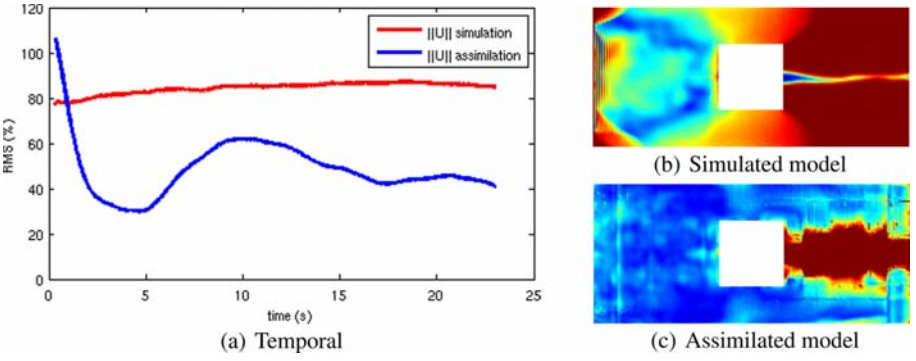


Fig. 6. Temporal RMS (a) - Spatial RMS (0 to 100%) at $t = 23s$ on velocity norm (b) and (c)

Let us now turn to experiments on real sequences.

4.2 Experimention on Real Crowd

The real sequence shows a crowd entering a railway station in the Principality of Monaco (Figure 7). This example is interesting since a variety of phenomena are present: a continuous flow at the beginning followed by a compression of some peoples in the left part of the images. In addition, the limit of the door is a barrier that creates an opposite flux in the crowd flow. In this example, our method has detected two sensible areas where the disturbance potential is growing larger : the end of the barrier and the wall on the right of the image. This is very informative for safety engineers, since it allows to highlight potential risky zones. From an online surveillance system point of view, our method can detect critical disturbance elevations and thus would allow to trigger alarms. It is also possible to connect this information in some motion pattern detector such as presented in [12]. Those aspects have been left as perspectives. Let us remark here that the problem of validation is difficult since no ground truth is available. Nevertheless, from the state-of-the-art on crowd behavior, our estimations seem coherent.

4.3 Discussion

One drawback of our method is that it requires an overhead view of the crowd flow, which is not always available in surveillance system. However, this constraint can be partially alleviated provided that a camera calibration can be given, allowing to project back the observations in a correct frame. Secondly, our method is well adapted to dense crowds, where it is possible to assume that each individual is driven by an underlying flow. With only a few pedestrians in the video, this assumption does not hold anymore. The range of validity of our system, in terms of crowd density, has thus to be clearly established. Also, we have only treated the case where only one type of crowd (one common goal) is present in the image sequence. We believe that it is possible to handle with our dynamical model more than one flux of people (distinct goals), thus allowing to some extent a segmentation of the crowd flow. Finally, the disturbance potential is a

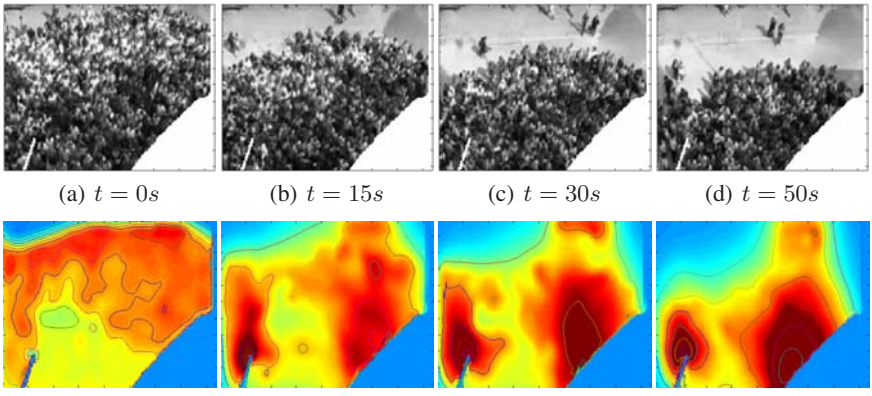


Fig. 7. (a) to (d) Images of the real sequence - (e) to (h) Estimated disturbance potential maps

combination of several physical quantities such as density or pressure. We plan to use a more sophisticated physical crowd model to estimate each of these quantities separately in a variational assimilation framework. Those aspects are part of our future works.

5 Conclusion and Future Works

We have presented in this paper a complete framework dedicated to the analysis of dense crowd video sequences. Our approach relies on the coupling of observed data extracted from the image sequences and an ad-hoc crowd dynamics model that capture the intrinsic relation between velocity and a disturbance potential (related to density, pressure, ...) in the crowd flow. This allows to derivate an efficient framework that computes from a crowd image sequences both the disturbance potential and the velocity fields over the entire sequence. The estimated disturbance potential has proved to be very interesting in highlighting the main characteristics of a scene. This can serve as inputs for event detection system or participate to our global comprehension on the underlying dynamics of human crowds. Future works will consider a more precise investigation of the disturbance potential to separate from this latter density, pressure and other quantities. We will also investigate sequences containing distinct fluxes of persons.

Acknowledgements. This work was supported by the french Région Bretagne AS-Foule project, the University of South Brittany Math-Stic department and a part by the ANR program ANR-06-JCJC-0083.

References

1. Helbing, D., Johansson, A., Al-Abideen, H.: Dynamics of crowd disasters: An empirical study. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 75(4), 046109 (2007)
2. Courty, N., Corpetti, T.: Crowd motion capture. CAVW, proc. of CASA 2007 18(4-5), 361–370 (2007)

3. Lions, J.L.: Optimal Control of Systems Governed by Partial Differential Equations. Springer, Heidelberg (1971)
4. Le-Dimet, F., Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. Tellus, 97–110 (1986)
5. Yang, T., Li, S., Pan, Q., JingLi: Real-time multiple object tracking with occlusion handling in dynamic scenes. In: CVPR, San Diego, USA, June 2005, pp. 406–413 (2005)
6. Zhao, T., Nevatia, R.: Tracking multiple humans in crowded environment. In: CVPR, Washington, DC, USA, pp. 406–413 (2004)
7. Rabaud, V., Belongie, S.: Counting crowded moving objects. In: CVPR, New York, June 2006), pp. 705–711 (2006)
8. Brostow, G.J., Cipolla, R.: Unsupervised bayesian detection of independent motion in crowds. In: CVPR, NYC, June 2006, pp. 594–601 (2006)
9. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
10. Khan, Z., Balch, T., Dellaert, F.: MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. IEEE PAMI 28(12), 1960–1972 (2006)
11. Ali, S., Shah, M.: A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: CVPR, Minneapolis, Minnesota, June 2007, pp. 1–6 (2007)
12. Andrade, E., Blunsden, S., Fisher, R.: Modelling crowd scenes for event detection. In: ICPR, Hong Kong, China, August 2006, pp. 175–178 (2006)
13. Hughes, R.L.: The flow of human crowds. Annual revue of Fluid. Mech. 20(10), 169–182 (2003)
14. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 1–14. Springer, Heidelberg (2008)
15. Papadakis, N., Corpetti, T., Mémin, E.: Dynamically consistent optical flow estimation. In: Proc. Int. Conf. Comp. Vis. (ICCV 2007), Rio de Janeiro, Brazil (October 2007)
16. Papadakis, N., Mémin, E.: A variational technique for time consistent tracking of curves and motion. Journal of Mathematical Imaging and Vision (2008) (available online first)
17. Polymenakos, L., Bertsekas, D., Tsitsiklis, J.: Implementation of efficient algorithms for globally optimal trajectories. IEEE Trans. on Automatic Control 43, 278–282 (1998)
18. Kimmel, R., Sethian, J.: Optimal algorithm for shape from shading and path planning. J. of Math. Ima. and Vis. 14(3), 237–244 (2001)
19. Talagrand, O.: Variational assimilation. Adjoint equations. Kluwer Academic Publishers, Dordrecht (2002)
20. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI, Vancouver, Canada, pp. 674–679 (1981)
21. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407(1), 487–490 (2000)

Human Action Recognition Using HDP by Integrating Motion and Location Information

Yasuo Ariki[1], Takuya Tonaru[2], and Tetsuya Takiguchi[1]

[1] Organization of Advanced Science and Technology, Kobe University,
1–1, Rokkodai, Nada, Kobe, Hyogo, Japan
{ariki,takigu}@kobe-u.ac.jp
[2] Graduate School of Engineering, Kobe University,
1–1, Rokkodai, Nada, Kobe, Hyogo, Japan
http://www.me.cs.scitec.kobe-u.ac.jp/index.html

Abstract. The method based on local features has an advantage that the important local motion feature is represented as bag-of-features, but lacks the location information. Additionally, in order to employ an approach based on bag-of-features, language models represented by pLSA and LDA (Latent Dirichlet Allocation) have to be applied to. These are unsupervised learning, but they require the number of latent topics to be set manually. In this study, in order to perform the LDA without specifying the number of the latent topics, and also to deal with multiple words concurrently, we propose unsupervised Multiple Instances Hierarchical Dirichlet Process MI-HDP-LDA by employing the local information concurrently. The proposed method, unsupervised MI-HDP-LDA, was evaluated for Weizmann dataset. The average recognition rate by LDA as conventional method was 61.8% and by the proposed method it was 73.7%, resulting in 11.9 points improvement.

Keywords: Motion, location, action recognition, LDA, HDP, HDP-LDA.

1 Introduction

Human action recognition is a challenging problem in computer vision. It can be applied to many applications such as surveillance, scene understanding, care monitoring, sport analysis, etc. In fact, for computers to support human works, they need to understand the human activities, so that human actions, the primitive units of human activities, become important information.

A lot of work has been done in recognizing human actions. Bobick[1] used motion energy images (MEI) and motion history images (MHI). Those shape descriptors represented information of human motion –"where" and "how". Grundmann[2] used 3D shape context extended into a temporal dimension. That method represented a human action as a histogram of 3D points by sampling shape of silhouette. Additionally, it increased the sampling density in the domain of fast moving body parts. Efros[3] used optical flow field for human figures at

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 291–300, 2010.

each frame. Their methods presented the human body as a whole to understand human actions as concatenation of motion and pose.

In contrast to the above methods, local motion approaches represent human action as a set of distinguished local motion features. Laptev[4] proposed space-time interest points using Harris operator extended into temporal and adapted scale. Dollár[5] proposed cuboid with local motion descriptor at interest point detected using separable linear filters. Scovanner[6] used 3D SIFT descriptor extended into a temporal dimension for these interest point. These approaches are effective to characterize distinguished local motion included in the action. Moreover, since these features can be represented as a histogram, language models with unsupervised learning can be applied to the features. These approaches are called bag-of-words and the features obtained as a result are called word. Niebles[7] classified actions by applying pLSA that is one of the language models, and Wang[8] used Semi-LDA.

However, local motion approaches do not take the location information into consideration. Moreover, bag-of-words approach using language model requires the number of latent topics, corresponding to action classes, to be set manually. In this study, in order to perform the LDA (Latent Dirichlet Allocation) without specifying the number, and also to deal with multiple words concurrently in an unsupervised manner, we propose unsupervised Multiple Instances Hierarchical Dirichlet Process MI-HDP-LDA. MI-HDP-LDA is the model capable of generating words from the latent topics. Hence it can provide co-occurrence of words occurring simultaneously. Moreover, it can estimate the number of latent topics automatically by using Hierarchical Dirichlet Processes(HDP).

The rest of this paper is organized as follows. In section 2, our basic idea is described and in section 3, motion feature and location information are described. In section 4, Hierarchical Dirichlet Processes - Latent Dirichlet Allocation (HDP-LDA) is briefly described. In section 5, MI-HDP-LDA is proposed to deal with features occurring concurrently. In section 6, the experimental result is described for Weizmann dataset introduced in [9] to evaluate our algorithm. Section 7 is for conclusion of this paper.

2 Basic Idea

Our study is motivated by the conventional approaches which extract local features by detecting interest points. Our basic idea is to extract various types of information at interest points such as motion feature, location information and limbs parts, etc for understanding human actions.

This paper regards the motion in terms of information “where” and “how”. For “where”, the relative position in human region is used as location word and for “how”, the motion feature is used as motion word at the interest point. Niebles[7] method is employed as interest point detection algorithm and motion descriptor.

3 Features

In this section, we describe briefly the motion feature proposed by Dollár[5] and location information representing "where".

3.1 Motion Feature

Assuming a stationary camera or a process that can account for camera motion, separable linear filters are applied to the video to obtain the response function as follows,

$$\begin{aligned} R(x,y) =& \Big(I(x,y) * g(x,y;\sigma) * h_{ev}(t;\tau,\omega)\Big)^2 \\ &+ \Big(I(x,y) * g(x,y;\sigma) * h_{od}(t;\tau,\omega)\Big)^2, \end{aligned} \tag{1}$$

where $g(x,y;\sigma)$ is a 2D Gaussian smoothing kernel, applied only along the spatial dimensions, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, which are defined as follows,

$$\begin{aligned} h_{ev}(t;\tau,\omega) &= -\cos(2\pi t\omega)e^{-\frac{t^2}{\tau^2}}, \\ h_{od}(t;\tau,\omega) &= -\sin(2\pi t\omega)e^{-\frac{t^2}{\tau^2}}. \end{aligned} \tag{2}$$

The two parameters σ and τ correspond to the spatial and temporal scales of the filters respectively. To make the response function effective, $\omega = 4/\tau$ was employed.

This function detects any regions where complex motion is caused spatially. In fact, a region with complex motion can induce a strong response, but a region with simple translational motion will not induce a strong response. The spatial-temporal interest points are extracted around the local maxima of the response function. At each interest point, a spatial-temporal cube is extracted that contains the output of the response function. Its size is approximately six times the spatial and temporal scales along each dimension. To obtain a motion descriptor, the brightness gradients are computed at all the pixels in the cube and are concatenated to form a vector. Then PCA is applied to reduce the dimensionality of the descriptors.

In order to obtain the cluster prototypes, a k-means algorithm is applied to the descriptors. Then each descriptor is assigned a descriptor type by mapping it to the prototype. Therefore a collection of descriptors included in a video is represented as a histogram of the descriptor types. The descriptor types are called motion words.

3.2 Location Information

As shown in Fig.1, the human rectangle region is divided into $N \times M$ blocks. Each block indicates a relative position within a human rectangle region. The

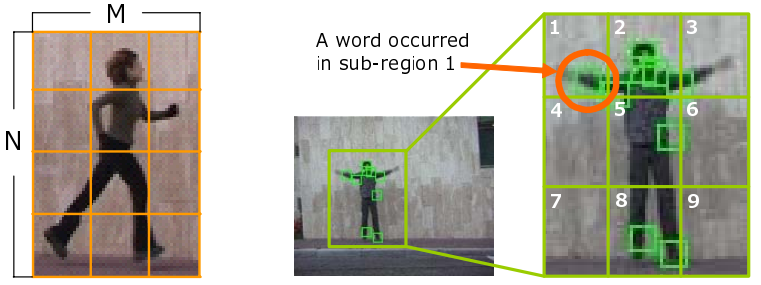


Fig. 1. (a)Human rectangle region is split into $N \times M$ blocks. (b)A motion word enclosed in orange circle also has a location word occurred in sub-region 1. This indicates that these words occurred at the interest point have motion feature and location information concurrently.

extremities of the motion such as arm and foot movement are extracted as the interest points and therefore they have two kinds of information, namely, motion feature and location information. The person detection is done manually in the experiment to exclude the detection errors, but it will be automatically performed by frame subtraction.

4 Hierarchical Dirichlet Processes - Latent Dirichlet Allocation

Our model is based on Hierarchical Dirichlet Processes - Latent Dirichlet Allocation (HDP-LDA) [10]. HDP-LDA is extended from LDA[11] by using multiple DPs. In contrast to LDA with a finite mixture model, HDP-LDA is an

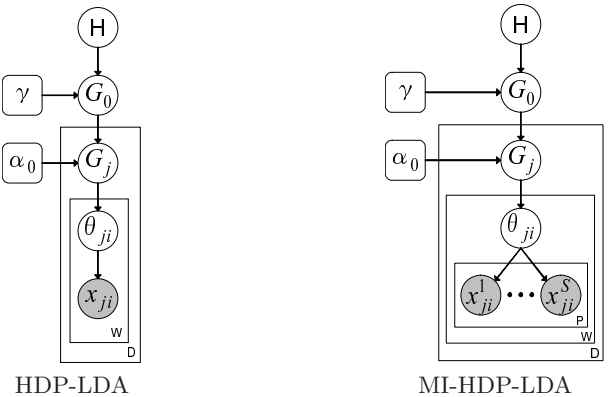


Fig. 2. Graphical representation of HDP-LDA model and MI-HDP-LDA modelD

infinite mixture model sharing topics across multiple DPs given an underlying base measure H. The graphical model of HDP-LDA is depicted in Fig.2 left.

Suppose we are given a collection D of video clips $\{1, \cdots, j, \cdots, J\}$. Video clip j has a collection of words $\{x_{j1}, \cdots, x_{ji}, \cdots, x_{jI}\}$ as described in the previous section, where x_{ji} is the i-th word in video clip j.

The global measure G_0 has a probability distribution decided by Dirichlet process(DP)[12] with concentration parameter γ and base probability measure H as follows,

$$G_0 \mid \gamma, H \sim \mathrm{DP}(\gamma, H). \tag{3}$$

DP is a process that a random probability measure is distributed as a Dirichlet distribution with concentration parameter and base probability measure. The random probability measure G_j for designated video clip j has a distribution decided by a Dirichlet process with concentration parameter α_0 and base probability measure G_0 under conditional independence given G_0,

$$G_j \mid \alpha_0, G_0 \sim \mathrm{DP}(\alpha_0, G_0). \tag{4}$$

Such hierarchical process of distributing a probability measure is Hierarchical Dirichlet Processes. For each video clip j, let $\theta_{j1}, \theta_{j2}, \cdots$ be independent and identically distributed random variables sampled from G_j. Each θ_{ji} is a topic corresponding to a single word x_{ji}. The likelihood is given by:

$$\theta_{ji} \mid G_j \sim G_j \tag{5}$$

$$x_{ji} \mid \theta_{ji} \sim F(\theta_{ji}), \tag{6}$$

where $F(\theta_{ji})$ denotes the probability distribution of the observation x_{ji} given θ_{ji}. Words are generated independently and distributed identically from the selected topic.

5 Multiple Instances HDP-LDA

HDP-LDA model generates a single word x_{ji} from the corresponding topic θ_{ji}, but not the multiple instances of the word concurrently such as motion word and location word. To solve this problem, we propose Multiple Instances HDP-LDA(MI-HDP-LDA) that allows multiple concurrent instances of the word.

MI-HDP-LDA can generate multiple instances $\boldsymbol{x}_{ji} = \{x_{ji}^1, \cdots, x_{ji}^S\}$ of the word from latent topic θ_{ji}. Each instances x_{ji}^s is generated as follows,

$$x_{ji}^s \mid \theta_{ji} \sim F_s(\theta_{ji}), \tag{7}$$

where $F_s(\cdot)$ is the distribution of x_{ji}^s given the latent topic θ_{ji}. Here S indicates the number of instances of the word i such as motion word and location word in the video clip j.

Next, we describe the Gibbs sampling scheme for MI-HDP-LDA in CRF (Chinese Restaurant Franchise) representation[10]. Basic scheme is exactly similar to HDP-LDA except it obtains the likelihood of generating $\boldsymbol{x}_{ji}$.

The variable $\boldsymbol{x}_{ji}$ is multiple instances of word i observed concurrently, so that $\boldsymbol{x}_{ji}$ is a vector with the size of S. Each $\boldsymbol{x}_{ji}$ is assumed to be generated based on a distribution $F(\theta_{ji})$. Let the factor θ_{ji} be associated with the table t_{ji} in CRF, i.e., let $\theta_{ji} = \psi_{jt_{ji}}$. The random variable ψ_{jt} is a topic k_{jt}; i.e., $\psi_{jt} = \phi_{k_{jt}}$. The prior over the parameters ϕ_k is H. Let $z_{ji} = k_{jt_{ji}}$ denote the topic associated with the observation $\boldsymbol{x}_{ji}$. We use the notation n_{jtk} to denote the number of customers in restaurant j at table t eating dish k, while m_{jk} denotes the number of tables in restaurant j serving dish k. Marginal counts are represented with dots.

Let $\boldsymbol{x} = \{\boldsymbol{x}_{ji} : \text{all } j, i\}$C $\boldsymbol{x}_{jt} = \{\boldsymbol{x}_{ji} : \text{all } i \text{ with } t_{ji} = t\}$C $\boldsymbol{t} = \{t_{ji} : \text{all } j, i\}$C $\boldsymbol{k} = \{k_{jt} : \text{all } j, t\}$C $\boldsymbol{z} = \{z_{ji} : \text{all } j, i\}$C $\boldsymbol{m} = \{m_{jk} : \text{all } j, k\}$C $\boldsymbol{\phi} = \{\phi_1, \cdots, \phi_K\}$. When a superscript is attached to a set of variables or a count, e.g., $\boldsymbol{x}^{-ji}$,$\boldsymbol{k}^{-jt}$ or $n_{jt\cdot}^{-ji}$, this means that the variable corresponding to the superscripted index is removed from the set or from the calculation of the count.

Sampling $\boldsymbol{t}$. The probability that t_{ji} takes on a previously used value t or new value t^{new} is given as follows;

$$p(\boldsymbol{x}_{ji}|\boldsymbol{t}^{-ji}, t_{ji} = t^{\text{new}}, \boldsymbol{k}) = \sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} f_k^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) + \frac{\gamma}{m_{\cdot\cdot} + \gamma} f_{k^{\text{new}}}^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) \quad (8)$$

$$f_k^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) = \int \prod_{s=1}^{S} F_s(x_{ji}^s|\phi_k) h(\phi_k) d\phi_k \quad (9)$$

$$f_{k^{\text{new}}}^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) = \int \prod_{s=1}^{S} F_s(x_{ji}^s|\phi_{k^{\text{new}}}) h(\phi_{k^{\text{new}}}) d\phi_{k^{\text{new}}} \quad (10)$$

where $h(\phi_{k^{new}})$ is probability density function of the base probability measure H. The conditional distribution of t_{ji} is then obtained as follows;

$$p(t_{ji} = t|\boldsymbol{t}^{-ji}, \boldsymbol{k}) \propto \begin{cases} n_{jt\cdot}^{-ji} f_{k_{jt}}^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) & \text{if } t \text{ is previously used,} \\ \alpha_0 f_{k^{\text{new}}}^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) & \text{if } t = t^{\text{new}}. \end{cases} \quad (11)$$

If the sampled value of t_{ji} is t^{new}, we obtain a sample of $k_{jt^{\text{new}}}$ according to the following probability:

$$p(k_{jt^{\text{new}}} = k|\boldsymbol{t}, \boldsymbol{k}^{-jt^{\text{new}}}) \propto \begin{cases} m_{\cdot k} f_k^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) & \text{if } k \text{ is previously used,} \\ \gamma f_{k^{\text{new}}}^{-\boldsymbol{x}_{ji}}(\boldsymbol{x}_{ji}) & \text{if } k = k^{\text{new}}. \end{cases} \quad (12)$$

Sampling $\boldsymbol{k}$. Since k_{jt} actually changes the component membership of all data items in table t, the likelihood obtained by setting $k_{jt} = k$ is given by $f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt})$, so that the conditional probability of k_{jt} is obtained as follows;

$$p(k_{jt}=k|\boldsymbol{t},\boldsymbol{k}^{-jt}) \propto \begin{cases} m_{\cdot k}^{-jt} f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k \text{ is previously used,} \\ \gamma f_{k^{\text{new}}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k = k^{\text{new}}, \end{cases} \tag{13}$$

$$f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) = \prod_{i:t_{ji}=t} \int \prod_{s=1}^{S} F_s(x_{ji}^s|\phi_k) h(\phi_k) d\phi_k, \tag{14}$$

$$f_{k^{\text{new}}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) = \prod_{i:t_{ji}=t} \int \prod_{s=1}^{S} F_s(x_{ji}^s|\phi_{k^{\text{new}}}) h(\phi_{k^{\text{new}}}) d\phi_{k^{\text{new}}}. \tag{15}$$

In recognition, topics of each $\boldsymbol{x}_{ji}$ are calculated using Gibbs sampling with $F(\theta)$ obtained in learning. Given a test video j' and words $\boldsymbol{x}_{j'i}$, the corresponding topic $\theta_{j'i}$ is computed and the topic histogram of test video j' is obtained as follows,

$$\text{hist}(\theta) += \theta_{j'i}, \tag{16}$$

$$k = \max(\text{hist}(\theta)). \tag{17}$$

An action label recognized for test video j' is the maximization topic k of the histogram.

6 Experiments

The proposed method was evaluated for Weizmann dataset which includes 10 motion classes such as jump, run, ship and walk. The total number of movies included in the database was 92. We employed leave-one-out cross validation as evaluation method. The following four experiments were conducted for the evaluation.

- Exp. 1: motion + LDA
  - LDA was evaluated using only motion word as a baseline method.
- Exp. 2: motion + HDP-LDA
  - HDP-LDA was evaluated using only motion word for comparison with Exp.1.
- Exp. 3: motion + loation + HDP-LDA
  - HDP-LDA was evaluated using motion word and location word to compare it with MI-HDP-LDA.
- Exp. 4: motion + loation + MI-HDP-LDA
  - The proposed method was evaluated using motion word and location word.

At first, LDA was evaluated using motion word as a baseline. Though LDA generates the prior distribution by using the Dirichlet distribution as well as HDP-LDA, it can not estimate the number of latent topics automatically as HDP can do.

In the motion word parameters, cuboid size was $15 \times 15 \times 15$ and codebook size was 1000. In the location word, the number of blocks in human region size was

10×13. The response parameter τ was set to 5 and PCA reduced the dimension to 779. The number of the latent topics was set to 10 manually for LDA.

As a result of the experiment 1, the recognition rate was 61.8% and the confusion matrix is shown in Fig.4(a). It recognized the motions of bend, jack, pjump and wave1 excellently, but the motions of jump, side and skip were considerably confused. The reason of the confusion will be attributed to their similar movements of the body except for the legs. The motions of wave1 and wave2 had been learned as the same action class by unsupervised LDA without location information, because the subject waves right hand only in the motion wave1 and waves both hands in the motion wave2. Therefore it classified them into the same action class in the recognition.

As a result of the experiment 2, the recognition rate was 64.9% and the confusion matrix is shown in Fig.4(b). The average number of classes automatically estimated was 16.33. It recognized bend and jack, etc. excellently, but the motion of Jump and wave2 was confused as experiment 1.

As a result of the experiment 3, the recognition rate was 64.0% and the confusion matrix is shown in Fig.4(c). The average number of classes estimated was 14.33. This experiment was carried out using both the motion words and the location words for HDP-LDA. In this experiment, it was assumed that the both words were not concurrently occurred but were independently generated, and the word of the location information was simply added. This experiment was carried out to compare with MI-HDP-LDA in terms of information concurrency. The same likelihood $F_s(\theta)$ was used for the same word in HDP-LDA and MI-HDP-LDA.

Finally, experiment 4 was carried out for MI-HDP-LDA using both the motion words and the location words. The recognition rate was improved up to the highest score 73.7%. The average number of topics estimated was 15.78. The confusion matrix is shown in Fig.4(d). Especially, in unsupervised MI-HDP-LDA, wave1 and wave2 were automatically learned as different motion owing to location information of the motion, therefore they were classified into different motion classes in the recognition.

Next, the number of latent topics estimated by HDP is described. There were about 15 latent topics at average with some variation over the experiment 2, 3 and 4. Fig.3 shows the number of topics sampled in experiment 4.

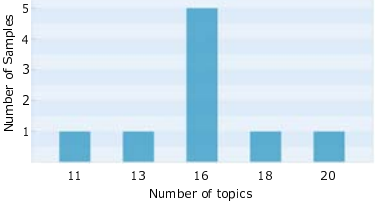


Fig. 3. Number of topics in experiment 4 (MI-HDP-LDA)D

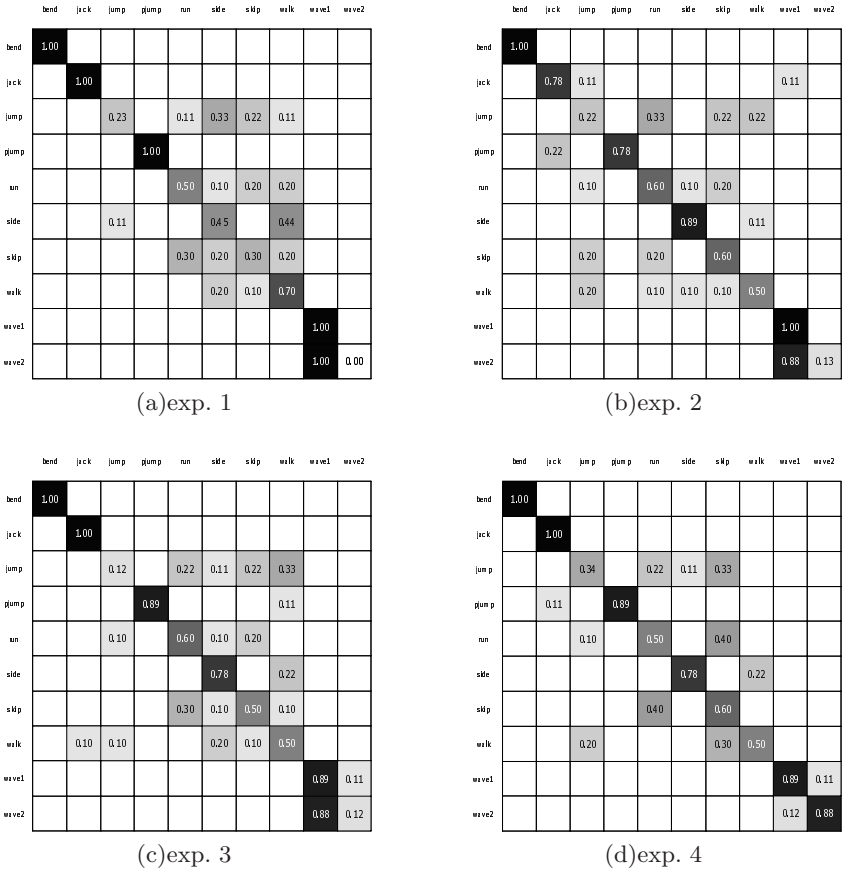


Fig. 4. Confusion matrices computed in respective experiment for Weizmann Dataset(%)

Weizmann Dataset has two kinds of motions in jump, run, side, skip and walk: motions toward right or left directions. If these actions are separately counted, the number of actions included in Weizmann Dataset becomes 15.

It can be confirmed that the extended number of actions 15 almost coincides with the number of latent topics estimated in the experiment. The number of latent topics is the number of mixtures with the highest likelihood and estimated through the experiment, depending on the training data.

7 Conclusion

In this paper, a new unsupervised learning method MI-HDP-LDA has been proposed to deal with motion feature and location information concurrently in the motion recognition task. This method can also estimate the number of

latent topics included in the training data automatically owing to the HDP (Hierarchical Dirichlet Processes).

In the experiments of motion learning and recognition for Weismann Dataset, LDA showed 61.8% recognition rate using only motion information. The proposed MI-HDP-LDA achieved 73.7% recognition rate, resulting in 11.9 points improvement.

Future work will be the incorporation of the various information such as "what" in addition to "where" and "how" which this paper pays attention to. Pose information of the limbs will be also important in learning and recognizing of the motion.

References

1. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3), 257–267 (2001)
2. Grundmann, M., Meier, F., Essa, I.: 3d shape context and distance transform for action recognition. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
3. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: IEEE International Conference on Computer Vision, pp. 726–733 (2003)
4. Laptev, I., Lindeberg, T.: Space-time interest points. In: IEEE International Conference on Computer Vision, pp. 432–439 (2003)
5. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS, pp. 65–72 (2005)
6. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of international conference on Multimedia, pp. 357–360 (2007)
7. Niebles, J., Wang, H., Li, F.-F.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision 79(3), 299–318 (2008)
8. Wang, Y., Sabzmeydani, P., Mori, G.: Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In: Workshop on Human Motion 2007, pp. 240–254 (2007)
9. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: IEEE International Conference on Computer Vision, pp. 1395–1402 (2005)
10. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. Journal of the American Statistical Association 101(476), 1566–1581 (2006)
11. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
12. Ferguson, T.: A bayesian analysis of some nonparametric problems. Annals of Statistics 1(2), 209–230 (1973)

Detecting Spatiotemporal Structure Boundaries: Beyond Motion Discontinuities

Konstantinos G. Derpanis and Richard P. Wildes

Department of Computer Science and Engineering
York University
Toronto, Ontario, Canada
{kosta,wildes}@cse.yorku.ca

Abstract. The detection of motion boundaries has been and remains a longstanding challenge in computer vision. In this paper, the recovery of motion boundaries is recast in a broader scope, as focus is placed on the more general problem of detecting spacetime structure boundaries, where motion boundaries constitute a special case. This recasting allows uniform consideration of boundaries between a wider class of spacetime patterns than previously considered in the literature, both coherent motion as well as additional dynamic patterns. Examples of dynamic patterns beyond standard motion that are encompassed by the proposed approach include, flicker, transparency and various dynamic textures (e.g., scintillation). Toward this end, a novel representation and method for detecting these boundaries in raw image sequence data are presented. Central to the representation is the description of oriented spacetime structure in a distributed manner. Empirical evaluation of the proposed boundary detector on challenging natural imagery suggests its efficacy.

1 Introduction

The detection of motion boundaries in (temporal) image sequences has been and remains a longstanding challenge in computer vision. The reason for continued interest is due in part to their providing boundary conditions for any process that requires knowledge of the spacetime support of coherent data for recovery of reliable local estimates (e.g., optical flow). In addition, these boundaries provide useful information about the 3D structure of the imaged scene.

Although of obvious importance, motion represents a particular instance of the myriad spatiotemporal patterns encountered in image sequences. Examples of non-motion-related patterns of significance include, unstructured (e.g., "blank wall"), flicker (i.e., pure temporal intensity change), and dynamic texture (e.g., as typically associated with stochastic phenomena, such as windblown vegetation and turbulent water). These types of dynamic patterns have received far less attention than motion in the literature.

The goal of the present work is the development of a unified approach to detecting spacetime boundaries that is broadly applicable to the diverse phenomena encountered in the natural world, including but not limited to motion. It is proposed that the choice of representation is key to meeting this challenge: If the representation cannot adequately distinguish the patterns of interest, then the recovery of boundaries, regardless of the

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 301–312, 2010.

chosen detector, will fail. For present purposes, local 3D, (x, y, t), spacetime orientation will be shown to be of appropriate descriptive power. Measures of spatiotemporal orientation capture the first-order correlation structure of the data irrespective of its origin (e.g., irrespective of its physical cause), even while distinguishing a wide range of patterns of interest (e.g., different motions, as well as the various aforementioned additional dynamic patterns). With visual spacetime represented according to its local orientation structure, boundaries will be extracted via detection of spatiotemporal change in the local orientation structure.

Previous dynamic boundary detection methods can be categorized as either local or global. Local methods restrict analysis to limited neighbourhoods around each point. In contrast, global methods generally attempt to simultaneously estimate a consistent flow field and its discontinuities across the image.

Early efforts focused on the local detection of motion discontinuities in dense optical flow fields through the use of edge operators (e.g., [1]). Alternatively, regions exhibiting a high percentage of unmatched features on a frame-to-frame basis are identified as motion boundaries [2]. Other methods have detected boundaries from the shape of the local template match surface (e.g., [3]). Boundary detection also has been performed using a detector over basis flows for simple events (e.g., motion of occluding edge or bar) [4]. In follow-up work, motion discontinuity regions were captured using a non-linear generative model [5]. Alternatively, hand-labeled motion boundaries have been used to train a discriminative classifier [6]. Further, motion boundary detection has been based on analysis of local distributions of image features (e.g., intensity, colour, flow) [7,8]. Perhaps most closely related to the approach proposed here are methods that detect motion boundaries from the structure of spatiotemporal brightness patterns as captured by local estimates of spatiotemporal orientation [9] or, more generally, oriented bandpass filters [10,11]. Also related are previous efforts using oriented energy measurements for boundary detection in 2D intensity images, e.g., [12,13].

Typically, the focus of global methods has been the recovery of regional flows, with inter-region boundaries made explicit to various degrees [14,15]. The particular formulations developed in these cases are limited to motion boundary detection and not more generally applicable to additional classes of spatiotemporal structure boundaries. Alternatively, global methods have been developed that indicate regions of dynamic texture and their boundaries, e.g., [16]; however, it does not appear that such methods are applicable directly to motion boundaries.

Overall, it appears that no single previous method for spatiotemporal boundary detection is capable of capturing the wide range of juxtaposed spacetime patterns encountered in the real world. Furthermore, the emphasis of most previous work has been on the special case of motion boundaries.

In the light of previous research, the following three major contributions are made. (i) The problem of detecting motion boundaries is recast in terms of the more general problem of identifying spacetime structural boundaries. This recasting allows for capturing, in a unified manner, boundaries between a wide range of important spatiotemporal patterns (unstructured, static, motion, flicker, (pseudo-)transparency, translucency, scintillation). (ii) A new representation is proposed for identifying spatiotemporal boundaries that captures local 3D, (x, y, t), image spacetime orientation structure in a distributed

manner. The representation converts structure differences to spatiotemporal contrast; correspondingly, simple contrast detection mechanisms (e.g., local differential operators) can mark boundaries. (iii) The proposed boundary detector's ability to identify boundaries along meaningful structural lines is shown quantitatively and outperforms several extant approaches on a wide range of challenging natural imagery.

2 Technical Approach

The proposed approach to spacetime representation and boundary analysis consists of an initial local oriented decomposition of the input video, followed by detecting spacetime structural boundaries across the decomposition. This approach is motivated by the fact that such a decomposition captures significant, meaningful aspects of its temporal variation [11]. As examples: A significant response in a single component of the decomposition is indicative of motion; significant responses in multiple components of the decomposition are indicative of transparency-based superposition; more uniform, yet still significant responses across the entire decomposition are indicative of dynamic texture (e.g., scintillation); lack of response in any component of the decomposition is indicative of unstructured regions (e.g., uniform intensity). Under this representation, coherency of spacetime is defined in terms of consistent patterns across the decomposition, while inconsistencies indicate spacetime structural boundaries. Integration of purely spatial cues (e.g., colour and texture), although of obvious benefit, is beyond the scope of this contribution.

2.1 Spatiotemporal Oriented Energy Representation

The spacetime orientation decomposition is realized using broadly tuned 3D Gaussian second derivative filters, $G_{2_{\hat{\theta}}}(x, y, t)$, and their Hilbert transforms, $H_{2_{\hat{\theta}}}(x, y, t)$, with the unit vector $\hat{\theta}$ capturing the 3D direction of the filter symmetry axis. The responses are pointwise rectified (squared) and summed to yield the following energy measure,

$$E_{\hat{\theta}}(x, y, t) = (G_{2_{\hat{\theta}}} * I)^2 + (H_{2_{\hat{\theta}}} * I)^2, \tag{1}$$

where $I \equiv I(x, y, t)$ denotes the input imagery and $*$ convolution.

Each oriented energy measure, (1), is confounded with spatial orientation. Consequently, in cases where the spatial structure varies widely about an otherwise coherent dynamic region (e.g., single motion of a surface with varying spatial texture), the responses of the ensemble of oriented energies will reflect this behaviour and thereby support spurious region segregation. To ameliorate this difficulty, the spatial orientation component is discounted by "marginalization" of this attribute, as follows.

In general, a pattern exhibiting a single spacetime orientation (e.g., velocity) manifests itself as a plane through the origin in the frequency domain [17]. Correspondingly, summation across a set of x-y-t-oriented energy measurements consistent with a single frequency domain plane through the origin is indicative of energy along the associated spacetime orientation, independent of purely spatial orientation. Since Gaussian derivative filters of order $N = 2$ are used in the oriented filtering, (1), it is appropriate to

consider $N+1=3$ equally spaced directions along each frequency domain plane of interest, as $N+1$ directions are needed to span orientation in a plane with Gaussian derivative filters of order N [13]. Let each plane be parameterized in terms of its unit normal, $\hat{\mathbf{n}}$; a set of equally spaced $N+1$ directions within the plane are given as

$$\hat{\theta}_i = \cos\left(\frac{2\pi i}{N+1}\right)\hat{\theta}_a(\hat{\mathbf{n}}) + \sin\left(\frac{2\pi i}{N+1}\right)\hat{\theta}_b(\hat{\mathbf{n}}), \quad 0 \leq i \leq N, \tag{2}$$

with

$$\hat{\theta}_a(\hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \hat{\mathbf{e}}_x / \|\hat{\mathbf{n}} \times \hat{\mathbf{e}}_x\| \quad \text{and} \quad \hat{\theta}_b(\hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \hat{\theta}_a(\hat{\mathbf{n}}), \tag{3}$$

where $\hat{\mathbf{e}}_x$ denotes the unit vector along the ω_x-axis[1]. In the case where the spacetime orientation is defined by velocity (u_x, u_y), the normal vector is given by $\hat{\mathbf{n}} = (u_x, u_y, 1)^\top / \|(u_x, u_y, 1)^\top\|$.

Now, energy along a spacetime direction, $\hat{\mathbf{n}}$, with spatial orientation discounted through marginalization, is given by summation across the set of measurements, $E_{\hat{\theta}_i}$,

$$\tilde{E}_{\hat{\mathbf{n}}}(x,y,t) = \sum_{i=0}^{N} E_{\hat{\theta}_i}(x,y,t), \tag{4}$$

with $\hat{\theta}_i$ one of $N+1=3$ specified directions, (2), and each $E_{\hat{\theta}_i}$ calculated via the oriented energy filtering, (1), (cf. [18] where a similar formulation is developed, but only applied to image motion analysis and without inclusion of the $H_{2\theta}$, which provides phase independence). In the present implementation, six different spacetime orientations are made explicit, namely, leftward, rightward, upward and downward motion, static (no motion/orientation orthogonal to the image plane) and flicker/infinite motion (orientation orthogonal to the temporal axis); although, due to the broad tuning of the filters employed, responses arise to a range of orientations about the peak tunings.

Finally, the resulting energies in (4) are confounded by the local contrast of the signal and as a result increase monotonically with contrast. This makes it impossible to determine whether a high response for a particular spacetime orientation is indicative of its presence or is instead a low match that yields a high response due to significant contrast in the signal. To arrive at a purer measure of spacetime orientation, the energy measures are normalized by the sum of consort planar energy responses at each point,

$$\hat{E}_{\hat{\mathbf{n}}_i}(x,y,t) = \tilde{E}_{\hat{\mathbf{n}}_i}(x,y,t) / \left(\sum_{j=1}^{M} \tilde{E}_{\hat{\mathbf{n}}_j}(x,y,t) + \epsilon\right), \tag{5}$$

where M denotes the number of spacetime orientations considered, and ϵ a constant introduced as a noise floor and to avoid instabilities at points where the overall energy is small. Conceptually, (1) - (5) can be thought of as taking an image sequence, $I(x,y,t)$, and carving its (local) power spectrum into a set of planes, with each plane corresponding to a particular spacetime orientation, to provide a relative indication of the presence of structure along each plane.

[1] Depending on the spacetime orientation sought, $\hat{\mathbf{e}}_x$ can be replaced with another axis to avoid the case of an undefined normal vector.

The constructed representation enjoys a number of attributes that are worth emphasizing. (i) Owing to the bandpass nature of the Gaussian derivative filters (1), the representation is invariant to additive photometric bias. (ii) Owing to the normalization (5), the representation is invariant to absolute contrast in the input signal. (iii) Owing to the marginalization (4), the representation is invariant to changes in appearance manifest as spatial orientation variation. Overall, these three invariances result in robust boundary detection that is invariant to pattern changes that do not correspond to dynamic pattern variation, even while making explicit local orientation structure that arises with temporal variation (motion, flicker, scintillation, etc.). (iv) The representation is efficiently realized via linear (separable convolution, pointwise addition) and pointwise non-linear (squaring, division) operations [19].

2.2 Anisotropic Smoothing

Prior to attempting to mark loci of significant spatiotemporal boundaries in the oriented energy decomposition, it is appropriate to smooth the derived representation to suppress noise. For this purpose, an anisotropic smoothing is performed as it serves to attenuate noise while enhancing structural boundaries. In the current implementation, mean-shift is employed as the anisotropic smoothing operation [20]. To promote spatiotemporal coherence at the smoothing stage, the orientation feature-space is augmented with positional information in the form of spacetime coordinates, (x, y, t). Putting the above features together yields a 9D feature vector (six oriented energies plus three for spacetime location), per image point.

Conceptually, mean-shift regards the feature-space as an empirical distribution. Each feature-point is associated with a mode (local maximum) of the distribution and thereby all points associated with a particular mode share a common feature value. In its simplest formulation (i.e., based on the Epanechnikov kernel), the mean-shift property can be written as (see [20], for details)

$$\widehat{\nabla} f(\mathbf{x}_c) \propto \left(\underset{\mathbf{x}_i \in \mathbf{S}_{h,\mathbf{x}_c}}{\text{mean}} \{\mathbf{x}_i\} - \mathbf{x}_c \right), \tag{6}$$

where $f(\mathbf{x})$ denotes the underlying probability density function of a n-dimensional space, $\mathbf{x}$, $\{\mathbf{x}_i\}$ the given set of samples, and $\mathbf{S}_{h,\mathbf{x}_c}$ a n-dimensional hyper-ball with radius h (the so-called kernel density bandwidth) centered at $\mathbf{x}_c$. Repeated application of (6) converges to a local mode of the distribution. In the present case, modes arise as particular values across 9D spatiotemporal feature vectors, $\mathbf{x}$. The final smoothed energy representation is realized by assigning the converged oriented energy portion of the feature vectors to their respective initial spacetime positions.

2.3 Spatiotemporal Structure Boundaries

In essence, the oriented energy representation converts spacetime structure differences to intensity differences across its decomposition. Correspondingly, boundaries simply correspond to image loci exhibiting significant spatiotemporal contrast in the representation. Figure 1 illustrates this point. In the orientation decomposition, it is seen

that the foreground tree yields relatively large and small intensities in the "static" and "rightward" components (resp.); whereas, the moving background yields the opposite behaviour. Therefore, spatiotemporal change (i.e., contrast) in the decomposition is indicative of the boundary between the tree and background. More generally, the orientation decomposition is a multivalued image, with spatiotemporal contrast indicative of spacetime boundaries in the underlying data. Here, it is interesting to note the difference in the behaviour of flow estimates and the proposed distributed representation across boundaries. In the former, the results are unpredictable due to a total failure of its intrinsic assumptions (e.g., brightness conservation). In the latter, due to the considerable overlap in spacetime and orientation tuning of the filters, the representation changes *smoothly* across structure boundaries reflecting the shift of energies among channels.

To capture the spatiotemporal contrast in the (smoothed) oriented energy representation, (5), a generalized gradient formulation is employed, as it captures change in a uniform manner across the multiple components of the decomposition. Let $\hat{E}_k$ be the kth band of the oriented energy representation, (5), and $\xi_i = x, y, t$ for $i = 1, 2, 3$, resp., define the directions along which partial derivatives are taken, then the generalized gradient is a 3×3 matrix $\mathbf{S}$ where

$$\mathbf{S}_{ij} \equiv \sum_{k=1}^{n} (\partial \hat{E}_{\hat{\mathbf{n}}_k} / \partial \xi_i)(\partial \hat{E}_{\hat{\mathbf{n}}_k} / \partial \xi_j). \tag{7}$$

Notice that $\mathbf{S}$ amounts to the summation of the more standard *structure/gradient tensor* [21] of each energy band[2]. The eigenvector of (7) associated with the greatest eigenvalue, λ_1, denoted $\mathbf{e}_1$, points in the direction of greatest change in the feature-space. For multivalued images (i.e., $n > 1$), a boundary is not indicated simply by a large value for λ_1; instead, it must be large relative to the other eigenvalues of $\mathbf{S}$ [22]. Correspondingly, a normalized measure of spacetime structure boundary salience is employed in the present context

$$\text{boundary}_{\text{salience}} = (\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2 + \phi), \tag{8}$$

where $\lambda_1 > \lambda_2$ denote the two largest eigenvalues of $\mathbf{S}$ and ϕ is a constant introduced as a noise floor. High values of the boundary salience measure, (8), (i.e, values close to one), are indicative of the presence of a spacetime structure boundary. Boundary salience for the example in Fig. 1 is shown in its rightmost panel. Next, similar to the non-maximum suppression principle used in intensity-based edge detection [24], a candidate boundary point is defined as a point that achieves a maximum in boundary salience, (8), in the direction of the eigenvector $\mathbf{e}_1$, as follow,

$$\begin{cases} \dfrac{\partial \text{boundary}_{\text{saliency}}}{\partial \mathbf{e}_1} = 0 \\ \dfrac{\partial^2 \text{boundary}_{\text{saliency}}}{\partial \mathbf{e}_1^2} < 0 \end{cases}. \tag{9}$$

Finally, candidate loci having a saliency value greater than a certain threshold, τ, are marked as boundary points.

[2] Other adaptations of the generalized gradient to multiband image boundary detection include application to colour [22] and spatial texture [23].

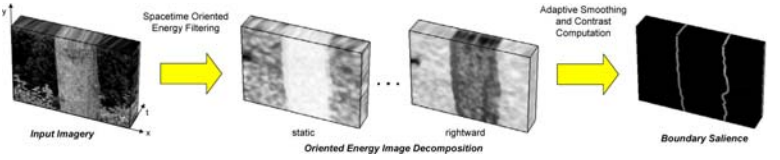


Fig. 1. Oriented energy decomposition maps structural differences to intensity differences. (left) Input image sequence of a foreground tree tracked (stabilized) by a moving camera with background in relative motion. (middle) Oriented energy decomposition of input shows marked differences in intensity corresponding to dynamic pattern differences of foreground vs. background. (right) Boundaries marked according to spatiotemporal contrast across the energy decomposition.

2.4 Algorithm

To recapitulate, the proposed approach can be given in algorithmic terms as follows.

Input: Greyscale image sequence
Input parameter: Boundary detection threshold, τ
Output: Binary image sequence marking spatiotemporal structure boundaries

Step 1: *Compute spacetime oriented energy representation (Section 2.1)*
1. Initialize 3D G_2/H_2 steerable basis.
2. Compute normalized spacetime oriented energy measure, Eqs. (1)-(5).

Step 2: *Anisotropic smoothing: Mean-shift (Section 2.2)*
1. Augment each normalized spacetime oriented energy measure, (5), with its spacetime coordinate (x, y, t).
2. Apply mean-shift smoothing iterations, (6).
3. Replace each energy measure in (5) with the final converged energy measure.

Step 3: *Compute spatiotemporal structure boundary salience (Section 2.3)*
for each spacetime point
1. Construct generalized gradient, (7), from (smoothed) oriented energy representation, (5).
2. Compute the eigenvector/eigenvalues of the generalized gradient, (7).
3. Compute boundary salience, (8).

Step 4: *Non-maximum suppression (Section 2.3)*
for each spacetime point
1. Apply non-maximum suppression, (9).
2. Retain candidate boundaries that have a saliency value, (8), greater than τ.

3 Empirical Evaluation

In evaluation, parameter settings for the proposed detector are as follows. The ϵ bias for contrast normalization, (5), empirically has been set to $\approx 1\%$ of the maximum expected response. The noise floor, ϕ, for boundary salience, (8), empirically has been set to $\phi = 0.01$. Mean-shift (anisotropic) smoothing includes three bandwidth parameters, h_{space},

h_{time} and h_{range}, which determine the resolution of detail along the spatial, temporal and range (here, spacetime orientation) dimensions, resp. Unless otherwise stated, the mean-shift bandwidths are set to: $h_{\text{space}} = 32$, $h_{\text{time}} = 10$, and $h_{\text{range}} = 0.12$.

Figure 4 shows a set of challenging natural image sequences containing a broad range of juxtaposed spacetime structures, including but not restricted to motion, and their boundary detection results (see caption for description of inputs). The challenging aspects of this data set include, regions that are unstructured, exhibit significant temporal aliasing due to fast motion, contain superimposed motion (transparency) and non-motion structure (e.g., flicker and scintillation). Coherent motion boundaries constitute a small fraction of the boundaries present in the data. Alternative available data sets are limited by their restricted focus on motion boundaries at the expense of more general spacetime structural boundaries [8]. The sequences presented here, consisting of juxtaposed natural and man-made structures, were obtained from a variety of sources: a Canon HF10 camcorder, the BBC documentary "Planet Earth" and the "BBC Motion Gallery" online video repository. Each sequence spans 10 frames. For each example, frame-by-frame hand-labeled ground truth was established. The identified boundaries in Fig. 4 provide compelling qualitative evidence that the proposed detector performs well on image sequences containing a wide variety of spacetime structures. This data set is available at `www.cse.yorku.ca/vision/research/spacetime-grouping`.

To quantify performance, results of the proposed detector are compared with the hand-labeled ground truth as well as alternative approaches. In particular, mean precision/recall scores [25] were calculated across all image sequences shown in Fig. 4 and are shown as tuning curves in Fig. 2 as detection parameters are varied. Here, over-partitioning is characterized in the curves by high recall but low precision, and the converse holds for under-partitioned image sequences.

The left panel of Fig. 2 shows several different curves for the proposed method, with each curve corresponding to a different value of the smoothing parameter, h_{range}; all curves are swept as the detection threshold varies from $0 - 1$. Matching between ground truth and identified boundary points was carried out using a distance threshold of eight, which is reasonable given that the support of the various compared detectors span approximately eight pixels. The consistently high recall indicates that ground truth boundaries are accurately marked. At the same time, a relatively high precision is attained, which indicates false boundaries are not prevalent. Further, the approach is seen to be stable with respect to variation of the smoothing parameter.

The right panel of Fig. 2 compares the best curve of the proposed approach, $h_{range} = 0.14$, with two alternative methods: (1) edge detection on dense optical flow fields [1] (implemented as a 3D Canny edge operator [24] applied to flow recovered using Lucas-Kanade [26]) and (2) the rank-based method that analyzes the gradient structure tensor over a neighbourhood [9]. These methods are selected for comparison as they are local (like the proposed method) and edge-detection in flow fields is a long standing approach, while the rank-based analysis is a recent proposal that has shown strong results for certain boundary types. Tuning curves were swept for the flow- and rank-based detectors by varying their detection thresholds from $0 - 10$ and $0 - 1$, resp. Curves for all three methods have the expected shape; however, flow- and rank-based are translated

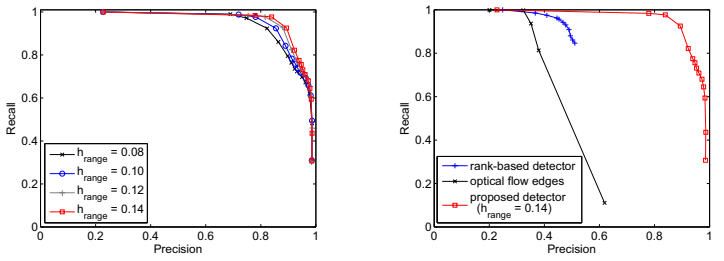


Fig. 2. Precision/recall curves. (left) Precision/recall of the proposed detector, each curve corresponds to a different setting of the range bandwidth used for smoothing. (right) Comparison of precision/recall with the proposed (optimal curve in (left)), flow- and rank-based detectors.

along the precision axis, which indicates significant over-partitioning relative to the proposed approach. Along these lines, rank-based outperforms flow, but is still noticeably worse than the proposed method.

To scrutinize the results in Fig. 2 further, Fig. 3 shows a comparison of the various boundary detectors on selected examples from Fig. 4 (c), (f) and (i). Also to compare against global methods, results from a recent level-set-based approach are shown [15]. Note that the global method must be supplied with a priori knowledge of the number of regions and hand initialization of its boundary[3]. For the motion parallax example, all of the alternative methods yield reasonable results. This is to be expected, as they are designed for motion boundaries. In the other two examples, the flow and rank methods yield spurious boundaries in the transparency and scintillation regions. This shortcoming arises from the inability of these methods to recover coherent measurements in non-coherent motion regions, as the assumption that coherency is well characterized by a single smoothly varying flow is violated. These spurious boundaries are the source of the low precision yet high recall rates indicated for the flow- and rank-based detectors in Fig. 2. For the transparency case using level-sets, the part of the initial contour that is outside the moving target evolves correctly; however, the part that started inside the moving region converges incorrectly, as the target interior does not conform to the method's assumption of a single smooth flow. In the scintillation case, the level-set collapses to a single region. Here, the failure is due to the relative lack of spatial structure in the ship interior, which allows the approach to fit a flow across the ship that is consistent with whatever flow it (erroneously) recovers for the scintillating water. The relative lack of structure in the ship interior also accounts for the apparent difference in performance of the flow and rank methods in such regions: Flow recovers highly variable vector fields that are interpreted as boundaries; whereas, unstructured regions are rank consistent and thus do not yield spurious boundaries. In contrast to the alternatives, the proposed detector naturally handles all three cases highlighted in Fig. 3.

[3] Due to the dependence of the extracted boundaries on hand contour initialization, number of regions and various scaling parameters in the level-set approach, it is not customary to sweep precision/recall curves for level-sets; hence, only qualitative comparisons are provided here.

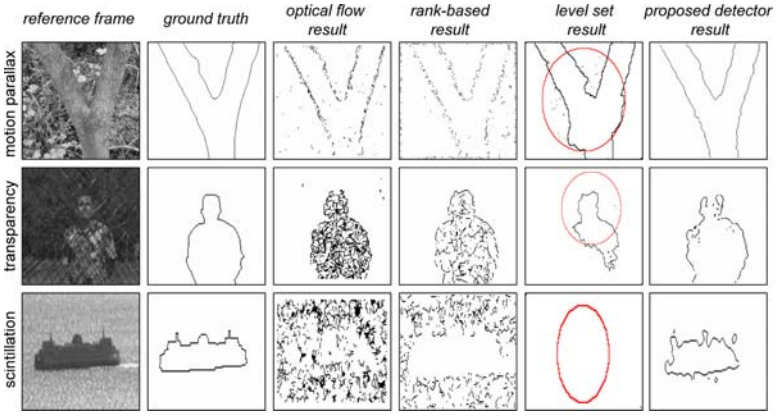


Fig. 3. Comparison of results for flow, rank, level-set and proposed approaches to boundary detection applied to Fig. 4 (c), (f) and (i). For level sets, elliptic contour shows hand initialized curve; other contour points are converged results. For the scintillation example the level set collapses to yield a single region.

4 Discussion and Summary

Most previous methods for spatiotemporal boundary detection are concerned with borders between regions of contrasting optical flow. Others are focused on dynamic textures. Improvements to these various methods might be realized via introduction of thresholds (e.g., confidence measures), multi-scale analyses (e.g., pyramid schemes for accommodating rapid motion), contour completion (cf. [9]), a more sophisticated flow estimator than considered here, etc. These approaches, however, fundamentally are limited by their underlying assumptions regarding the classes of visual phenomena that are to be encountered, which in turn limit their applicability to detecting a very circumscribed class of boundaries (e.g., motion). In comparison, it has been demonstrated that the proposed approach can naturally deal with the wide variety of real-world scenarios presented.

In summary, this paper has presented a unified approach to representing and detecting boundaries between a wide range of juxtaposed spacetime patterns (unstructured, static, motion, flicker, (pseudo-)transparency, translucency, scintillation). The approach is based on a distributed characterization of visual spacetime in terms of 3D, (x, y, t), spatiotemporal orientation, followed by application of a spatiotemporal differential operator (generalized gradient) to mark boundaries. Empirical evaluation on a wide variety of imagery demonstrates the proposed detector's ability to delineate boundaries between coherently structured regions.

Acknowledgments. This research was supported in part by NSERC and Precarn. F. Estrada provided software for computing precision/recall curves and helpful discussion.

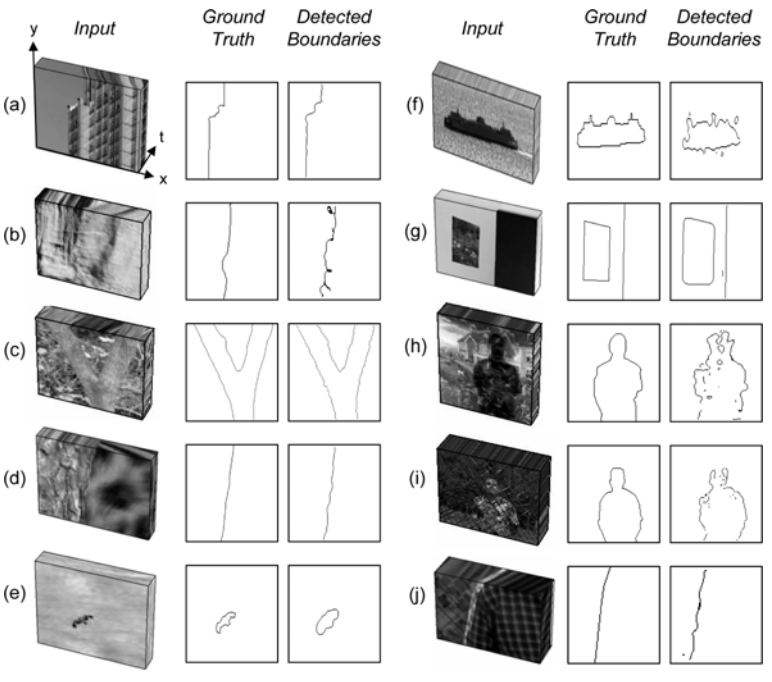


Fig. 4. Boundary detection results on a diverse and challenging set of natural imagery. In each example, the input sequence, a frame from the human-labeled ground truth and the boundary detection result, resp. are given. (a) A panning sequence consisting of a clear sky (i.e., unstructured) and a building (source: HF10). (b) Motion parallax sequence consisting of two mountain faces, where the foreground surface moves rapidly revealing a slower moving surface (source: "Planet Earth"). (c) Tree in foreground being coarsely stabilized by moving camera operator with resulting background motion (source: HF10). The background consisting of the ground plane is not fronto-parallel with respect to the camera, as a result the motion varies across the surface. (d) A leopard rapidly moving leftward behind a static tree (source: "Planet Earth"). (e) A flying bird crudely tracked by the camera operator to yield a slow moving target and a rapidly moving background (source: "Planet Earth"). (f) A ship moving over a scintillating water surface (source: "BBC Motion Gallery"). (g) A painting hanging on an unstructured wall with a light flickering in an adjacent hallway (source: HF10). (h) A translucency sequence realized by projecting (using an LCD projector) a walking person over a static painting (source: HF10). (i) A pseudo-transparency sequence consisting of a person walking behind a fence (source: HF10). (j) A juxtaposed motion and pseudo-transparency sequence consisting of two people moving rightward, one moving in front of a fence while the second is moving behind it (source: HF10). To view these videos, see supplemental material.

References

1. Thompson, W., Mutch, K., Berzins, V.: Dynamic occlusion analysis in optical flow fields. PAMI 7, 374–383 (1985)
2. Mutch, K., Thompson, W.: Analysis of accretion and deletion at boundaries in dynamic scenes. PAMI 7, 133–138 (1985)
3. Anandan, P.: Computing dense fields displacement with confidence measures in scenes containing occlusion. In: DARPA IUW, pp. 236–246 (1984)
4. Fleet, D., Black, M., Jepson, A.: Motion feature detection using steerable flow fields. In: CVPR, pp. 274–281 (1998)
5. Black, M.J., Fleet, D.J.: Probabilistic detection and tracking of motion boundaries. IJCV 38(3), 231–245 (2000)
6. Apostoloff, N., Fitzgibbon, A.: Learning spatiotemporal T-junctions for occlusion detection. In: CVPR, pp. II: 553–559 (2005)
7. Spoerri, A., Ullman, S.: The early detection of motion boundaries. In: ICCV, pp. 209–218 (1987)
8. Stein, A.N., Hebert, M.: Local detection of occlusion boundaries in video. IVC 27, 514–522 (2009)
9. Feldman, D., Weinshall, D.: Motion segmentation and depth ordering using an occlusion detector. PAMI 30, 1171–1185 (2008)
10. Niyogi, S.: Detecting kinetic occlusion. In: ICCV, pp. 1044–1049 (1995)
11. Wildes, R., Bergen, J.: Qualitative spatiotemporal analysis using an oriented energy representation. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 768–784. Springer, Heidelberg (2000)
12. Morrone, M.C., Owens, R.A.: Feature detection from local energy. PRL 6, 303–313 (1987)
13. Freeman, W., Adelson, E.: The design and use of steerable filters. PAMI 13, 891–906 (1991)
14. Heitz, F., Bouthemy, P.: Multimodal estimation of discontinuous optical flow using Markov random fields. PAMI 15, 1217–1232 (1993)
15. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. IJCV 62, 249–265 (2005)
16. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: ICCV, pp. 1236–1242 (2003)
17. Watson, A., Ahumada Jr., A.: A look at motion in the frequency domain. In: Motion Workshop, pp. 1–10 (1983)
18. Simoncelli, E.: Distributed Analysis and Representation of Visual Motion. PhD thesis. MIT (1993)
19. Derpanis, K., Gryn, J.: Three-dimensional nth derivative of Gaussian separable steerable filters. In: ICIP, pp. III: 553–556 (2005)
20. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24, 603–619 (2002)
21. Jähne, B.: Digital Image Processing, 6th edn. Springer, Berlin (2005)
22. Sapiro, G., Ringach, D.L.: Anisotropic diffusion of multivalued images with applications to color filtering. T-IP 5, 1582–1586 (1996)
23. Rubner, Y., Tomasi, C.: Coalescing texture descriptors. In: ARPA IUW, pp. 927–936 (1996)
24. Canny, J.: A computational approach to edge detection. PAMI 8, 679–698 (1986)
25. Estrada, F., Jepson, A.: Benchmarking image segmentation algorithms. In: IJCV (2009) (to appear)
26. Lucas, B., Kanade, T.: An iterative registration technique with an application to stereo vision. In: IJCAI, pp. 674–679 (1981)

An Accelerated Human Motion Tracking System Based on Voxel Reconstruction under Complex Environments

Junchi Yan, Yin Li, Enliang Zheng, and Yuncai Liu

Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China
{yanesta,happyharry,bobzheng1983,whomliu}@sjtu.edu.cn
http://www.springer.com/lncs

Abstract. In this paper, we propose an automated and markless human motion tracking system, including voxel acquisition and motion tracking. We first explore the problem of voxel reconstruction under a complex environment. Specifically, the procedure of the voxel acquisition is conducted under cluttered background, which makes the high quality silhouette unavailable. An accelerated Bayesian sensor fusion framework combining the information of pixel and super-pixel is adopted to calculate the probability of voxel occupancy, which is achieved by focusing the computation on the image region of interest. The evaluation of reconstruction result is given as well. After the acquisition of voxels, we adopt a hierarchical optimization strategy to solve the problem of human motion tracking in a high-dimensional space. Finally, the performance of our human motion tracking system is compared with the ground truth from a commercial marker motion capture. The experimental results show the proposed human motion tracking system works well under a complex environment.

1 Introduction

Tracking of the human body, also called motion capture or posture estimation, is a problem of estimating the parameters of the human body model (such as joint angles) from the video data as the position and configuration of the tracked body change over time [1].

In this paper, we present a markerless and automated system for motion capture under complex environments that includes both the voxel reconstruction and the motion tracking. While many reaserchers have taken the approach of working directly with the image data, our system reconstruct the 3D voxel for each frame as input to the model acquisition and tracking.

The preprocess of voxel reconstruction gains some merits against its computation price. First, the main problem in working with the image plane data is that different body parts appear in different sizes and may be occluded depending on the relative position of the body to the camera. Our approach leads to simple and robust algorithms that take advantage of the unique qualities of voxel data [1]. Second, our system is tested under complex environments, while most present image based methods

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 313–324, 2010.

have not been proven good performance under complex environments. In the tracking stage, different from tracking human motion within the Bayesian framework [15-18], we adopt a hierarchical optimization approach to accelerate the tracking procedure.

Great efforts have been made on 3D voxel reconstruction and its use in human pose estimation. The first real-time human tracking system was made by Cheung et al [3]. Traditional silhouette-based 3D reconstruction [3, 4, etc.] gain advantages in its simplicity and computational efficiency. [5, 6] used graph cuts to minimize formulation of the voxel occupancy problem. Recently many works [7, 8, etc.] resort to the photo-consistency information for 3D models reconstruction.

Tracking methods based on voxel data are usually classified into two categories. One kind is based on model-free [9], which is no prior kinematic model is predefined. In most work, model-based approaches are used. There are two kinds of skeletal model applied. One is to use geometric primitives to represent human body parts [10-12]. The other methods use "stick Figure" to represent body parts and the dimension is ignored [13, 14]. Model-based tracking leads to the increasing of subparts of the human model invariably incur high dimensionality and make tracking a difficult task. To solve this problem, many approaches [15-18] have been investigated.

Our system flowchart is shown in figure 1. The components are 3D voxel reconstruction, model initialization and motion tracking. Under complex environments, our 3D voxel reconstruction adopts a Bayesian framework incorporating segmentation on pixel and region level. Model initialization works under some particular gesture but feasible in practice, e.g. standing, jogging, and walking. Finally, 26 motion parameters are searched in our tracking stage by a hierarchical optimization procedure.

The contributions of this paper lie in two respects. (1) A markerless and automated human motion tracking system is provided, combining both the reconstruction and the tracking procedures. (2) Great acceleration and more robustness are achieved in our tracking system under complex environments compared to some previous work [2, 19]. The acceleration is achieved in both reconstruction and tracking procedures. First, we get a more robust reconstruction result while speed up the reconstruction. Second, an hierarchical searching strategy is designed to solve the tracking problem.

This paper is organized as follows: Section 2 presents the accelerated Bayesian framework for voxel occupancy inference and reconstruction. Section 3 refers to our hierarchical optimization algorithm to solve human motion tracking using voxel data. Section 4 presents the experimental results and evaluation. Finally, the conclusion and future work are presented in Section 5.

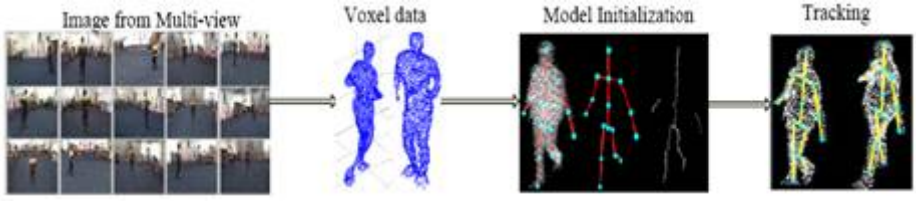


Fig. 1. The system flowchart

2 Accelerated Robust Reconstruction of Human Voxels

Our experiments are carried out under a complex environment, a fast while robust reconstruction is preferred. This section introduces the way of classifying foreground and background by fusing the information at pixel and region levels to calculate the occupancy probability of each voxel, the reconstruction comes from a binarization to the voxel's occupancy probability. Some symbols used in this paper are introduced.

Direct variables:

- I_r: denotes the image captured by camera r, r=1, 2…n.
- S_i: denotes one space grid (a voxel) occupancy state, i=1, 2, …m.
- B_r: denotes the image background captured by camera r.
- I_r^p: denotes the color feature vector of pixel p in image r.
- b_r^p: denotes the pixel p in image r belongs to background or not.
- *Threshold*: denotes the voxel's classification occupancy probability threshold.

Hidden variables:

- O_r^p: denotes whether there exits some objects occluding the voxel i on the straight line connecting voxel i and the center of camera r.
- $Dect_r$: denotes the foreground pixel detection rate from the image r.

Different from traditional shape-from-silhouette algorithm, in our algorithm, we take the voxel reconstruction process as a multi-view information fusion procedure to infer the occupancy probability of each voxel.

2.1 The Fusion Framework Using Multiple Camera Information on the Pixel Level

The volume of interest is subdivided into *m* voxels with equal size. Since the two states of each voxel are exclusive and exhaustive, $P(S_i=1)+P(S_i=0)=1$. The purpose in this section is to find the posterior probability of occupancy status of each voxel from observations of all the *n* images: $P(S_i=1|\{I\}_n)$, $\{I\}_n=\{I_1,I_2\ldots I_n\}$. The probability of occupancy is updated once a new image is available. Given the current estimation of a voxel *i* after observing *r*-1 images $\{I\}_{r-1}=\{I_1,I_2\ldots I_{r-1}\}$ and a new observation of I_r, estimation of the voxel's occupancy can be updated by Bayesian theory:

$$P(S_i=1|\{I\}_r)=\frac{P(\{I\}_r|S_i=1)P(S_i=1)}{P(\{I\}_r|S_i=1)P(S_i=1)+P(\{I\}_r|S_i=0)P(S_i=0)} \tag{1}$$

$$P(\{I\}_r|S=1)=P(\{I\}_{r-1}|S=1)P(I_r|S=1)=P(I_1|S=1)P(I_2|S=1)\ldots P(I_r|S=1) \tag{2}$$

Then the problem is reduced to solve $P(\{I\}_r|S_i)$. We adopt the framework first proposed by Franco [20]:

$$\begin{aligned}P(I_r^p|S_i)&=\sum_{O_r^p}P(I_r^p|O_r^p,S_i)P(O_r^p)\\&=\sum_{O_r^p}\Big(\sum_{Dect_r}P(I_r^p|Dect_p)P(Dect_p|O_r^p,S_i)\Big)P(O_r^p)\end{aligned} \tag{3}$$

As for $P(O_r^p)$, for simplicity, we ignore the inherent spatial relationship among the foreground voxels. Thus, we assume $P(O_r^p=1)=P(O_r^p=0)=1/2$, which is regarded as a uniform distribution. See equation 4, the four hidden variable related terms are discussed as follows.

$$\begin{aligned} &P\left(Dect_r=1 \mid O_r^p=0, S_i=0\right)= PFA_r \\ &P\left(Dect_r=1 \mid O_r^p=1, S_i=0\right)= P\left(Dect_r=1 \mid O_r^p=0, S_i=1\right)=\mathrm{P}\left(\mathrm{Dect}_r=1 \mid \mathrm{O}_r^p=1, \mathrm{S}_i=1\right)= \mathrm{PD}_r \end{aligned} \tag{4}$$

The term $P(Dect_r=1|O_r^p=0,S_i=0)$ denotes that a false detection is reported while no foreground object exists at all. $\mathrm{PFA_r}$ models the false alarm rate. Other three terms indicate that the detection report is correct, no matter whether the object on the line connecting camera center and voxel i lies in front of or behind the voxel i. In our experiments, we set $PFA_r=0.95$, $PD_r=0.05$(see equation 4). Up till now, our reconstruction problem is reduced to represent $P(I_r^p|Dect_r=0)$. Since this term indicates $\mathrm{I_r^p}$ probably should be a background pixel, and we notice the fact that the background is ambiguous with foreground. Thus if we conduct classification on pixel level, the result is vulnerable to the color ambiguity between foreground and background (see figure 2 (a)).Section 2.2 will focus on how to model $P(I_r^p|Dect_r=0)$ both on the region and pixel level.

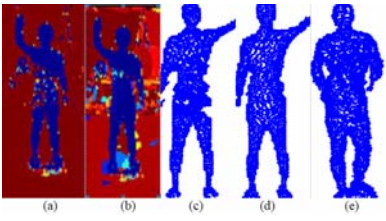


Fig. 2. (a): The inference result of single-pixel level Bayesian classification (notice there are many holes in the body) (b) the inference result of Bayesian classification on region and pixel level (the holes have been filled) (c): voxel surface got by Bayesian classification on pixel level (d), (e): voxel surface got by Bayesian classification on region and pixel level

2.2 Combine Pixel and Super-Pixel Classification under Complex Environment

The posterior probability of I_r^p representing background can be calculated by Bayesian theory :

$$P(b_r^p=1 \mid I_r^p)=\frac{P(I_r^p \mid b_r^p=1) P(b_r^p=1)}{\sum_{b_r^p} P(I_r^p \mid b_r^p) P(b_r^p)} \tag{5}$$

We model $P(I_r^p|b_r^p=1) \sim N(I_r^p|\mu_r^p,\sigma_r^p)$, a Gaussian distribution. Also, we assume the foreground pixel's color feature obeys uniform distribution that $P(I_r^p|b_r^p=0) \sim U(I_r^p)$. If we use equation 5 to represent $P(I_r^p|Dect_r=0)$, the result is not desirable as shown in figure 2(a). There tends to be some holes on the surface of reconstructed voxels, the main problem is that the environment is complex resulting that usually the foreground

is misclassified as background. To solve this problem, we propose the method combining single-pixel and super-pixel, which proves to be more robust to image color noise and ambiguity.

To obtain super-pixel, we choose the mean shift out of some excellent image segmentation algorithms [21-23] for two reasons. First, mean shift provides discontinuity preserving smoothness, which eliminates the image sensor noise, hence ensuring the correct segmentation. Second, it gains much computational efficiency compared to normalized cut [21]. This is important because the number of multi-view images at one time instant is often large. Once the segmented super-pixel is attained, we model the probability for each pixel that belongs to background, and apply it to represent $P(I_r^p|Dect_r=0)$.

$$P(I_r^p \mid Dect_r = 0) = \min(P(b_r^p = 1 \mid I_r^p), dist(R_r, R_b)) \tag{6}$$

$$dist(R_r, R_b) = \left(\frac{\sum_i f(R_r, i) f(R_b, i)}{\left(\sum_i f(R_r, i)^2 \sum_i f(R_b, i)^2\right)^{1/2}}\right)^{1/2} \tag{7}$$

R_r refers to the segmented area i.e. super-pixel where the single pixel I_r^p belongs in image r. R_b denotes the counterpart region in the background image. See equation 7, the distance is defined similar to a inner-product formation, which lies within [0,1]. Thus it can be used to model probability. There are two reasons for this definition. First, Bayesian classification will usually misclassify the foreground as background because of color ambiguities in our experiments; however, it does not tend to misclassify the background as foreground. Second, this formula reduces the impact of occasionally false segmentation around the boundary of foreground objects**.** $P(I_r^p|Dect_r=0)$ is demonstrated in figure 2 (b)(blue part indicates a low value of $P(I_r^p|Dect_r=0)$).

2.3 Reconstruction Acceleration by Focusing on Region of Interest

The computation cost of the aforementioned algorithm mainly comes from the stages of segmentation for super-pixels and calculation for the distance between two super-pixels. Thus, the reconstruction process can be significantly accelerated if the region of interest needing computation is downsized from the original size to a bounding rectangle that incorporates the region of interest tightly. Inspired by this idea, we fulfill the acceleration in this paper.

Although the foreground area can not be well classified in a single image; however, once we reconstruct the surface of 3-D human of current frame, whose projection in the image is a good prediction for the foreground in the next frame. In our experiments, we give the bounding rectangle 50 pixels abundance in the projection's each direction (left, right, top, bottom). By this way, a bounding rectangle is formed as the region of interest in the next frame. The segmentation and region distance are calculated only inside the bounding rectangle. Figure 3 demonstrates our Bayesian classification result ($min(P(b_r^p=1|I_r^p),dist(R_r,R_b))$).

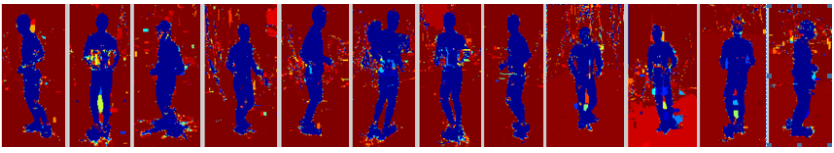


Fig. 3. Classification result in bounding rectangle (region of interest), from 12 views. (Notice the image size has been normalized for display, real size is shown in Table 1 in Section 4.1.2).

3 Human Motion Tracking Using Hierarchical Optimization

Human motion tracking using voxel data consists of two stages: initialization and tracking. In the initialization stage, body part sizes and their locations in the beginning frame of the sequence are estimated. In the tracking stage, based on the acquired parameters from initialization, the tracker updates the model position and configuration to reflect the motion of the tracked person for every new frame.

3.1 Skeleton Model and Automated Tracking Initialization

Through our method proposed in section 2, we acquire the voxel data. Before starting tracking, an automated initialization is preferred, especially considering its practical application.

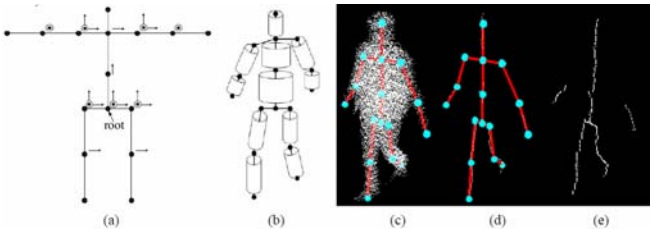


Fig. 4. (a): The articulated human body model and the rotation axes of each joint. (b): the human model fleshed out by 11 cylinders (c): the skeleton fitting voxels after initialization (d): the initialized skeleton (e): rough skeleton.

In our skeletal model, there are 11 joints having total 26 degrees of freedom (see figure 4 (a), (b)). The Euler angle of open joints, the length of each bone and the outer and inner radius for each cylinder can be automatically modulated to fit the human body in the initialization. For each joint, we use a presentation based on Euler angles and the articulation constraints are encoded.

At the beginning of tracking, we adopt the method in the previous work [24], which performs well under some conditions: standing, walking, jogging, making the initialization tractable. Once the initialization is finished, bone's length and cylinder's radius are determined. In the tracking process, they are assumed as constant leaving other 26 motion parameters to search. See figure 4(c), (d), (e) for the illustration of our automatic initialization.

3.2 Tracking by Hierarchical Optimization

After the process of model initialization, the motion and physical parameters of our model are known from the first frame. Considering the physical constraints of joints, the 26 motion parameters of the human body model is regarded as 26 variables which have certain ranges of allowed values. PEA algorithm proposed in [2] is used in our work to track the human body. However, we do not merely use its original form in our work. Our strategy is different from that previous work, rendering our algorithm more efficient. There are three substantial differences.

First, in order to make computation more efficient, only the surface voxels are concerned in our tracking process. We flesh out a skeletal model with two coaxial hollow cylinders for each bone. The inner cylinder's radius is a little smaller than the dimension of body parts, and the outer cylinder's radius a little bigger. The radius is determined in the initialization procedure.

Second, a hierarchical optimization strategy is proposed to decrease the computation cost to large extent. Intuitively, the optimization sequence is in the following order: torso with head, arms, and then legs. The torso with head is firstly optimized because the root of our skeletal model belongs to the torso, the father joints should be optimized earlier than the son joints.

Third, we define a new fitness function. Considering our algorithm is based on a hierarchical searching strategy, reliability of observation should be more concerned. We adopt a more reliable fitness function. For every stage, the fitness function is defined as: the number of surface voxels that fall between the pair of two hollow cylinders, minus the number of voxels that fall into the inside hollow cylinder.

$$fitness(\theta_{k1},\theta_{k2},...,\theta_{kn}) = \sum_{i=1}^{N} pos(V_i) - \sum_{i=1}^{N} neg(V_i) \tag{8}$$

V_i denotes the i-th voxel; N is the number of voxels. θ_{k1}, θ_{k2} ... θ_{kn} indicate the n parameters in the current searching stage k. $Pos(V_i)$ and $Neg(V_i)$ are defined as below:

$$pos(V_i) = \begin{cases} 1 & \text{if the i-th voxel fall between the pair of two hollow cylinders} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$neg(V_i) = \begin{cases} 1 & \text{if the i-th voxel fall into the inside cylinder} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The advantage of this fitness function is to avoid some cases when the arms is too near to the torso, and the observation that only focus on the voxel falling between the hollow cylinders would be unreliable.

4 Experimental Results and Quantitative Evaluation

Our experiments are conducted to evaluate the voxel reconstruction and tracking performance.

4.1 Reconstruction Results and Evaluation

In our experiments, 15 calibrated cameras (704*576, at 25Hz) share the same field of view of 2m*3m*3m. The volume of interest is divided into 200*300*200 voxels, each with the size of 1cm*1cm*1cm. The reconstructed voxels are generated by the probability threshold = 0.9 in our algorithm.

4.1.1 Reconstruction Results

Our reconstruction results are demonstrated in figure 5 (b), i.e. the 3D voxel surface. Compared with the results in figure 5 (a), whose surface is vulnerable to the holes caused by classification ambiguity on single-pixel level.

Fig. 5. Reconstruction results: (a): reconstructed surface on pixel level(Threshold = 0.9) (b): reconstructed surface on pixel and region level (Threshold = 0.9)

4.1.2 Reconstruction Performance Evaluation

Our evaluation concerns two respects, the reconstruction speed and reconstruction results. As for reconstruction speed performance, we test the acceleration performance in our experiments.

The original image's size is 576*704. Our experiments are carried on a 2.4GHz CPU, 2G memory PC, by unoptimized matlab code and with 15 views per frame. Our method focusing on the local region of interest accelerates the reconstruction to great extent. For detailed acceleration experimental comparison results, see table 1.

Table 1. Average computation time for one frame reconstruction

Projection	No Acceleration	Acceleration
Seconds	142.14s	26.59s

As for reconstruction result, we provide an evaluation method as well as the detailed results of our evaluation. The accurate ground truth is unavailable in practice. Thus, our evaluation is based on the comparison between the foreground segmented by manual-labeling as the ground truth and the image projection of 3D voxel surface. The evaluation offers us benefits at least in two respects. First, it leads to a quantitative analysis to our reconstructing results. Second, the evaluation result ensures us to adopt our acceleration strategy. The bounding rectangle covers the ground truth in abundance. See figure 6(a),(b),(c).

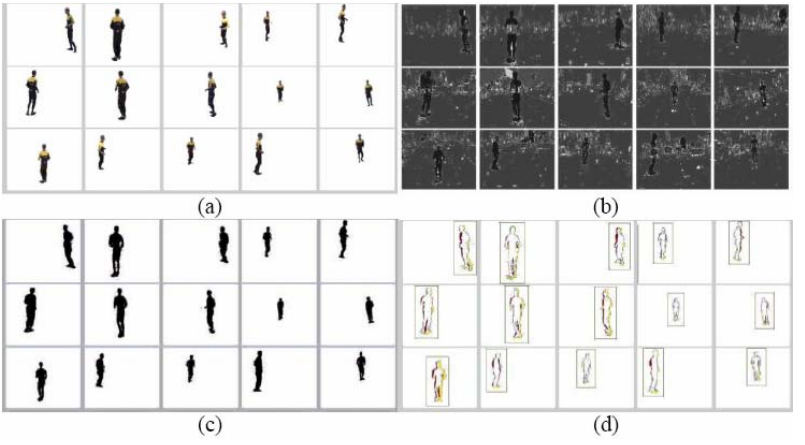


Fig. 6. (a): The ground truth (acquired by hand label from 15 views) (b) Bayesian inference classification results: min (P (b_r^P=1|I_r^P), dist(R_r,R_b)) (c) Projection on image plan from recon structed voxel surface (d) Comparison between projection and ground trut

Table 2 lists the reconstruction evaluation. The 1st column denotes the area of projection from reconstructed voxel surface to the camera image plane. The 2nd column indicates the area of the ground truth silhouette segmented by manual-label. The 3rd column means the area of intersection of surface's projection and ground truth. The 4th column displays the precision of projection-the rate of voxel whose projection falls into the ground truth silhouette. The 5th column shows the rate that the pixel of ground truth silhouette is matched by projection. Figure 6(d) illustrates the comparison between ground truth and our reconstruction's projection. Red and yellow indicate the part of projection and ground truth falling out of the intersection respectively.

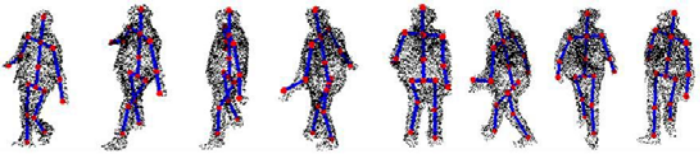


Fig. 7. Tracking results: some frames from a tracking sequence

Table 2. Reconstruction evaluation(Average of 15 Camera Views)

Projection	Ground truth	Intersection	Precision	Recall
20566	19626	17419	0.8465	0.8937

4.2 Tracking Results and Evaluation

We use our skeletal model to track the acquired voxels. After that, the tracking results are compared with the data from commercial marker motion capture device.

4.2.1 Tracking Result

Some tracking results for 3D voxel data are shown in figure 7. The average computation time for one frame of voxels is shown in table 3.

Table 3. Tracking algorithm performance. (second/frame)

Population	PEA	Hierarchical searching PEA
10	16.62	9.53
30	43.59	28.81

4.2.2 Motion Evaluation for Position and Angular Error

We quantitatively compare the motion parameters with that from commercial optical motion capture systems. We define the Motion parameters (Euler Angles) got from the multi-view cameras and motion capture systems as J_i, M_i respectively, i=1, 2, …m, m is the number of motion parameters. The angle error is computed as follows:

$$Error = \sum_{i=1}^{m} \left| J_i - M_i \right| / m \tag{11}$$

We define the position of joints got from the multi-view cameras as P_i, i=1, 2,…n, n is the number of joints. Also we define the positions of reflective markers on the human body as Q_i. The position error is computed as follows:

$$Error = \sum_{i=1}^{n} \left| P_i - Q_i \right| / n \tag{12}$$

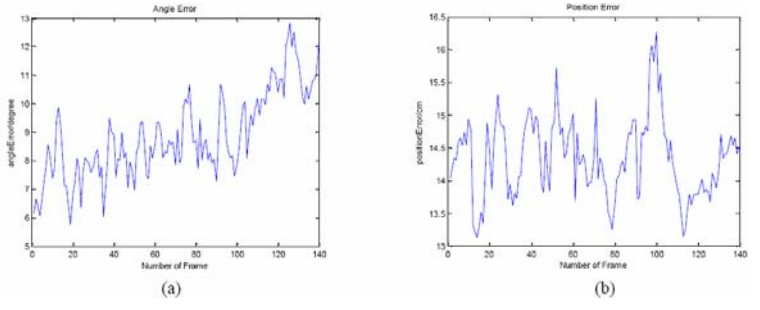


Fig. 8. (a): Result of Euler angles errors (b): Result of position errors

The final comparison for a jogging video stream between our system and the commercial motion capture is shown in figure 8.

Compared to the work [25], the error in our experiments is not negligible. One reason for the error is that the quality of our acquired voxel data is not so great as some previous work whose experiments are carried under well-controlled indoor environments, where perfect silhouette is available. Another reason is that our current model is an approximation to the true human body. In the figure 8(a), the angle error is growing as the tracking sequence goes longer while the position error can maintain stable over 140 frames. In our analysis, this is due to our observation during tracking is not well distinctive in regards of joint angle.

Apart from these aforementioned factors, some other factors should not be ignored. While we treat the commercial optic motion capture data as the "ground truth" it is worth noting that the "true" human motion is somewhat elusive. First, the synchronization between the motion capture and the video is estimated from data and likely has estimation errors that are difficult to quantify. Second, while the marker locations are on the body skin rather than on the bone, hence even the highest quality motion capture data can only provide approximation to the true limb locations.

5 Conclusions and Future Work

This paper proposes an automated and markless human motion tracking system which is robust to complex environment to some extent, and the whole process is accelerated significantly. Also a quantitative evaluation for reconstruction results is given too. Our tracking algorithm is based on a hierarchical optimization process with the help of an appropriate fitness function. An evaluation shows our tracking system is feasible.

There are some improvements needing to be done in the future work. First, although our algorithm shows some resistance to the complex indoor environments, voxel acquisition under outdoor and dynamic background is still not extensively and well investigated. Second, a more robust tracking initialization algorithm need be done since the present method can work only in limited conditions. Last but not least, based on human motion tracking, the human motion recognition can be studied.

Acknowledgements

This work is supported by the National Key Basic Research and development Plan of China (973) under Grant No. 2006CB303103, and also supported by National High Technology Research and development Plan of China (863) under Grant No. 2009AA01Z330 and the National Natural Science Foundation of China under Grant No. 60833009.

References

1. Ivana, M., Mohan, T., Edward, H., Cosman, P.: Human Body Model Acquisition and Tracking using Voxel Data. IJCV 53(3), 199–223 (2003)
2. Shen, S., Deng, H., Liu, Y.: Probability Evolutionary Algorithm based human motion tracking using voxel data. In: Proc. of CEC 2008, Hong Kong, China, pp. 44–49 (2008)
3. Cheung, G., Kanade, T., Bouguet, J., Holler, M.: A real time system for robust 3D voxel reconstruction of human motions. In: Proc. of CVPR 2000, Hilton Head, SC, USA (2000)
4. Szeliski, R.: Rapid octree construction from image sequences. CVGIP: Image Understanding 58(1), 23–32 (1993)
5. Snow, D., Viola, P., Zabih, R.: Exact voxel occupancy with graph cuts. In: Proc. of CVPR 2000, Hilton Head, SC, USA, vol. 1, pp. 345–352 (2000)
6. 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)

7. Vogiatzis, G., Torr, P.H.S., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: Proc. of CVPR 2005, San Diego, CA, USA, vol. 2, pp. 391–398 (2005)
8. Kutulakos, K., Seitz, S.: A theory of shape by space carving. IJCV 38(3), 199–218 (2000)
9. Chu, C.-W., Jenkins, O.C., Matarić, M.J.: Markerless kinematic model and motion capture from volume sequences. In: Proc. of CVPR 2003, Madison, Wisconsin, USA, vol. 2, pp. 475–482 (2003)
10. Drummond, T., Roberto, C.: Real-time tracking of highly articulated structures in the presence of noisy measurements. In: Proc. of ICCV 2001, Vancouver, pp. 315–320 (2001)
11. Delamarre, Q., Faugeras, O.: 3D articulated models and multi-view tracking with physical forces. The special issue of the CVIU journal on modeling people 81(3), 328–357 (2001)
12. Quentin, D., Olivier, D.F.: 3D Articulated Model and Multi-View Tracking with Silhouettes. In: Proc. of ICCV 1999, Kerkyra, Greece, pp. 716–721 (1999)
13. Fabrice, C., Toby, H.: Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction. In: Proc. of BMVC 2004, Norwich, UK, vol. 2, pp. 597–606 (2004)
14. Clement, M., Edmond, B., Bruno, R.: 3D Skeleton-Based Body Pose Recovery. In: Proc. of 3DPVT 2006, Chapel Hill, USA, pp. 389–396 (2006)
15. Gavrila, D., Davis, L.: 3-d model-based tracking of humans inaction: a multi-view approach. In: Proc. of CVPR 1996, San Francisco, CA, USA, pp. 73–80 (1996)
16. Deutscher, J., Davison, A., Reid, I.: Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In: Proc. of CVPR 2001, Kauai, HI, USA, vol. 2, pp. 669–676 (2001)
17. Wu, Y., Hua, G., Yu, T.: Tracking articulated body by dynamic markov network. In: Proc. of ICCV 2003, Nice, France, vol. 2, pp. 1094–1101 (2003)
18. Zhao, T., Nevatia, R.: Bayesian human segmentation in crowd situations. In: Proc. of CVPR 2003, Madison, WI, USA, vol. 2, pp. 459–466 (2003)
19. Zheng, E., Chen, Q., Yang, X., Liu, Y.: Robust 3D Modeling From Silhouette Cues. In: Proc. of ICASSP 2009, Taipei, Taiwan, China, pp. 1265–1268 (2009)
20. Franco, J., Boyer, E.: Fusion of multi-View silhouette cues using a space occupancy grid. In: Proc. of ICCV 2005, Beijing, China, vol. 1, pp. 1747–1754 (2005)
21. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. PAMI 24(5), 603–619 (2002)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI 22(8), 888–905 (2000)
23. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Trans. PAMI 19(4), 302–313 (1997)
24. Zheng, E., Zhao, X., Shen, S., Chen, Q., Liu, Y.: 3D Human Body Model Initialization and Tracking Based On Multi-View Cameras. In: Proc. of SCIS&ISIS, Nagoya, Japan (2008)
25. Balan, A.O., Sigal, L., Black, M.J.: A Quantitative Evaluation of Video-based 3D Person Tracking. In: Proc. of ICCCN 2005, San Diego, CA, USA, pp. 349–356 (2005)
26. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evolutionary Computation. 6(6), 580–593 (2002)

Automated Center of Radial Distortion Estimation, Using Active Targets

Hamed Rezazadegan Tavakoli[1] and Hamid Reza Pourreza[2]

[1] Young Researchers Club, Islamic Azad University, Mashhad Branch, Mashhad, Iran
hrtavakoli@aol.com
[2] Department of Computer Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, 91775-1111, Iran
hpourreza@um.ac.ir

Abstract. In this paper an automated center of radial distortion estimation algorithm is explained. The method applied to the development of an autonomous camera calibration algorithm. The idea of active targets, which are controlled by calibration algorithm is the key to the autonomy in this work.

The proposed method decouples the center of radial distortion from other calibration parameters. It is shown that the proposed method approximates the center of radial distortion correctly. Also it helps to the accuracy of calibration framework.

1 Introduction

Lens distortion is part of calibration process. The research on lens distortion can be traced back to 1919, when decentering distortion was introduced [1]. Decentering distortion consists of radial and tangential components. However, different distortions can be present, tangential, radial, and thin-prism. Each distortion is described using a model.

There has been lots of research on different distortion models. In fact, there are various assumptions about the lens model in calibration process.Tsai[2] assumes only radial distortion is present. Weng et al.[3] assumes the presence of radial, decentering, and thin-prism. There are also methods that assume no distortion[4]. It has been shown that the first coefficient of radial distortion is enough for most of industrial applications[2]. As a matter of fact, the research has been focused on radial distortion model, and different models has been proposed for the case of radial distortion such as polynomial models, rational model [5] and FOV model[6]. The simple polynomial model[3] is the most popular model. However, there exist extensions to the polynomial model, such as division model[7], cubic rational polynomial model[8] and rational polynomial model[9].

A radial distortion model with known center of distortion is equal to decentering model as there is no need for worrying about tangential distortion[10]. In comparison to decentering model, there are fewer parameters to estimate. Also

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 325–334, 2010.

the model is more complete than the radial distortion model because of considering tangential displacements. In consequence radial distortion with known distortion center is more accurate than radial and decentering models.

In most of radial distortion models the center of image[11] is considered to be the center of radial distortion. However, it is possible to estimate the actual center. In order to estimate the center of radial distortion, it is possible to initialize center of radial distortion to the center of image and later optimize these values with other camera parameters obtained in the camera calibration process, but it can result in non-optimum results. Avoiding non-optimal result, Devernay and Faugeras[6] suggested optimization of radial distortion coefficients first and then extending the optimization to all of the parameters including center of distortion. Tardif et al.[12] provided a new constraint optimization criteria which eliminates the risk of non-optimum result. Hartley and Kang[13] introduced a method that can estimate the center of radial distortion with the use of fundamental matrix.

In this article a method of center of radial distortion estimation is introduced. The proposed method decouples the center of distortion from other parameters. The proposed method is rooted in the active target idealogy. It would be shown that the method estimates the center of distortion accurately.

In the next section active target is explained. the third section is devoted to the center of radial distortion estimation algorithm. Section four explains the experiments followed by the conclusion.

2 Active Target

Active target concept can be confused by active calibration. The key difference is the interaction style. Active camera calibration mechanisms interact with the environment by camera movements[14], and have gained attention in the field of robot vision; such algorithms' examples could be found in[15,16].

All the methods of calibration, such as Tsai[2], Weng et al.[3], Zhang[17], and Heikkila[18] where active camera is not present can be categorized as passive calibration methods. None of them has interaction with the calibration environment.

It is possible to have an active calibration algorithm while the camera is not active; and is fixed on a tripod. The idea of such an active calibration algorithm, is that the information gained from each frame could be used to signal the calibration target for the next frame. This requires the calibration target to be active and controllable by the calibration algorithm. The term active calibration could be used in term of both methods. Meanwhile, the two are totally different.

The active target approach was implemented using a pattern generator program and a LCD monitor which was responsible for screening generated patterns. These components plus the automatic image acquisition and feature extraction provided the maximum flexibility and accuracy needed for having an active target.

3 Center of Radial Distortion Estimation

In this section the basis of radial distortion center estimation is explained. At first the method of Hartley and Kang[13] is explained. Afterwards, the proposed method is introduced. The two methods are similar on the aspect of decoupling the center of radial distortion.

3.1 Hartley's Method

This method utilizes fundamental matrix for approximating radial distortion center. The idea behind fundamental matrix is that a point considered projected to the image plane using an ideal non-distorted camera becomes distorted by expansioning away from a center of distortion. The expansion can be compared with the forward movement of a camera towards a scene. In such a movement points undergo a radial distortion. In this case the center of expansion, epipole, is the same with the center of radial distortion. The center of distortion is estimated by computing the fundamental matrix[19] relating known coordinates of points in the scene and the corresponding points in the distorted image.

$$x_d F X = 0 \ . \tag{1}$$

where, X is the point coordinate in the scene; x_d is the distorted corresponding image point; F is the fundamental matrix. The center of distortion ,the left epipole, could be computed using (2).

$$F^T e = 0 \ . \tag{2}$$

where, e is the left epipole.

The main disadvantage of this technique is that if no distortion is present or the amount of distortion is small, fundamental matrix computation would not be stable and the value of epipole is meaningless.

3.2 Active Center of Distortion Estimation

Active estimation of radial distortion center is referred to the estimation of distortion center using active calibration techniques. Considering the polynomial radial distortion model, it is inferred that radial distortion is symmetric. It is also known that distortion center in optical space is the center of lens. However, because of manufacturing displacement of sensor, mechanical parts, and optical system of a camera; optical center would hardly lie on the center of sensor. As a result the imaged center of distortion would not be the center of image. This makes the search for distortion center vital.

Some properties of lens and radial distortion are self-evident. One of those properties that could be used to find the distortion center is the relationship of line's straightness and center of distortion; stated in Postulate 1.

Postulate 1. *Under presence of radial distortion a straight line is straight if and only if it passes through the distortion center.*

Postulate1 originates from radial distortion nature. As the points of a line passing through the distortion center are in radial alignment on a line, the line straightness is not affected by radial distortion. This property is the basis of Theorem1 which is used to find the distortion center.

Theorem 1. *Under radial distortion, two concurrent lines l_1, l_2 would stay straight if and only if the intersecting point p is positioned on the distortion center o.*

Proof. If the intersecting point p is positioned on o, then the two lines both are passing through the distortion center and are straight as stated in Postulate1. Now consider the situation where l_1 and l_2 are both straight. From Postulate1 could be inferred the both lines are passing through o, and the only point the two lines have in common is p; which means p lies on o. □

Now it would be possible to use two concurrent line segments to find the distortion center, as shown in Fig. 1. If the straight lines $\overline{a_i c_i}$ and $\overline{b_i d_i}$ intersect point p_i lies on o, the distortion center is found. A simple search algorithm is proposed for finding distortion center; the aim of search method is minimization of p_i and o deviation by moving the calibration target in front of camera.

The main advantage of this technique is that it can even work under small amount of distortion. In case of no radial distortion, the deviation would not change that much trough movements. Such a case could be identified by testing two different positioning and presence of identical deviations.

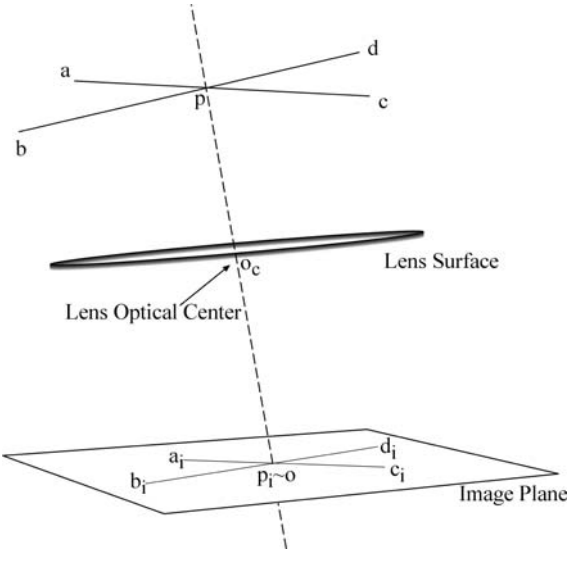


Fig. 1. Line segments and straightness property under radial distortion, the image of p, p_i, would lie on o if the imaginary line $\overline{pp_i}$ passes through optical center

3.3 Active Approach Implementation

An active target could be a *Light-emitting Diode* (LED) carried by a controlled robotic arm; or a board of LEDs, which switching them on and off shapes patterns. Approaches that rely on mechanical instruments is not versatile; flexible; precise and economical. The same is true for a board of LEDs. Another approach could be use of monitors for screening of patterns. In this case a simple *Cathode Ray Tube* (CRT) monitor would not be applicable because of its convex surface of screening area. However, a *Liquid Crystal Display* (LCD) is suitable.

A monitor depending on its setting can provide different precisions. As an example, a monitor with 1024×768 resolution; and 317mm×236mm viewable screen has pixels of approximately 0.31mm tall and 0.31mm wide; which means the pattern can have movements with precision of 0.31mm. It is obvious the precision would increase at higher resolutions.

A computer program can be used for generating different patterns and screening them on a LCD monitor. The main advantage of a monitor and a pattern generator program, is that patterns can be controlled and changed regarding the circumstances through the calibration process adaptively; having a fair accuracy. This approach also makes a fully automatic image acquisition phase possible.

Fig. 2 shows a calibration framework utilizing LCD and pattern generator program. The camera calibration framework consists of two major independent programs; one is the pattern generator and the other one is a program that performs all the computation, referred to as computational program. Both programs are in connection with each other using a communication channel. A communication center is in charge of transferring information and commands between these two programs. An interpreter is in charge of coding and decoding messages from numerical string into meaningful structures and vice versa.

The pattern generator consists of a graphic unit, and a pixel-metric convertor except the communication center. The graphic unit is in charge of displaying patterns. Patterns are generated by means of feature points. The type of pattern, and feature point is requested by the computational program. A pattern is imaged using multiple frames, where only one feature point is displayed on each frame. Pattern generator is capable of performing relative and absolute positioning of a pattern (e.g. request for relative movement of a pattern to the left by one centimeter). Computational program can get metric and pixel based information of monitor by requesting it from pixel-metric convertor unit.

Computational program consists of five components except the communication center. These components are image acquisition; feature extraction; geometrical lens distortion handler; camera parameter handler; and decision unit. Image acquisition is responsible for capturing frames. Geometrical lens distortion handler is responsible for finding distortion center and radial distortion coefficients. Camera parameter handler is responsible for approximation of internal parameters using undistorted images. The decision unit is in charge of these components. Decision unit decides on the information sent from pattern generator and decides where the information should be routed. It also handles the requests from different components and decides on the destination data should

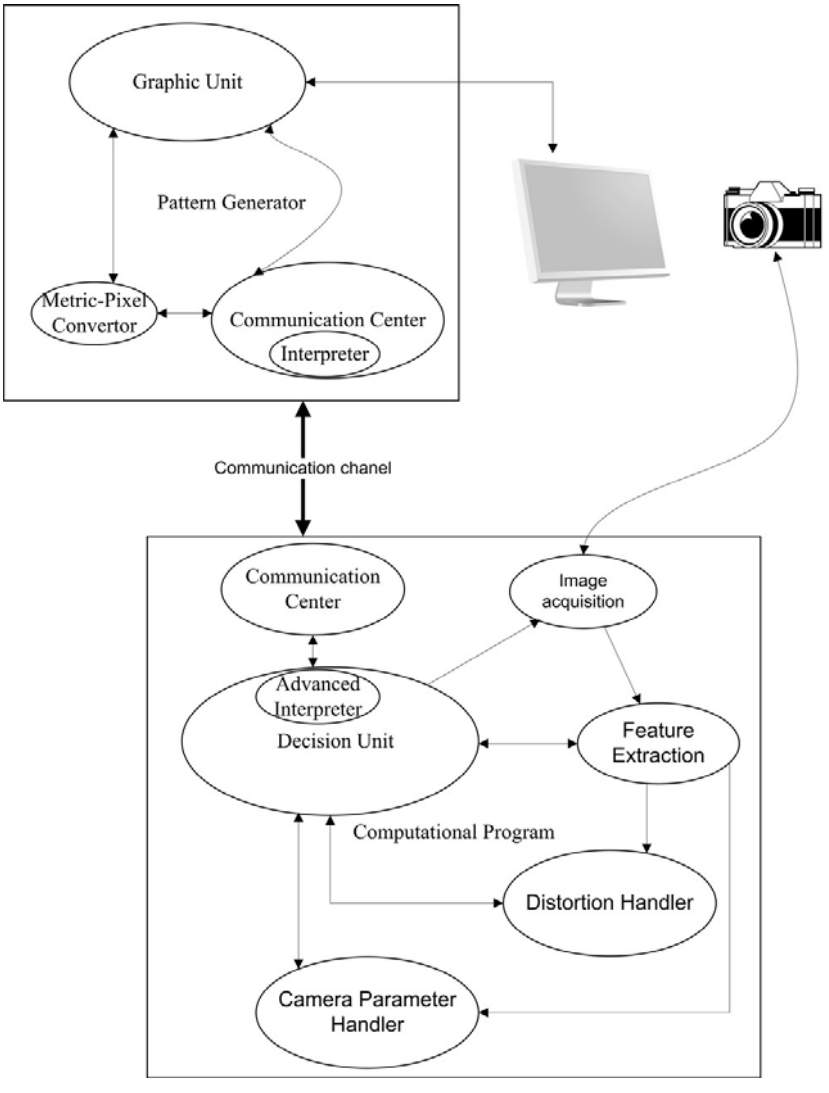


Fig. 2. Calibration framework using proposed active target implementation

be sent (e.g. it decides which component should receive the extracted features information).

4 Experiments

In this section the hardware setup and experiments performed is explained. Because of nature of active target approach the experiments are done using real data.

The video camera used in this experiment is a Sony camcorder (DCR-TRV460E) equipped with a CCD sensor, and a 2.5–50mm Sony lens. The lens focal length was kept to 2.5mm, which is the widest possible focal length in all the experiments. The camera is capable of USB streaming, so no digital to analog converter is needed. The frames are directly grabbed at the resolution of 640×480 in RGB color space and later converted to grayscale. A 15" TFT monitor with native resolution of 1024×768 (Sony SDM-HS53/H) was used to screen the patterns generated by pattern generator. A user defined color space with maximum backlight used meanwhile the experiment. All the frames where grabbed at daylight setting, where no other external light source is present. It was tried to have the camera optical axis orthogonal to image plane as shown in Fig. 3.

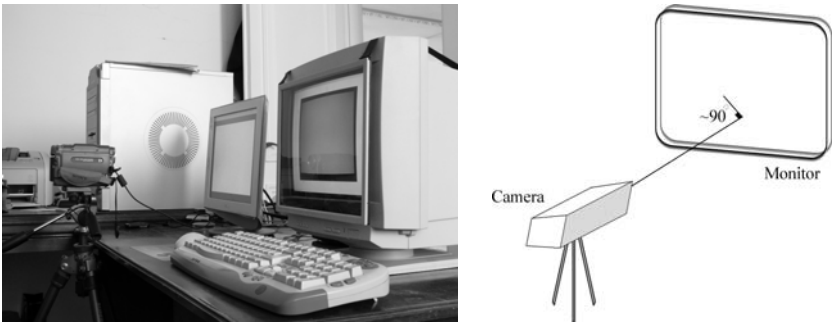


Fig. 3. Hardware setup used during the calibration process

Two crossing line segments are used as the calibration target. Their length grow to the size of image plane. The lines are generated and screened on the monitor. The hardware setup makes movement of lines by 0.3mm precision possible. However, as only the start, end and crossing point of these line segments are needed [20], the pattern generator only screens these interest points. Each interest point is screened and imaged independently.

The proposed active algorithm and the method of Harley and Kang [13] were compared. The video camera used has a fairly small distortion. In consequence the Harley's method fails to approximate the center and converges to the center of image as the distortion center. However, the proposed method approximates the distortion center accurately. The result is summarized in Table 1.

In order to test the accuracy of proposed method. The camera was fully calibrated under two different assumptions. First using the estimated center of distortion and again under the assumption of center of image. The result is summarized in Table 2. The calibration result was used to approximated the angle between two plane of a holder. Fig. 4 shows the holder. The ground truth is $90\,^{\circ}$. As shown in Table 2 the accuracy of the calibration using estimated center of distortion outperforms the center of image assumption.

Table 1. Center of distortion estimation

Algorithm	**Distortion Center**
Proposed Method	(321.6408, 247.4743)
Harley and Kang [13]	failed→(320, 240)

Table 2. Camera calibration and angle estimation results, f_i is the focal length in i_{th} direction, s is skew, u_0 and v_0 are principal point's coordinates, c is the center of radial distortion, k_1 and k_2 are the first two coefficients of radial distortion

	Calibration result	
	with known center of distortion	*without known center of distortion*
f_x	713.4747	719.5320
f_y	732.942	741.2015
s	−0.2157	−0.1802
u_0	242.4362	241.3101
v_0	322.1570	323.4587
$c(c_x, c_y)$	(321.6408, 247.4743)	(320, 240)
k_1	−0.01450	−0.01550
k_2	−0.00126	−0.00086
Estimated Angle	91.5585 °	93.8413 °

Fig. 4. A holder with ground truth 90 °, used for angle estimation

5 Conclusion

In this article a new approach to center of radial distortion estimation was introduced. The center of distortion estimation method originates in active calibration idea. However, the approach used has been based on active targets, which gives a new synthesis to active calibration.

The center of radial distortion can help to the increase of calibration precision. The center of radial distortion could be approximated using parametrization of distortion center and other camera parameters. Afterwards applying an iterative optimization technique. However, this require full camera calibration and is also vulnerable to trivial solutions.

There are also other techniques which require no iterative scheme. These methods focus on some especial properties of vision systems such as fundamental matrix. However, they sometimes suffer from some limitations such as amount of distortion.

The proposed algorithm is capable of approximating center of radial distortion without any prior information. It decouples the distortion parameter. This increases distortion coefficient approximation precision and calibration accuracy. As it was shown the proposed method approximates radial distortion center even in presence of small distortion value.

Camera calibration using the estimated center of radial distortion results in a more precise angle estimation. The meticulous result of angle estimation is an exemplar of center of radial distortion importance and accuracy of proposed algorithm.

References

1. Wang, J., Shi, F., Zhang, J., Liu, Y.: A new calibration model of camera lens distortion. Pattern Recognition 41(2), 607–615 (2008)
2. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation RA-3(4), 323–344 (1987)
3. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(10), 965–980 (1992)
4. Heikkila, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: Conference on Computer Vision and Pattern Recognition, CVPR 1997 (1997)
5. Claus, D., Fitzgibbon, A.: A rational function lens distortion model for general cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 213–219 (2005)
6. Devernay, F., Faugeras, O.: Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured environments. Machine Vision and Applications 13(1), 14–24 (2001)
7. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. I-25–I-32 (2001)
8. Hartley, R., Saxena, T.: The cubic rational polynomial camera model. In: ARPA Image Understanding Workshop, pp. 649–653 (1997)
9. Li, H.: Hartley, R.: A non-iterative method for lens distortion correction from point matches. In: OmniVis 2005 (workshop in conjunction with ICCV 2005), Beijing (2005)
10. Stein, G.P.: Internal camera calibration using rotation and geometric shapes. Technical report (1993)

11. Willson, R.C., Shafer, S.A.: What is the center of the image? In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1993), pp. 670–671 (1993)
12. Tardif, J.P., Sturm, P., Trudeau, M., Roy, S.: Calibration of cameras with radially symmetric distortion. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(9), 1552–1566 (2009)
13. Hartley, R., Kang, S.: Parameter-free radial distortion correction with centre of distortion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(8), 1309–1321 (2007)
14. Basu, A., Ravi, K.: Active camera calibration using pan, tilt and roll. IEEE Transactions on Systems, Man and Cybernetics, Part B 27(3), 559–566 (1997)
15. Konstantinos, D., Jorg, E.: Active intrinsic calibration using vanishing points. Pattern Recognition Letters 17(11), 1179–1189 (1996)
16. McLauchlan, P.F., Murray, D.W.: Active camera calibration for a head-eye platformusing the variable state-dimension filter. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(1), 15–22 (1996)
17. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1130–1134 (2000)
18. Heikkila, J.: Geometric camera calibration using circular control points. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1066–1077 (2000)
19. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, vol. 2. Cambridge University Press, Cambridge (2003)
20. Rezazadegan Tavakoli, H.: Automatic camera calibration mechanism. Master's thesis, Islamic Azad University of Mashhad, Iran (September 2008)

Rotation Averaging with Application to Camera-Rig Calibration

Yuchao Dai[1,2], Jochen Trumpf[2], Hongdong Li[3,2], Nick Barnes[3,2], and Richard Hartley[2,3]

[1] School of Electronics and Information,Northwestern Polytechnical University
Shaanxi Key Laboratory of Information Acquisition and Processing, China
[2] Research School of Information Sciences and Engineering
The Australian National University
[3] Canberra Research Lab, NICTA*

Abstract. We present a method for calibrating the rotation between two cameras in a camera rig in the case of non-overlapping fields of view and in a globally consistent manner. First, rotation averaging strategies are discussed and an L_1-optimal rotation averaging algorithm is presented which is more robust than the L_2-optimal mean and the direct least squares mean. Second, we alternate between rotation averaging across several views and conjugate rotation averaging to achieve a global solution. Various experiments both on synthetic data and a real camera rig are conducted to evaluate the performance of the proposed algorithm. Experimental results suggest that the proposed algorithm realizes global consistency and a high precision estimate.

1 Introduction

Multiple-camera systems have recently received much attention from the computer vision community. Two typical scenarios of applying multi-camera systems are (1) multi-camera networks for surveillance and (2) multi-camera rigs for motion recovery and geometry reconstruction. This paper is exclusively concerned with the latter case of multiple individual cameras rigidly mounted on a rig. Example applications of multi-camera rigs include camera tracking, 3D city modeling or creation of image panoramas and structure from motion [1,2,3].

Multi-camera systems use a set of cameras which are placed rigidly on a moving object like a vehicle with possibly non-overlapping or only slightly overlapping fields of view. In this case, images captured by different cameras do not share any or only a few common points. The system moves rigidly and correspondences between subsequent frames taken by the individual cameras are captured before and after the motion.

* 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. The first author would like to thank the Chinese Scholarship Council and Prof. Mingyi He for his immeasurable support and encouragement.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 335–346, 2010.

This non-overlapping arrangement poses difficulties in calibrating the multi-camera rig. Recent work done by Pollefeys et al. suggests a simple approach using a flat planar mirror [4]. Since it requires the use of a mirror, it is less convenient to use.

Esquivel et al. [5] proposed an approach for rig parameter estimation from non-overlapping views using sequences of time-synchronous poses of each camera. The presented approach works in three stages: internal camera calibration, pose estimation and rig calibration. They solve the problem using the relative motion measurements directly. However, according to our analysis, multiple relative motions are not consistent in general; using an averaging of motion strategy we can estimate the relative motion with high precision and in a globally consistent manner.

Our main contributions are: L_1-optimal rotation averaging strategy; L_2-optimal quaternion mean; global minimum with respect to the quaternion distance metric for the conjugate rotation problem and iterative rotation averaging for rotation calibration of multi-camera rig with non-overlapping views.

2 Existing Works on Rotation Averaging

Given several estimates of relative orientation of coordinate frames, a posteriori enforcement of global consistency has been shown to be an effective method of achieving improved rotation estimates. Govindu seems to be the first who introduced the idea of motion averaging for structure-from-motion computation in computer vision. He published a series of papers addressing this problem [6,7,8]. In [6] a simple linear least squares method is proposed where rotations in SO(3) are parameterized by quaternions and a closed-form linear least squares solution is derived. Although Govindu made a claim of optimality, the linear solution is not in fact optimal because the linear solution can not require each quaternion in the solution to have unit norm. It also ignores the difficulty that both a quaternion and its negative represent the same rotation, which can sometimes cause the method to fail.

The paper [7] further developed the above linear method by following a non-linear optimization on manifold approach. Because the set of all rotations carries the structure of a Lie group, it makes more sense to define the distance between two rotations as the geodesic distance on that Lie group. Based on this, the averaged "mean rotation" should be defined with respect to the geodesic distance. It will be made clear later that, while our new methods to be presented share the same spirit in this regard, Govindu's Lie-averaging algorithm uses a first order approximation only, whereas our approach makes no such approximation. Similar Lie-averaging techniques have been applied to the distributed calibration of a camera network [9], and to generalized mean-shifts on Lie groups [10]. A generic mathematical exposition of this topic can be found in [11].

Another paper by Govindu [8] basically tackles robustness problems where a RANSAC-type approach is adopted. In the present paper we demonstrate that the L_1-distance can be directly used for this purpose as the L_1-distance is well

known to be robust. What we really have achieved here is that we give an L_1-based averaging algorithm and prove its global convergence.

Martinec and Pajdla [12] discussed rotation averaging using the "chordal" metric, defined by $d_{\text{chord}}(\mathtt{R}_1, \mathtt{R}_2) = \|\mathtt{R}_1 - \mathtt{R}_2\|_{\text{Fro}}$. Averaging using the chordal metric suffers from similar problems to quaternion averaging. An analysis of averaging on SO(3) under the chordal metric has recently appeared [13].

When covariance uncertainty information is available for each local measurement, Agrawal shows how to incorporate such information in the Lie-group averaging computation [14]. Alternatively, one could apply the belief propagation framework to take the covariance information into account [15].

In the above discussions, the problem in question is to find the averaged rotation $\bar{\mathtt{R}}$ from a set of rotations $\{\mathtt{R}_1, \ldots, \mathtt{R}_n\}$ measured in the same coordinate frame. In this paper, we consider two more challenging rotation averaging problems: rotation averaging over several views and conjugate rotation averaging. In the case of *conjugate rotations*, the distance is defined as $d(\mathtt{R}_i\mathtt{S}, \mathtt{S}\mathtt{L}_i)$ where rotation pairs $\mathtt{R}_i, \mathtt{L}_i$ are given, and the rotation $\mathtt{S}$ is to be found. One traditional way to solve the conjugate-rotation problem is by solving a Sylvester equation treating each of the rotations as a generic 3×3 matrix (e.g. used in robot hand-eye calibration) [16].

Most of the papers on rotation averaging in the vision literature have omitted any discussion of optimality or global convergence. In addition, it seems they all overlooked the ambiguity of the sign problem associated with the quaternion representation, which invalidates previously known algorithms in some configurations. We have obtained rigorous conditions for convergence for most of our algorithms, though space does not allow us to include all proofs here.

3 Problem Formulation

We consider a camera rig consisting of two cameras, denoted left and right, fixed rigidly with respect to each other and individually calibrated. The camera rig undergoes rigid motion and captures several image pairs. We denote the coordinate frames of the cameras at time i by $\mathtt{M}_i^L$ and $\mathtt{M}_i^R$, respectively.

$$\mathtt{M}_i^L = \begin{bmatrix} \mathtt{L}_i & \mathbf{t}_i^L \\ \mathbf{0}^\top & 1 \end{bmatrix} \text{ and } \mathtt{M}_i^R = \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i^R \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

The first three rows of these matrices represent the projection matrices of the corresponding cameras, where image points are represented in coordinates normalized by the calibration matrix.

We denote the relative motion of $\mathtt{M}_0^R$ with respect to $\mathtt{M}_0^L$ by a transformation $\mathtt{M}^{LR}$, such that $\mathtt{M}^{LR} = \mathtt{M}_0^R\,(\mathtt{M}_0^L)^{-1}$. Since this relative motion remains fixed throughout the motion, we observe that $\mathtt{M}^{LR} = \mathtt{M}_i^R\,(\mathtt{M}_i^L)^{-1}$ for all i.

Next, the relative motion of $\mathtt{M}_j^L$ with respect to $\mathtt{M}_i^L$ is denoted by $\mathtt{M}_{ij}^L = \mathtt{M}_j^L\,(\mathtt{M}_i^L)^{-1}$. Similarly, $\mathtt{M}_{ij}^R = \mathtt{M}_j^R\,(\mathtt{M}_i^R)^{-1}$. Using the relation $\mathtt{M}_i^R = \mathtt{M}^{LR}\,\mathtt{M}_i^L$, we find

$$\mathtt{M}_{ij}^R = \mathtt{M}^{LR}\,\mathtt{M}_{ij}^L\,(\mathtt{M}^{LR})^{-1} \tag{1}$$

for all i, j. Now, we denote

$$\mathtt{M}_{ij}^{L} = \begin{bmatrix} \mathtt{L}_{ij} & \mathbf{t}_{ij}^{L} \\ \mathbf{0}^{\top} & 1 \end{bmatrix} \text{ and } \mathtt{M}_{ij}^{R} = \begin{bmatrix} \mathtt{R}_{ij} & \mathbf{t}_{ij}^{R} \\ \mathbf{0}^{\top} & 1 \end{bmatrix}.$$

Observe that the relative rotations $\mathtt{R}_{ij}, \mathtt{L}_{ij}$ and relative translations $\mathbf{t}_{ij}^{R}, \mathbf{t}_{ij}^{L}$ may be computed via the essential matrix for the (i, j) image pairs.

Writing the transformation $\mathtt{M}^{LR}$ as $\begin{bmatrix} \mathtt{S} & \mathbf{s} \\ \mathbf{0}^{\top} & 1 \end{bmatrix}$, we deduce from (1) the equations

$$\mathtt{R}_{ij} = \mathtt{S}\mathtt{L}_{ij}\mathtt{S}^{-1} \tag{2}$$

$$\mathbf{t}_{ij}^{R} = \mathtt{S}\mathbf{t}_{ij}^{L} + (\mathtt{I} - \mathtt{L}_{ij})\mathbf{s} \tag{3}$$

Calibration strategy. Our prescribed task is to find the relative motion between the right and left cameras, namely the transformation $\mathtt{M}^{LR}$. Our method uses the following general framework.

1. Compute the relative rotations and translations $(\mathtt{R}_{ij}, \mathbf{t}_{ij}^{R})$ $(\mathtt{L}_{ij}, \mathbf{t}_{ij}^{L})$ for many pairs (i, j) using the essential matrix.
2. Compute the relative rotation $\mathtt{S}$ from (2).
3. Solve linearly for $\mathbf{s}$ using (3).

Both these equations may be solved linearly. The rotation equation may be written as $\mathtt{S}\mathtt{L}_{ij} = \mathtt{R}_{ij}\mathtt{S}$, which is linear in the entries of $\mathtt{S}$. In solving for the translation $\mathbf{s}$, we note that the relative translations $\mathbf{t}_{ij}^{L}$ and $\mathbf{t}_{ij}^{R}$ are known only up to scale factors λ_{ij} and μ_{ij}. Then (3) may be written more exactly as $\lambda_{ij}\mathbf{t}_{ij}^{R} = \mu_{ij}\mathtt{S}\mathbf{t}_{ij}^{L} + (\mathtt{I} - \mathtt{L}_{ij})\mathbf{s}$, where everything is known except for $\mathbf{s}$ and the scales λ_{ij} and μ_{ij}. Three image pairs are required to solve these equations and find $\mathbf{s}$.

The strategy outlined here is workable, but relies on accurate measurements of the rotations $\mathtt{L}_{ij}$ and $\mathtt{R}_{ij}$. In the following sections of this paper, we will explain our strategies for rotation averaging that will lead to significantly improved results in practice. Although we have implemented the complete calibration algorithm, including estimation of the translation $\mathbf{s}$, for the rest of this paper, we will consider only rotation estimation.

4 Averaging Rotations

The relative rotation estimates $\mathtt{R}_{ij}$ and $\mathtt{L}_{ij}$ obtained from individual estimates using the essential matrix will not be consistent. In particular, ideally, there should exist rotations $\mathtt{L}_i$, $\mathtt{R}_i$ and $\mathtt{S}$ such that $\mathtt{L}_{ij} = \mathtt{L}_j\mathtt{L}_i^{-1}$ and $\mathtt{R}_{ij} = \mathtt{R}_j\mathtt{R}_i^{-1} = \mathtt{S}\mathtt{L}_{ij}\mathtt{S}^{-1}$. If these two conditions are satisfied, then the relative rotation estimates $\mathtt{R}_{ij}$ and $\mathtt{L}_{ij}$ are consistent. In general they will not be, so we need to adjust them by a process of rotation averaging.

A distance measure $d : \mathrm{SO}(3) \times \mathrm{SO}(3) \to \mathbb{R}$ is called *bi-invariant* if $d(\mathtt{S}\mathtt{R}_1, \mathtt{S}\mathtt{R}_2) = d(\mathtt{R}_1, \mathtt{R}_2) = d(\mathtt{R}_1\mathtt{S}, \mathtt{R}_2\mathtt{S})$ for all $\mathtt{S}$ and $\mathtt{R}_i$. Given an exponent $p \geq 1$ and a set

of $n \geq 1$ rotations $\{\mathtt{R}_1, \ldots, \mathtt{R}_n\} \subset \mathrm{SO}(3)$ we define the L^p-mean rotation with respect to d as

$$d^{\mathrm{p}}\text{-mean}(\{\mathtt{R}_1, \ldots, \mathtt{R}_n\}) = \underset{\mathtt{R} \in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d^p(\mathtt{R}_i, \mathtt{R}). \tag{4}$$

4.1 The Geodesic L_2-Mean

The *geodesic* distance function $d_{\mathrm{geod}}(\mathtt{R}, \mathtt{S})$ is defined as the rotation angle $\angle(\mathtt{R}\mathtt{S}^\top)$. It is related to the angle-axis representation of a rotation in which a rotation is represented by the vector $\theta\mathbf{v}$, where $\mathbf{v}$ is a unit 3-vector representing the axis, and θ is the angle of rotation about that axis. We denote by $\log(\mathtt{R})$ the angle-axis representation of $\mathtt{R}$. Then $d(\mathtt{R}, \mathtt{S}) = \|\log(\mathtt{R}\mathtt{S}^\top)\|$. The inverse of this mapping is the exponential $\mathtt{R} = \exp(\theta\mathbf{v})$.

The associated L_2-mean is usually called the Karcher mean [17] or the geometric mean [11]. A necessary condition [11, (3.12)] for $\mathtt{R}$ to be a d^2_{geod}-mean of $\{\mathtt{R}_1, \ldots, \mathtt{R}_n\}$ is given by $\sum_{i=1}^{n} \log(\mathtt{R}^\top \mathtt{R}_i) = 0$.

The mean is unique provided the given rotations $\mathtt{R}_1, \ldots, \mathtt{R}_n$ do not lie too far apart [17, Theorem 3.7], more precisely if $\{\mathtt{R}_1, \ldots, \mathtt{R}_n\}$ lie in an open ball $B(\mathtt{R}, \pi/2)$ of geodesic radius $\pi/2$ about some rotation $\mathtt{R}$. For this case Manton [18] has provided the following convergent algorithm where the inner loop of the algorithm is computing the average in the tangent space and then projecting back.

1: Set $\mathtt{R} := \mathtt{R}_1$. Choose a tolerance $\varepsilon > 0$.
2: **loop**
3: Compute $\mathbf{r} := \frac{1}{n} \sum_{i=1}^{n} \log\left(\mathtt{R}^\top \mathtt{R}_i\right)$.
4: **if** $\|\mathbf{r}\| < \varepsilon$ **then**
5: **return** $\mathtt{R}$
6: **end if**
7: Update $\mathtt{R} := \mathtt{R}\exp(\mathbf{r})$.
8: **end loop**

Algorithm 1. computing the Karcher mean on SO(3)

4.2 The Geodesic L_1-Mean

Another interesting mean with respect to the geodesic distance d_{geod} is the associated L_1-mean

$$d_{\mathrm{geod}}\text{-mean}(\{\mathtt{R}_1, \ldots, \mathtt{R}_n\}) = \underset{\mathtt{R} \in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_{\mathrm{geod}}(\mathtt{R}_i, \mathtt{R}) \ , \tag{5}$$

which we might assume to be more robust to errors.

We propose a Riemannian gradient descent algorithm with geodesic line search to compute the L_1-mean. As long as Algorithm 2 avoids arbitrarily small but fixed δ-neighborhoods of the $\mathtt{R}_i$s, convergence to the set of critical points of f

1: Set $\mathtt{R} := d^2_{\mathrm{geod}}$-mean$(\{\mathtt{R}_1, \ldots, \mathtt{R}_n\})$. Choose a tolerance $\varepsilon > 0$.
2: **loop**
3: Compute $\mathbf{r} := \sum_{i=1}^{n} \log(\mathtt{R}^\top \mathtt{R}_i)/\| \log(\mathtt{R}^\top \mathtt{R}_i)\|$.
4: Compute $s^* := \mathrm{argmin}_{s \geq 0}\ f(\mathtt{R} \exp(s\mathbf{r}))$.
5: **if** $\|s\mathbf{r}\| < \varepsilon$ **then**
6: **return** $\mathtt{R}$
7: **end if**
8: Update $\mathtt{R} := \mathtt{R} \exp(s\mathbf{r})$.
9: **end loop**

Algorithm 2. computing the geodesic L_1-mean on SO(3)

follows from [19, Corollary 4.3.2] applied to a modification of f obtained by smoothing f within those δ-neighborhoods [20].

Note that possibly the easiest way to implement the line search in Step 4 is a Fibonacci search on a large enough interval. We have suggested to initialize the algorithm with the Karcher mean, but other initializations would of course be possible.

4.3 Quaternion Averaging

A rotation $\mathtt{R}$ may be represented by a quaternion $\mathbf{r}$, which is a unit 4-vector, defined as follows. If $\mathbf{v}$ is the axis of the rotation and θ is the angle of the rotation about that axis, then $\mathbf{r}$ is defined as $\mathbf{r} = (\cos(\theta/2), \mathbf{v}\sin(\theta/2))$. We may think to define a distance $d_{\mathrm{quat}}(\mathtt{S}, \mathtt{R})$ between two rotations to be $d_{\mathrm{quat}}(\mathtt{R}, \mathtt{S}) = \|\mathbf{r} - \mathbf{s}\|$. Unfortunately, this simple equation will not do, since both $\mathbf{r}$ and $-\mathbf{r}$ represent the same rotation, and it is not clear which one to choose. However, this is resolved by defining

$$d_{\mathrm{quat}}(\mathtt{R}, \mathtt{S}) = \min(\|\mathbf{r} - \mathbf{s}\|, \|\mathbf{r} + \mathbf{s}\|)\ .$$

Since quaternions satisfy the condition $\|\mathbf{r} \cdot \mathbf{t}\| = \|\mathbf{r}\|\,\|\mathbf{t}\|$, where $\mathbf{r} \cdot \mathbf{t}$ represents the quaternion product, it is easily verified that the quaternion distance is bi-invariant.

The relationship of this to the geodesic distance is as follows. Let $d_{\mathrm{geod}}(\mathtt{R}, \mathtt{S}) = d_{\mathrm{geod}}(\mathtt{I}, \mathtt{R}^\top \mathtt{S}) = \theta$, which is equal to the angle of the rotation $\mathtt{R}\mathtt{S}^\top$. Then simple trigonometry provides the relationship $d_{\mathrm{quat}}(\mathtt{I}, \mathtt{R}^\top \mathtt{S}) = 2\sin(\theta/4)$. For small rotations, we see that $d_{\mathrm{quat}}(\mathtt{R}, \mathtt{S}) \approx d_{\mathrm{geod}}(\mathtt{R}, \mathtt{S})/2$.

The following theorem shows how the L_2 quaternion mean of a set of rotations $\mathtt{R}_i$ may be computed, it is defined as $\mathrm{argmin}_{\mathtt{R}} \sum_{i=1}^{n} d^2_{\mathrm{quat}}(\mathtt{R}, \mathtt{R}_i)$ [20].

Theorem 1. *Let $\mathtt{R}_i; i = 1, \ldots, n$ be rotations, and suppose that there exists a rotation $\mathtt{S}$ such that $d_{\mathrm{geod}}(\mathtt{R}_i, \mathtt{S})$ is less than $\pi/2$. Let $\mathbf{r}_i$ be the quaternion representation of $\mathtt{R}_i$ chosen with sign such that $\|\mathbf{r}_i - \mathbf{s}\|$ is the smaller of the two choices. Then the L_2 quaternion mean of the rotations $\mathtt{R}_i$ is represented by the quaternion $\bar{\mathbf{r}}/\|\bar{\mathbf{r}}\|$, where $\bar{\mathbf{r}} = \sum_{i=1}^{n} \mathbf{r}_i$.*

4.4 The Conjugate Averaging Problem

We now consider the problem of conjugate averaging. This problem is motivated by the second step of the calibration algorithm outlined in Section 3. The general form of this problem is as follows. Let $(\mathtt{R}_i, \mathtt{L}_i); i = 1, \ldots, n$ be pairs of rotations. (In Section 3 these rotations have two subscripts, namely $\mathtt{R}_{ij}, \mathtt{L}_{ij}$). The conjugate averaging problem is to find the rotation $\mathtt{S}$ that minimizes

$$\sum_{i=1}^{n} d^p(\mathtt{R}_i\mathtt{S}, \mathtt{S}\mathtt{L}_i) \ . \tag{6}$$

This problem has not been explicitly addressed in the context of multi-camera rigs, as far as we know, though it has been studied as the "hand-eye coordination problem" in robotics [16]. We give here an optimal solution for the L_2 quaternion distance metric under certain conditions.

We make the observation that if $\mathtt{R}_i$ and $\mathtt{L}_i$ are exactly conjugate, then they have the same rotation angle. In general, we assume that they do not differ by too much. One condition we need to give a closed form solution to this problem is that the rotations $\mathtt{R}_i$ and $\mathtt{L}_i$ should not be too large. In fact, we assume that the angle θ associated with $\mathtt{R}_i$ or $\mathtt{L}_i$ is less than some angle $\theta_{\max} < \pi$. For the application we are interested in, where $\mathtt{R}_i$ and $\mathtt{L}_i$ are relative rotations between two positions of a camera, the rotation angle of $\mathtt{R}_i$ can not be very large. If for instance the rotation $\mathtt{R}$ between two positions of a camera approaches π, then at least for normal cameras, there will be no points visible in both images, and hence no way to estimate the rotation $\mathtt{R}$. Normally, the rotation $\mathtt{R}_{ij}$ between two positions of the camera will not exceed the field of view of the camera, otherwise there will not be any matched points for the two cameras (except possibly for points lying between the two camera positions).

We now state the conditions under which we can guarantee an optimal solution to the conjugate averaging problem.

1. The rotations $\mathtt{L}_i$ and $\mathtt{R}_i$ satisfy the conditions $\angle(\mathtt{L}_i) < \theta_{\max}$ and $\angle(\mathtt{R}_i) < \theta_{\max}$.
2. In the optimal solution to problem (6), the errors $d_{\text{geod}}(\mathtt{R}_i\mathtt{S}, \mathtt{S}\mathtt{L}_i) < \alpha_{\max}$.
3. $\theta_{\max} + \alpha_{\max}/2 < \pi$.

Thus, we are assuming that the errors plus angles are not too large. In particular, since $\alpha_{\max} \leq \pi$, we see that the last two conditions always hold if $\theta_{\max} < \pi/2$.

Linear solution. We now outline a linear algorithm for estimating the matrix $\mathtt{S}$, under the L_2 quaternion distance. Let $\mathbf{r}_i$ and $\mathbf{l}_i$ be quaternion representatives of the rotations $\mathtt{R}_i$ and $\mathtt{L}_i$, chosen such that $\mathbf{r}_i = (\cos(\theta_i/2), \sin(\theta_i/2)\mathbf{v})$ with $\theta_i < \pi$. This means that the first component $\cos(\theta_i/2)$ of the quaternion is positive. This fixes the choice between $\mathbf{r}_i$ and $-\mathbf{r}_i$. We define $\mathbf{l}_i$ similarly.

Now, consider the equation $\mathtt{R}_i\mathtt{S} = \mathtt{S}\mathtt{L}_i$, and write it in terms of quaternions as $\mathbf{r}_i \cdot \mathbf{s} - \mathbf{s} \cdot \mathbf{l}_i = \mathbf{0}$. As before, $\cdot$ represents quaternion multiplication. Since quaternion multiplication is bilinear in terms of the entries of the two quaternions involved,

this gives a homogeneous linear equation in terms of the entries of $\mathbf{s}$. Stacking all these equations into one and finding the solution such that $\|\mathbf{s}\| = 1$, we may solve for $\mathbf{s}$. This gives a simple linear way to solve this problem. Under the conditions stated above, we can prove that this algorithm finds the global minimum with respect to the quaternion distance metric [20].

4.5 Iterative Rotation Averaging for Camera Rig Calibration

The cost function that we minimize is the residual error in the rotation measurements $\mathtt{R}_{ij}$ and $\mathtt{L}_{ij}$, defined by

$$\min_{\mathtt{S},\mathtt{L}_i} \sum_{(i,j)\in\mathcal{N}} d^p(\mathtt{L}_{ij}, \mathtt{L}_j\mathtt{L}_i^{-1}) + d^p(\mathtt{R}_{ij}, \mathtt{S}\mathtt{L}_j\mathtt{L}_i^{-1}\mathtt{S}^{-1}) \tag{7}$$

There seems to be no direct method of minimizing this cost function under any of the metrics we consider. Therefore, our strategy is to minimize the cost function by using rotation averaging to update each $\mathtt{L}_i$ in turn, then conjugate rotation averaging to find $\mathtt{S}$. At each step of this algorithm, the total cost decreases, and hence converges to a limit. We do not at present claim a rigorous proof that the algorithm converges to even a local minimum, though that seems likely under most reasonable conditions. In particular, the sequence of estimates must contain a convergent subsequence, and the limit of this subsequence must be at least a local minimum with respect to each $\mathtt{L}_i$ and $\mathtt{S}$ individually.

Initial values for each $\mathtt{L}_i$ are easily found by propagating from a given rotation $\mathtt{L}_0$ assumed to be the identity, and then obtaining the initial $\mathtt{S}$ through conjugate averaging.

The complete rotation estimation procedure follows.

1: Choose a tolerance $\epsilon > 0$.
2: Estimate initial values of $\mathtt{L}_i$ through rotation propagation.
3: Estimate $\mathtt{S}$ from $\mathtt{R}_{ij}\mathtt{S} = \mathtt{S}\mathtt{L}_{ij}$ solving the quaternion least squares problem.
4: **loop**
5: Update each $\mathtt{L}_j$ in turn by averaging all the rotations $\mathtt{L}_{ij}\mathtt{L}_i$ and $\mathtt{S}^{-1}\mathtt{R}_{ij}\mathtt{S}\mathtt{L}_i$.
6: Recompute and update $\mathtt{S}$ from the equation $\mathtt{R}_{ij}\mathtt{S} = \mathtt{S}\mathtt{L}_j\mathtt{L}_i^{-1}\mathtt{S}$ using conjugate rotation averaging.
7: **if** the RMS error has decreased by less then ϵ since the last iteration, **then**
8: **return** S
9: **end if**
10: **end loop**

Algorithm 3. Iterative Rotation Averaging

5 Experiments

To evaluate the performance of the proposed algorithms, we conducted experiments on both synthetic data and real images. A comparison with other methods is presented to show the improved accuracy of the proposed method.

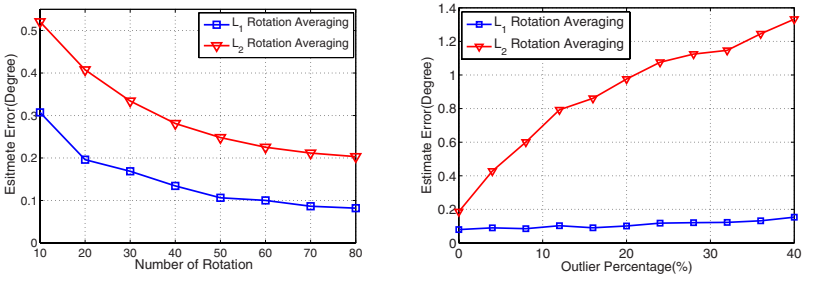
In the experiments reported below, we used L_1 and L_2 geodesic rotation averaging to compute each $\mathtt{L}_i$, but used quaternion averaging for the conjugate

rotation averaging to compute S. Theoretically this is not ideal, but we find it works well in practice and we will discuss all the possible combinations in future work.

5.1 Synthetic Rotation Averaging

In the first group of synthetic experiments, we evaluate the performance of L_1 rotation averaging and L_2 rotation averaging on a bunch of rotation measurements. First we generate a random rotation $\mathbf{r}$ and the corresponding rotation matrix R. A normally distributed angle θ with mean 0 and standard deviation σ is generated to simulate the effect of random rotation noise. The rotation axis is generated uniformly in the cube $[-1, 1]^3$ and then normalized to a unit vector $\mathbf{r}$. Then the rotation noise is expressed as $\theta\mathbf{r}$ and the corresponding rotation matrix is denoted $\mathtt{R}_{err}$. Finally the simulated rotation measurement is taken as $\mathtt{RR}_{err}$.

All the results are obtained as the mean of 200 trials. The evaluation metric is the angle between the ground truth rotations and the estimates.



(a) (b)

Fig. 1. Performance comparison of L_1-optimal rotation averaging and L_2-optimal rotation averaging. (a) Angle difference between the ground truth rotations and the averaging results for various numbers of rotations, where normally distributed rotation noise with standard deviation parameter $\sigma = 2$ degrees is added and no outliers are included. (b) Angle difference between the ground truth rotations and the averaging results on 100 rotations for various levels of outliers, where normally distributed rotation noise with standard deviation parameter $\sigma = 2$ degrees is added, and the outliers are simulated using normally distributed noise with standard deviation parameter $\sigma = 20$ degrees followed by selecting the samples with an angle error larger than 5 degrees.

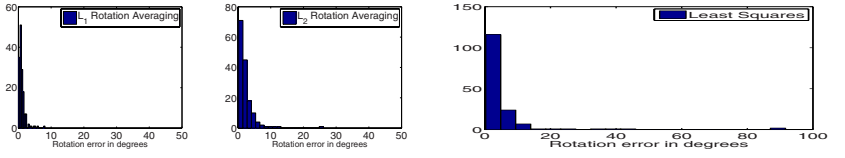
From both figures in Figure 1 we conclude that the L_1-mean is more robust than the L_2-mean, especially in the presence of outliers.

5.2 Synthetic Camera Rig

To simulate a camera rig system, a rig with two cameras is generated with various numbers of frames. First the relatives rotation S of the camera rig is randomly generated. Second, the orientation $\mathtt{L}_i$ of the left camera is generated and the corresponding orientation $\mathtt{R}_i$ of the right camera is obtained. Third, whether a

pair of frames has an epipolar geometry relationship is determined according to some probability distribution. If there exists epipolar geometry, the relative rotation measurement is obtained as $\mathtt{L}_{ij} = \mathtt{L}_j \mathtt{L}_i^{-1}$ and $\mathtt{R}_{ij} = \mathtt{R}_j \mathtt{R}_i^{-1}$. A random error rotation is applied to simulate noise in the rotation measurements.

To evaluate the performance of L_1-mean based rig rotation calibration, L_2-mean based rig rotation calibration and direct least squares rig rotation calibration, we conducted 200 separate experiments on synthetic camera rig data which contains 20 frames of motion. The possibility of existence of a relative measurement is 0.5 and 10% outliers are added where the rotation error is larger than 5 degrees. The histograms of the resulting errors are illustrated as Figure 2 and the histograms imply that our proposed L_1 rotation calibration estimates the rotation better than L_2 rotation calibration and direct least squares.



(a) (b) (c)

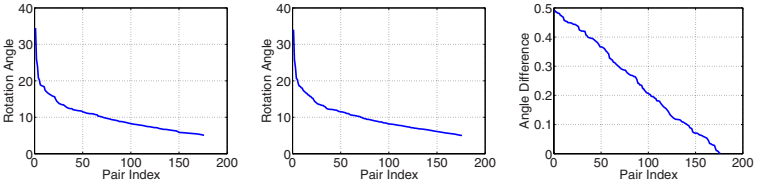
Fig. 2. (a) Histogram of rotation error using L_1 rotation averaging. It shows a mean of 1.12 degrees and standard deviation of 1.05 degrees. (b) Histogram of rotation error using L_2 rotation averaging. It shows a mean of 2.41 degrees and standard deviation of 2.75 degrees. (c) Histogram of rotation error using direct least squares. It shows a mean of 5.14 degrees and standard deviation of 11.65 degrees.

5.3 Experiments on Real Images

As a real example of a two-camera rig system, we have used a pair of wide-angle cameras to capture sequences of images. Images are captured at each camera illustrated in Figure 3. Feature points on the images are extracted using SIFT and tracked through image sequences. These tracked features are transformed to image vectors on the unit sphere given the individual intrinsic calibrations. Outliers in the tracked features are removed using RANSAC [21] to fit the essential matrix using the normalized 8 point algorithm. Pairwise relative pose is obtained through decomposition of the essential matrix, and two frames bundle adjustment is utilized to refine the estimate, thus obtaining the relative rotations $\mathtt{L}_{ij}, \mathtt{R}_{ij}$. Finally, L_1 and L_2 algorithms are applied to calibrate the camera rig, obtaining the relative rotation $\mathtt{S}$ and relative translation $\mathbf{s}$.

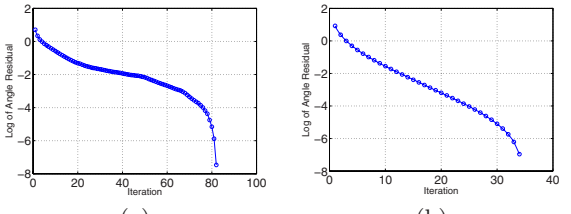
The image sequences captured by the left camera and the right camera contain 200 frames individually. As some pairs of image frames do not supply relative motion estimates, we ultimately obtained 11199 pairs of relative motion estimates. Since relative rotation estimates $\mathtt{L}_{ij}$ and $\mathtt{R}_{ij}$ should have equal angle rotations, we use this criterion along with a minimum rotation angle requirement to select the best image pairs for further processing. After rotation selection, we obtained 176 pairs of synchronized motions. The distributions of the rotation angles and angle differences for these pairs are shown in Figure 4.

Fig. 3. Images captured by camera rig with non-overlapping views



(a) (b) (c)

Fig. 4. (a) Angle distribution of the left camera. (b) Angle distribution of the right camera. (c) Distribution of the difference between the angle of the left camera and the angle of the right camera.



(a) (b)

Fig. 5. Convergence process on real camera rig image sequences. (a) Log of Angle Residual of L_1 rotation averaging. (b) Log of Angle Residual of L_2 rotation averaging.

The convergence process is shown in Figure 5 with the L_1 rotation averaging and quaternion conjugate result corresponding to an angle of 143.1° and the L_2 rotation averaging result corresponding to an angle of 169.4°. Measured from the scene, the ground truth is about 140°.

6 Conclusion and Future Work

Rotation averaging is an important component of our method of camera rig calibration. Individual rotation estimation is sensitive to outliers and geometrically critical configurations. It was shown that our new L_1 rotation averaging method gives markedly superior results to L_2 methods. Global bundle adjustment is recommended for final polishing. Previous computer vision literature has largely ignored issues such as convergence and optimality of rotation averaging algorithms. We have addressed this issue. Our complete analysis will be made available in an extended version of this paper.

References

1. Kim, J.H., Li, H., Hartley, R.: Motion estimation for multi-camera systems using global optimization. In: Proc. Comput. Vis. Pattern Recognit., pp. 1–8 (2008)
2. Kim, J.H., Hartley, R., Frahm, J.M., Pollefeys, M.: Visual odometry for non-overlapping views using second-order cone programming. In: Proc. Asian Conf. on Computer Vision, pp. 353–362 (2007)
3. Li, H., Hartley, R., Kim, J.H.: A linear approach to motion estimation using generalized camera models. In: Proc. Comput. Vis. Pattern Recognit., pp. 1–8 (2008)
4. Kumar, R.K., Ilie, A., Frahm, J.-M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: Proc. Comput. Vis. Pattern Recognit., pp. 1–7 (2008)
5. Esquivel, S., Woelk, F., Koch, R.: Calibration of a multi-camera rig from non-overlapping views. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 82–91. Springer, Heidelberg (2007)
6. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proc. Comput. Vis. Pattern Recognit., pp. 218–225 (2001)
7. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proc. Comput. Vis. Pattern Recognit., pp. 684–691 (2004)
8. Govindu, V.M.: Robustness in motion averaging. In: Proc. Asian Conf. on Computer Vision, pp. 457–466 (2006)
9. Tron, R., Vidal, R., Terzis, A.: Distributed pose averaging in camera networks via consensus on SE(3). In: Second ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1–10 (2008)
10. Tuzel, O., Subbarao, R., Meer, P.: Simultaneous multiple 3d motion estimation via mode finding on Lie groups. In: Int. Conf. Computer Vision, pp. 18–25 (2005)
11. Moakher, M.: Means and averaging in the group of rotations. SIAM J. Matrix Anal. Appl. 24(1), 1–16 (2002)
12. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Proc. Comput. Vis. Pattern Recognit., pp. 1–8 (2007)
13. Sarlette, A., Sepulchre, R.: Consensus optimization on manifolds. SIAM J. Control Optim. 48(1), 56–76 (2009)
14. Agrawal, M.: A Lie algebraic approach for consistent pose registration for general euclidean motion. In: Int. Conf. Intelligent Robots and Systems, pp. 1891–1897 (2006)
15. Devarajan, D., Radke, R.J.: Calibrating distributed camera networks using belief propagation. EURASIP J. Appl. Signal Process. 2007(1) (2007)
16. Park, F., Martin, B.: Robot sensor calibration: solving AX=XB on the euclidean group. IEEE Transactions on Robotics and Automation 10(5), 717–721 (1994)
17. Grove, K., Karcher, H., Ruh, E.A.: Jacobi fields and Finsler metrics on compact Lie groups with an application to differentiable pinching problems. Math. Ann. 211, 7–21 (1974)
18. Manton, J.H.: A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups. In: Proceedings of the Eighth Int. Conf. on Control, Automation, Robotics and Vision, Kunming, China, pp. 2211–2216 (2004)
19. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press, Princeton (2008)
20. Dai, Y., Trumpf, J., Li, H., Barnes, N., Hartley, R.: On rotation averaging in multi-camera systems. Technical report, Northwestern Polytechnical University and Australian National University (2009) (to appear)
21. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun. Assoc. Comp. Mach. 24, 381–395 (1981)

Single-Camera Multi-baseline Stereo Using Fish-Eye Lens and Mirrors

Wei Jiang*, Masao Shimizu**, and Masatoshi Okutomi

Graduate School of Science and Engineering, Tokyo Institute of Technology, Japan

Abstract. This report proposes a monocular range measurement system with a fish-eye lens and mirrors placed around the lens. The fish-eye lens has a wide view-angle; the captured image includes a centered region of direct observation and surrounding regions of mirrored observations. These regions correspond to observations with multiple cameras at different positions and orientations. The captured image can be used for direct observation of a target with the centered region. Simultaneously, it can be used for multi-baseline stereo to reconstruct three-dimensional information. After calibration of the projection function of the fish-eye lens, the mirror positions and orientations are obtainable from the external parameters, which are used for the multi-baseline stereo measurement. Experimental results demonstrate the effectiveness of a real working system.

1 Introduction

Range estimation or three-dimensional (3D) shape measurement using a noncontact method is a fundamental technique used for fields of security, intelligent transport systems (ITS) and robotic navigation. This technique has been widely studied; many commercially available products have been developed through its adaptation. Nonetheless, the study field remains active. Triangulation, which includes stereovision, active-stereo and structured light projection, is a basic method of range estimation. Triangulation requires some observations from different camera positions; multiple synchronized cameras, or a structured light projection device such as a PC projector and a camera are necessary to realize the method.

On the other hand, strong demands exist for the use of a single camera and a single image. Image layering and divided field-of-view (FOV) methods are the two major methods used to satisfy this demand and capture images from different camera positions in a single image. Both methods bring the range information to a single image; they can be used for dynamic objects and real-time applications.

As the image layering method, single-lens aperture [1], coded aperture [8] and reflection stereo [16] have been proposed. Nevertheless, the use efficiency of the incident light is not good in the single-lens aperture and coded aperture

* His current affiliation is the State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou, China.

** His current affiliation is the College of Science and Technology, Nihon University, Japan.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 347–358, 2010.

methods. The baseline length for stereo measurement is very narrow (around 10 [mm]) in all methods; the results are limited to a low resolution of range.

For the divided FOV method, a mirror [6] or a prism [10] is used to divide the FOV. Exactly identical brightness, contrast, and color are obtainable from the mirrored cameras at different positions, enabling simple matching between camera observations. In the divided FOV method, a multi-baseline stereo system with a single camera [15] was proposed to detect near objects, by using multiple specular spheres. Another system using a curved mirror [9] was proposed to reconstruct the 3D information from a large number of view-points. The proposed system also belongs to this category.

This study proposes a monocular range measurement system with a fish-eye lens and mirrors that are placed around the lens, as shown in Fig. 1. The fish-eye lens has a wide view-angle; the captured image includes a centered region of direct observation and surrounding regions of mirrored observations, as presented in Fig. 2. The captured image can be used for the direct observation of a target with the centered region. Simultaneously, it can be used for the multi-baseline stereo to reconstruct 3D information. After calibration of the projection function of the fish-eye lens, the mirror positions and orientations are obtainable from external parameters, which are used for the multi-baseline stereo [13] measurement.

The wide view-angle imaging system cannot be modeled by the perspective projection. The *generic* camera calibration [7],[14] has been proposed for such imaging system to describe the relation between ray direction and its corresponding pixel location. Instead of the fully generic calibration, we employ a parametric expression [5],[11] for many types of wide view-angle imaging system. This method is realized as an improvement [12] from the well-known camera calibration toolbox [3].

The remainder of the paper is structured as follows. Section 2 presents the proposed system and its equivalent multi-camera system. Section 3 explains fish-eye lens calibration and system calibration. Then in section 4 we represent a

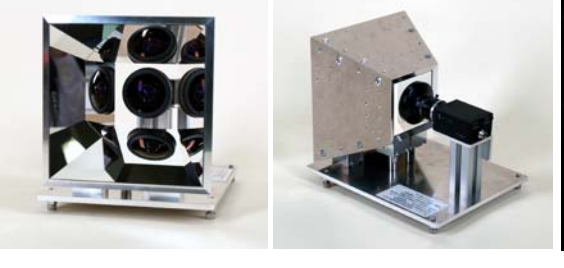


Fig. 1. Prototype of the proposed system

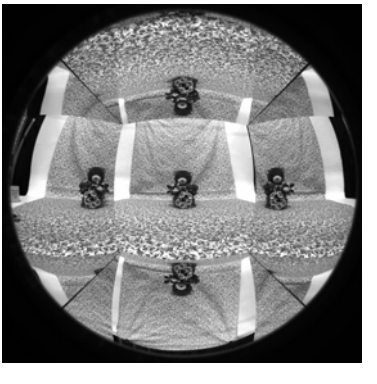


Fig. 2. Circular fish-eye image captured by the proposed system

two stereo algorithm for use in range estimation of the proposed system. Section 5 presents a description of the experimental results. We conclude this paper with some relevant remarks in section 6.

2 Proposed System

2.1 System Configuration

As shown in Fig. 1, four trapezoidal planar mirrors are placed around the fish-eye lens. The optical axes and FOV of the *mirrored cameras*, which capture the reflected light from the object at the mirrors, can be determined by the placement angle and size of the mirrors (will be described in the following subsections).

Figure 2 presents a circular fish-eye image, as captured by the single camera. The circular image is divided into five regions: the center and four surrounding regions. The center region can be considered as a regular (but with a heavy lens distortion) camera observation. The proposed system views the object directly. Moreover, the surrounding regions enable measurement of the 3D shape of the object by the multi-baseline stereo.

Four mirrors are used in the prototype system, but the number of mirrors (≥ 1) is arbitrary.

2.2 Equivalent Multi-Camera System

Figure 3 portrays a sectional side view of the proposed system. A camera coordinate is set with its origin at the lens optical center and the Z axis equating to the optical axis.

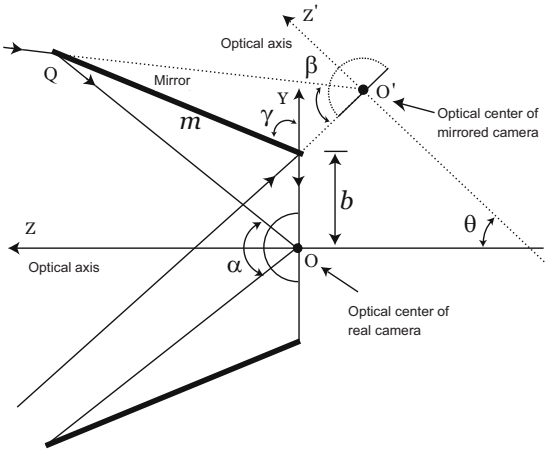


Fig. 3. Mirror placement and mirrored camera

Consider a mirrored camera O', with a reflection at the upper mirror. The vertical view-angle of the center region α, the vertical view-angle of the upper mirrored camera β, and the angle between the axes of the two cameras θ are functions of mirror position b, mirror size m, and mirror angle γ, as follows[1].

$$\alpha = \pi - 2\tan^{-1}\frac{\hat{m}\sin\gamma}{1+\hat{m}\cos\gamma}, \tag{1}$$

[1] The horizontal view-angle of the upper mirrored camera is almost the same as that of the center region in the prototype system.

$$\beta = \frac{1}{2}(\pi - \alpha), \tag{2}$$

$$\theta = \pi - 2\gamma, \tag{3}$$

$$O'(y, z) = (b(1 - \cos 2\gamma), -b \sin 2\gamma) \tag{4}$$

Therein, $\hat{m} = m/b$ denotes a normalized mirror size. Furthermore, $O'(y, z)$ denotes the position of the mirrored camera.

The other three mirrored cameras can be represented similarly.

2.3 Dimensional Design of Mirrors

As described above, the view angle α and β, and the position of the mirrored camera can be determined by the normalized mirror size $\hat{m}$ and mirror angle γ. This subsection represents a design guideline of the mirror by evaluating the respective common view angles of two cameras (center and upper) Ω_2 and of three cameras (center, upper, and lower) Ω_3. The common view angle Ω_2 indicates that at least two of the three cameras can detect a far distant object in this angle range, whereas the angle Ω_3 indicates that all three cameras can see the object.

As presented in Fig. 4, the common angle Ω_2 is obtainable from the following four angles, which are the functions of the mirror angle γ.

$$\begin{cases} \alpha_1 = \tan^{-1} \frac{\hat{m} \sin \gamma}{1 + \hat{m} \cos \gamma} \\ \alpha_2 = \pi - \alpha_1 \end{cases} \quad \begin{cases} \beta_1 = 2\gamma - \alpha_1 \\ \beta_2 = 2\gamma \end{cases} \tag{5}$$

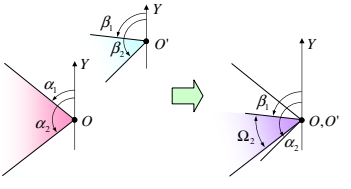


Fig. 4. Common angle for a far distant object

The common angle Ω_2 has a maximum at a specific mirror angle because the magnitude relation of α_1 and β_1, and that of α_2 and β_2 change with the mirror angle γ. The common angle Ω_3 is similarly obtainable.

Figures 5 and 6 respectively portray view angles and common angles. Larger view angles are preferred, but the view angle of the center region α has a priority over the view angle of the upper camera β. A larger common angle Ω_2 is preferred

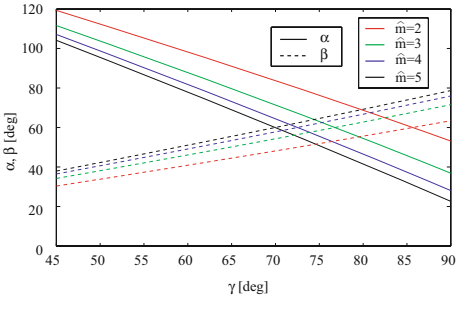


Fig. 5. View angle α and β

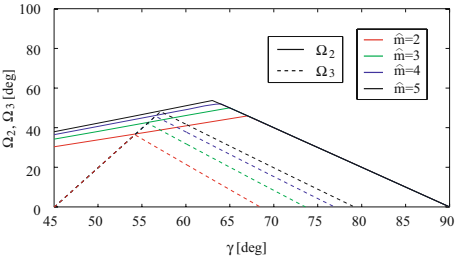


Fig. 6. Common angle Ω_2 and Ω_3

because the object in this angle range can be measured. Moreover, a smaller normalized mirror size $\hat{m}$ is preferred for a smaller size of the mirror system.

For the prototype system, we chose the mirror angle $\gamma = 65.0$ [deg] and the normalized mirror size $\hat{m} = 3.0$. In this case, the view angle of the center region is $\alpha \approx 80$ [deg], the view angle of the upper camera is $\beta \approx 50$ [deg], and the common angle of the two cameras is $\Omega_2 \approx 50$ [deg].[2]

The proposed system captures an image of five regions: the center region with a real camera and the surrounding four regions with mirrored cameras. The common angles described above are the angles in a horizontal or vertical direction. They tell that the available camera number for the multi-baseline stereo differs with respect to the image position in the center region. Figure 7 portrays the number of cameras available for stereo measurement in the center region.

3	4	3
4	5	4
3	4	3

Fig. 7.

3 System Calibration

3.1 Unified Projection Model for a Fish-Eye Lens

The unified projection model [5] was proposed to model the projection of omnidirectional cameras such as a camera with a normal lens and hyperbolic or parabolic mirror, and a camera with a fish-eye lens. Then a calibration method [11] with a lens distortion model [17] was proposed.

This subsection briefly describes the fish-eye lens calibration method with the lens distortion model. In the unified projection model, an object in 3D space is projected in the image plane according to the following four steps (refer Fig. 8):

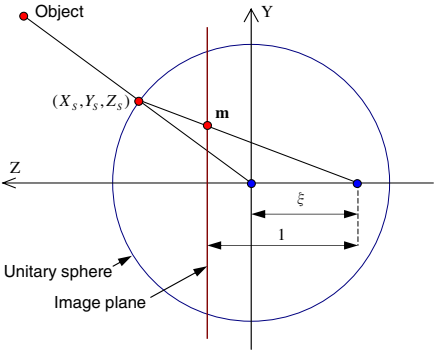


Fig. 8. Unified projection model

(1) Projection onto a unitary sphere: An object $\mathbf{X} = (X, Y, Z)$ is projected onto a unitary sphere surface with its center at the coordinate origin. The projection origin is also the center of the sphere.

$$(X_s, Y_s, Z_s) = \frac{\mathbf{X}}{\|\mathbf{X}\|} \tag{6}$$

(2) Projection onto a normalized plane: The coordinate system origin is set to $(0, 0, -\xi)$. The projected object on the unitary sphere is then projected to

[2] The common angles Ω_2 and Ω_3 are useful for far distant objects; they vary according to the object distance. We are seeking a better mirror design for actual measuring situations.

a normalized plane that is orthogonal to the Z axis at a unit distance from the new origin.

$$\mathbf{m} = (x, y, z)^\top = \left(\frac{X_s}{Z_s + \xi}, \frac{Y_s}{Z_s + \xi}, 1 \right)^\top \tag{7}$$

(3) Considering the lens distortion: The following radial and tangential lens distortions are considered.

$$\rho \rightarrow \rho(1 + k_1\rho^2 + k_2\rho^4 + k_5\rho^6), \quad \rho = \sqrt{x^2 + y^2} \tag{8}$$

$$x \rightarrow x + 2k_3xy + k_4(3x^2 + y^2) \tag{9}$$

$$y \rightarrow y + k_3(x^2 + 3y^2) + 2k_4xy \tag{10}$$

(4) Projection onto an image plane: The distorted object is then projected onto an image plane with the following intrinsic camera parameters.

$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} \tag{11}$$

3.2 Lens Calibration

We employ a MATLAB implementation [12] of the calibration method using the unified projection model [11] described previously. Figure 9 shows an image example for the calibration. The whole process of calibration is as follows.
(1) Initialize the calibration parameter as $\{\xi, k_1, k_2, k_3, k_4, k_5, f_u, f_v, u_0, v_0\} = \{1, 0, 0, 0, 0, 0, f, f, \hat{u}_0, \hat{v}_0\}$, where $(\hat{u}_0, \hat{v}_0)$ is the half size of the image.
(2) An initial focal length f is estimated using at least three user-defined [3] points on a line.
(3) The user specifies four corners of the calibration target in the image.
(4) The fish-eye lens projection function (intrinsic parameter) and the relative position and orientation of the calibration target are estimated using an optimization method.

One calibration target (a checkerboard) can be taken by the proposed system as five targets with different positions and orientations. These five targets can be considered as five observations of a single target; a single image including five targets is sufficient to estimate all calibration parameters. However, the target position for five such observations is limited to a specific region in the image, as described in subsection 2.3. The calibration accuracy is insufficient because biased regions are used in the circular fish-eye image.

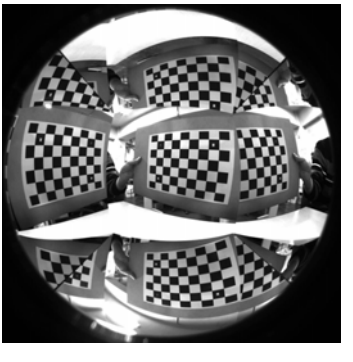


Fig. 9. An image used for calibration

[3] Four points are minimum in the MATLAB implementation.

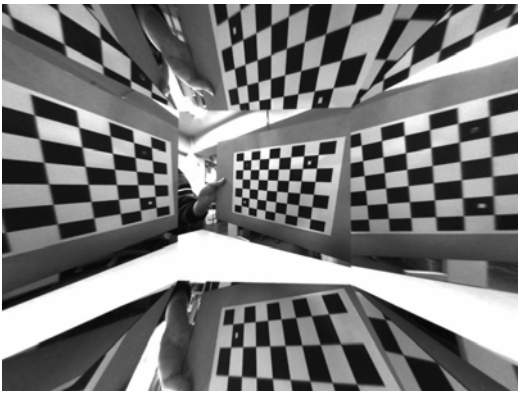


Fig. 10. Perspective reprojection of Fig. 9

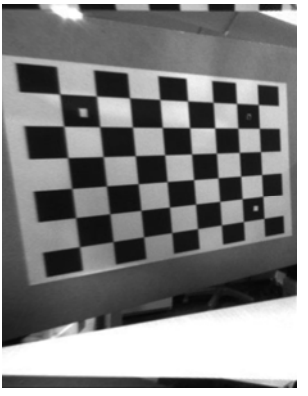


Fig. 11. Another perspective reprojection of Fig. 9

In our calibration, the target was placed evenly in the image, irrespective of whether it is a direct or a mirrored observation. The lens calibration was performed using 23 such observations.

The estimated parameters are $\{\xi, k_1, k_2, k_3, k_4, k_5, f_u, f_v, u_0, v_0\} = \{1.6988, -0.06093, 0.18404, -0.00015, -0.00017, 0, 871.54278, 868.49105, 791.49429, 595.47177\}$ for an image size of 1600×1200 [pixel].

Figure 10 shows a perspective reprojection of the image shown in Fig. 9 using the estimated lens parameters. Figure 11 shows another perspective reprojection to a different plane from one in Fig. 10. It is readily apparent that the fish-eye lens distortions are perfectly compensated for a heavily distorted part in Fig. 9.

3.3 Position and Orientation of the Mirrored Cameras

The proposed system can capture the images of one real camera and four mirrored cameras in a single image. This subsection explains the extrinsic parameter calibration between these cameras.

Calibration Target: In a similar fashion to that of a well known calibration tool [3] for a perspective camera, the extrinsic parameters are estimated iteratively with intrinsic parameters by considering that a stationary target is observed from different camera locations.

Mirrors: The mirror position and orientation can be determined by the obtained extrinsic parameters between a real and mirrored target.

As depicted in Fig. 12, the mirror position and orientation $[\mathbf{n}_i^m, d_i^m]$ that reflect the i-th mirrored target are obtainable as follows.

$$[\mathbf{n}_i^m, d_i^m] = \left[\frac{\mathbf{n}_0^t - \mathbf{n}_i^t}{|\mathbf{n}_0^t - \mathbf{n}_i^t|}, \frac{(\mathbf{n}_0^t - \mathbf{n}_i^t) \cdot \mathbf{n}_i^t}{|\mathbf{n}_0^t - \mathbf{n}_i^t| \, (1 - \mathbf{n}_0^t \cdot \mathbf{n}_i^t)} \left(d_i^t - d_0^t\right) \right] \tag{12}$$

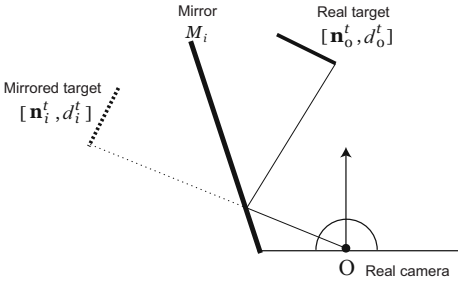


Fig. 12. Real (0-th) and mirrored (i-th) targets

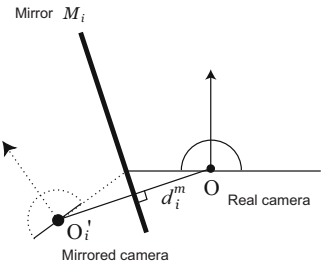


Fig. 13. Real and mirrored cameras

In that equation, $[\mathbf{n}_i^t, d_i^t]$ denotes a set of the target normal $\mathbf{n}_i^t$ and the distance from the optical center d_i^t, as the position and orientation of the i-th target.

As described before, only a single image including the five targets is sufficient to estimate the four mirror positions and orientations, but they are estimated by minimizing the sum of the reprojection error of many target positions.

Mirrored Cameras: As depicted in Fig. 13, the optical center O_i' and the rotation matrix $\mathbf{R}_i$ of the i-th mirrored camera is obtainable using the estimated i-th mirror position and orientation, as follows [6].

$$O_i' = -2d_i^m \mathbf{n}_i^m \tag{13}$$

$$\mathbf{R}_i = \mathbf{I} - 2\mathbf{n}_i^m \mathbf{n}_i^{m\top} \tag{14}$$

4 Single-Camera Multi-baseline Stereo

The five observations from different positions and orientations that have been estimated by the calibration are acceptable for use with the stereo method, especially the multi-baseline stereo method [13]. This section presents two range estimation methods using the proposed system.

4.1 Perspective Reprojection of the Fish-Eye Image

In the first approach, we create five perspective projection images used for the multi-baseline stereo. The view angle of a fish-eye lens is very wide (180 degree); it is impossible to convert the whole image taken by a fish-eye lens to a single perspective projection image. Each view of the real and mirrored cameras should be converted separately to perspective projection images. The division of the whole circular fish-eye image to the five observations is done manually; it is done only once for manual division because the mirror positions and orientations are stationary to the fish-eye lens.

The conversion has two steps. The first step converts the fish-eye projected image to incident and azimuthal angles using the estimated projection function

of the lens. The second step reprojects the angles to a perspective projected image.

The optical axis of the mirrored camera has an offset angle θ to the axis of real camera, as described in Eq. (3). Moreover, the central direction of view angle of the mirrored camera differs from the real camera. In our reprojection, the perspective projection plane is placed parallel to the projection plane for the real camera, as shown in Fig. 14. The placement of the five cameras is therefore a parallel stereo with an anteroposterior offset.

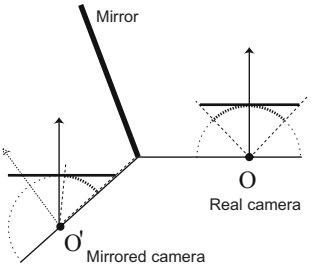


Fig. 14. Projection planes for the reprojection

The epipolar constraints for the four mirrored cameras in their reprojection images are derived from the extrinsic parameters estimated in the calibration. The multi-baseline stereo method [13] is applicable to the proposed system by evaluating SSSD (sum of the sum of squared differences) between a small ROI (region of interest) set in the centered real image and small ROIs on the constraint lines in the converted mirrored camera images, with respect to the object range.

4.2 Matching in the Fish-Eye Image

The preceding subsection described a method to convert the images. This subsection explains a method not to convert the images, but to convert the epipolar constraint in the fish-eye image for matching directly in the fish-eye image.

Figure 15 depicts a stereo camera pair with a fish-eye lens, which sees a point P from the optical centers O and O'. The following epipolar plane Π_e includes the three points P, O, and O' for this situation.

$$(\overrightarrow{OP} \times \overrightarrow{OO'}) \cdot \mathbf{x} = (\mathbf{p} \times (-2d^m \mathbf{n}^m)) \cdot \mathbf{x} = 0 \tag{15}$$

Therein, $\mathbf{x} = (x, y, x)$ and $\mathbf{p}$ respectively signify a point on the plane Π_e and the position of P.

Consider that the camera O detects an object P that changes the distance from the camera O. As the distance changes, the spheral projection of point P moves along the intersection of the epipolar plane Π_e and the unitary sphere with its center at O' [2]. A point $\mathbf{x} = (x, y, z)$ on the unitary sphere with its center at O' is representable as follows.

$$(\mathbf{x} - O') \cdot (\mathbf{x} - O') = (\mathbf{x} + 2d^m \mathbf{n}^m) \cdot (\mathbf{x} + 2d^m \mathbf{n}^m) = 1 \tag{16}$$

Then the epipolar constraint curve is obtainable as the projection by the calibrated fish-eye projection function of $\mathbf{x}$, which satisfies both Eqs. (15) and (16). Figure 16 presents examples of the epipolar curves.

Specifically, the epipolar curve can be determined as follows.

(1) The incident and azimuthal angle of point $\mathbf{x} = (x, y, x)$, which satisfies both Eqs. (15) and (16), is obtainable, as

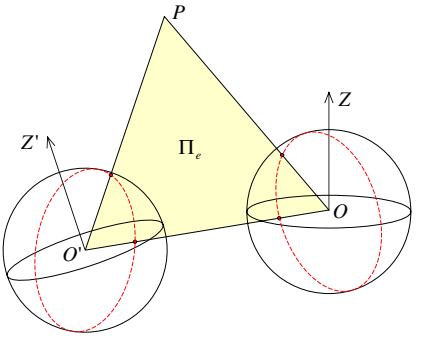


Fig. 15. Unitary spheres for the fish-eye projection and their epipolar plane

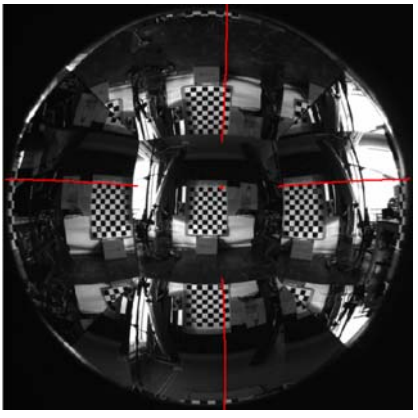


Fig. 16. Epipolar curves in mirrored images

$$\begin{cases} \phi = \tan^{-1}\dfrac{y}{x} \\ \theta = \dfrac{\pi}{2} - \sin^{-1}\dfrac{z}{\sqrt{x^2+y^2+z^2}} \end{cases}, \tag{17}$$

where θ denotes the angle to the Z axis.

(2) Then Eq. (17) is projected to the image using the calibrated projection function $\hat{\rho}(\theta)$ and image center (u_0, v_0), as

$$\begin{cases} u = u_0 + \hat{\rho}(\theta)\cos\phi \\ v = v_0 + \hat{\rho}(\theta)\sin\phi \end{cases}, \tag{18}$$

where $\hat{\rho}(\theta)$ includes the lens distortion compensation.

An epipolar curve is obtainable for each mirrored camera. As described in the preceding subsection, the multi-baseline stereo method [13] is applicable to the proposed system by evaluating the SSSD between a small region of interest (ROI) set in the centered real image and small ROIs on the constraint lines in the converted mirrored camera images, with respect to the object range.

5 Experimental Results

Figure 17 shows the perspective reprojected images converted from Fig. 2 using the method described in 4.1.

Figure 18 (left) shows the perspective reprojected image of the real camera. Figure 18 (center) and 18 (right) show the range estimation results using the perspective reprojection method described respectively in 4.1 and the constraint in the fish-eye image method described in 4.2. The ROI size is 7×7 [pixel] for both methods. Both results are almost the same because the only difference is the reprojection of the image or the projection of the constraint. In both results, the detailed 3D shapes are clearly recovered such as the edge of leaves in front of the bear.

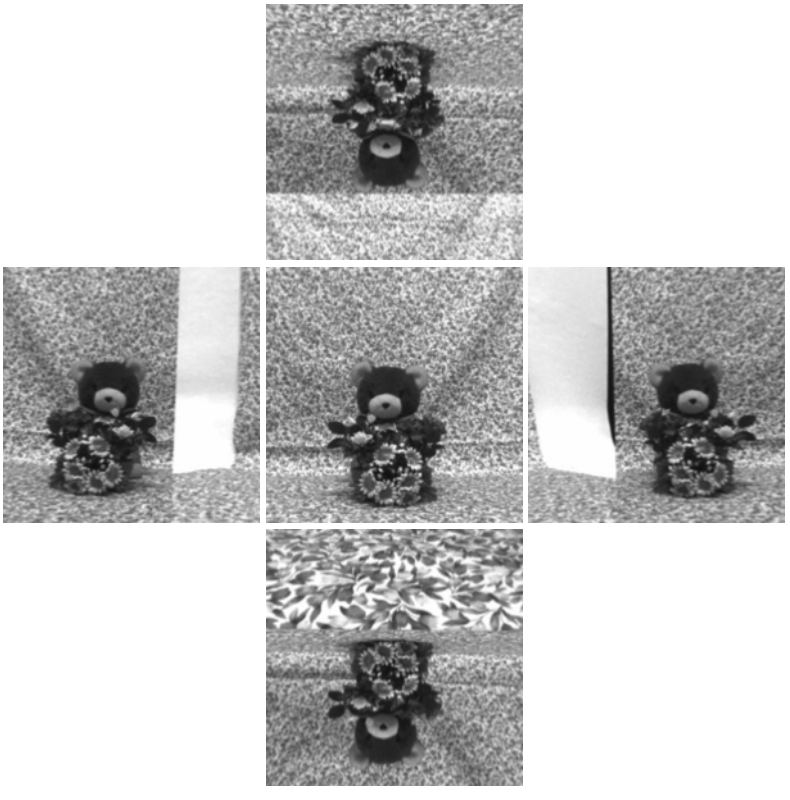


Fig. 17. Perspective reprojected images from Fig. 2

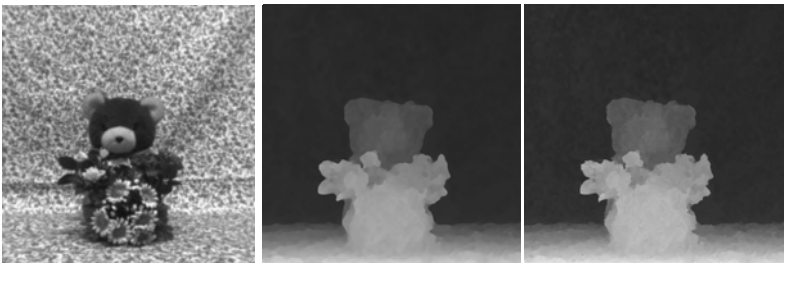


Fig. 18. Perspective reprojected image of the real camera (left), and range estimation results (center and right). The center and right images are results obtained respectively using the methods described in 4.1 and 4.2.

6 Conclusions

This paper presented our proposed monocular range measurement system with a fish-eye lens and mirrors that are placed around the lens. The captured

image can be used for the direct observation of a target with the centered region. Simultaneously, it can be used for the multi-baseline stereo to reconstruct three-dimensional information. After calibration of projection function of the fish-eye lens, the mirror positions and orientations are obtainable from external parameters that are used for the multi-baseline stereo measurement. Experimental results demonstrate the effectiveness of a real working system.

Future studies will address size reduction of the system and the efficiency of computations.

References

1. Adelson, E.H., Wang, J.Y.A.: Single lens stereo with a plenoptic camera. PAMI 14(2), 99–106 (1992)
2. Barreto, J.P., Daniilidis, K.: Epipolar geometry of central projection systems using Veronese maps. In: CVPR (2006)
3. Bouguet, J.-Y.: Camera calibration toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
4. Courbon, J., Mezouar, Y., Eck, L., Martinet, P.: A generic fisheye camera model for robotic applications. In: Proc. of Intelligent Robots and Systems, pp. 1683–1688 (2007)
5. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 445–461. Springer, Heidelberg (2000)
6. Gluckman, J., Nayar, S.K.: Catadioptric stereo using planar mirrors. IJCV 44(1), 65–79 (2001)
7. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. In: ICCV, pp. 108–115 (2001)
8. Hiura, S., Matsuyama, T.: Depth measurement by the multi-focus camera. In: CVPR, pp. 953–959 (1998)
9. Kuthirummal, S., Nayar, S.K.: Multiview radial catadioptric imaging for scene capture. ACM Trans. on Graphics 25(3), 916–923 (2006)
10. Lee, D.-H., Kweon, I.-S.: A novel stereo camera system by a biprism. Robotics and Automation 16(5), 528–541 (2000)
11. Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: ICRA (2007)
12. http://www.robots.ox.ac.uk/~cmei/
13. Okutomi, M., Kanade, T.: A multiple-baseline stereo. PAMI 15(4), 353–363 (1993)
14. Ramalingam, S., Sturm, P., Lodha, S.: Towards complete generic camera calibration. In: CVPR, pp. 1093–1098 (2005)
15. Sagawa, R., Kurita, N., Echigo, T., Yagi, Y.: Compound catadioptric stereo sensor for omnidirectional object detection. In: ICIRS, pp. 2612–2617 (2004)
16. Shimizu, M., Okutomi, M.: Calibration and rectification for reflection stereo. In: CVPR (2008)
17. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. PAMI 14(10), 965–980 (1992)

Generation of an Omnidirectional Video without Invisible Areas Using Image Inpainting

Norihiko Kawai, Kotaro Machikita, Tomokazu Sato, and Naokazu Yokoya

Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{norihi-k,tomoka-s,yokoya}@is.naist.jp
http://yokoya.naist.jp/

Abstract. Omnidirectional cameras usually cannot capture the entire direction of view due to a blind side. Thus, such an invisible part decreases realistic sensation in a telepresence system. In this study, an omnidirectional video without invisible areas is generated by filling in the missing region using an image inpainting technique for highly realistic sensation in telepresence. This paper proposes a new method that successfully inpaints a missing region by compensating for the change in appearance of textures caused by the camera motion and determining a searching area for similar textures considering the camera motion and the shape of the scene around the missing region. In experiments, the effectiveness of the proposed method is demonstrated by inpainting missing regions in a real image sequence captured with an omnidirectional camera and generating an omnidirectional video without invisible areas.

1 Introduction

Telepresence systems that enable us to experience a remote site are expected to be used in various fields such as entertainment and education. In these fields, omnidirectional videos captured with a moving omnidirectional camera are sometimes used [1,2]. However, an ordinary omnidirectional camera cannot capture the entire direction of view due to a blind side as shown in Fig. 1. Thus, such an invisible part decreases realistic sensation in telepresence. In order to achieve telepresence with highly realistic sensation, this research aims at generating an omnidirectional video without invisible areas by inpainting the missing region caused by the blind side. Conventionally, many image inpainting methods for a still image have been proposed [3,4,5]. Missing regions in not only a still image but also a video can be filled in by applying these methods to each frame in a video. However, textures may discontinuously change between successive frames because the methods use only information in a frame.

On the other hand, methods that fill in missing regions in a video considering temporal continuity have been proposed [6,7,8,9,10,11]. These methods are classified into two categories. One uses the motion information of a scene in

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 359–370, 2010.

Fig. 1. Omnidirectional panorama image with missing region (black region) caused by the blind side

an image sequence [6,7,8,9] and the other does not [10,11]. The former method specifies the appropriate textures for missing regions by calculating the motion of objects in a video or the motion of a camera and fills in the missing regions using the specified texture. The latter method searches whole the video for the spatial-temporal volume similar to that around missing regions and fills in the missing regions using the similar volumes. Both methods can generate a video with temporally continuous change in texture. However, these methods do not consider the change in the appearance of textures caused by the camera motion. Therefore, it is difficult for these methods to successfully inpaint missing regions in an omnidirectional video caused by the blind side of an omnidirectional camera because the appearance of the texture appropriate for a missing region in a frame changes in different frames of a moving omnidirectional camera.

To overcome these problems, this paper proposes a new method that successfully inpaints a missing region compensating for the change in the appearance of textures. Concretely, by assuming that the shape of the blind side of the target scene is planar, the change in the appearance of the texture caused by the camera motion is compensated by projecting omnidirectional images onto the planar surface fitted to the 3-D positions of natural feature points on the ground acquired by structure-from-motion (SFM). In addition, by using the fitted plane and the camera motion, the data region in which appropriate textures for missing regions may exist is determined. Finally, good quality images are obtained by using an image inpainting technique. In this research, we employ an omnidirectional multi-camera system (OMS) that is composed of radially arranged multiple cameras and we assume that the ground exists in the direction of the blind side of a moving OMS.

2 Generation of an Omnidirectional Video without Invisible Areas

The flow of the proposed method is as follows. (a) The position and posture of an OMS and 3-D positions of natural feature points are estimated using SFM for

an omnidirectional video. (b) A plane for each frame is fitted to natural feature points near the ground by using the position and the posture of the OMS and the 3-D positions of natural feature points. (c) An image sequence projected on the fitted plane is generated from the omnidirectional video. (d) Data regions in which appropriate textures for missing regions may exist are specified on the projected image plane using the position and posture of the OMS and the fitted planes. (e) A missing region in the projected image plane of each frame is successively inpainted by minimizing an energy function based on the similarity between the texture in the missing region and the specified data region. (f) An omnidirectional video without invisible areas is generated by re-projecting the inpainted image onto the omnidirectional panoramic video with a missing region. In the following sections, each process is described in detail.

2.1 Estimation of Extrinsic Camera Parameters and Positions of Natural Feature Points

The position and posture of an OMS and 3-D positions of natural feature points are estimated by SFM [12] for an omnidirectional image. In this method, first, a target scene is captured with a moving OMS. Next, initial extrinsic camera parameters and 3-D positions of feature points are estimated by tracking the natural feature points in a video, which are detected by Harris operator. Finally, the accumulative errors of the camera parameters and the 3-D positions of feature points are minimized by bundle adjustment for whole the video.

2.2 Generation of Images Projected on Planes by Estimating Shapes Around Missing Regions

In this research, on the assumption that an omnidirectional video is captured while moving on the ground and the shape around a missing region is planar, an image sequence that includes missing regions is generated by projecting the omnidirectional video to the planes in order to compensate for the change in the appearance of textures caused by the camera motion.

Concretely, first, natural feature points for plane fitting are selected from the points obtained by SFM described in Section 2.1. Here, the points that satisfy the following conditions are selected: (i) a point exists in the spherical area whose center is a projection center of a representative camera unit of an OMS and radius is l, and (ii) the height z of a point in the world coordinate system is $(p < z < p+m)$ (p and m are constants) as shown in Fig 2. Next, the expression of the plane that represents the ground in the world coordinate system is set as $z = ax + by + c$, and the parameters (a, b, c) are determined by the least-square method so as to minimize the following cost function L.

$$L = \sum_{i=1}^{n} (ax_i + by_i + c - z_i)^2, \tag{1}$$

where (x_i, y_i, z_i) are the coordinates of a feature point and n is the number of selected feature points. An image sequence is generated by projecting the

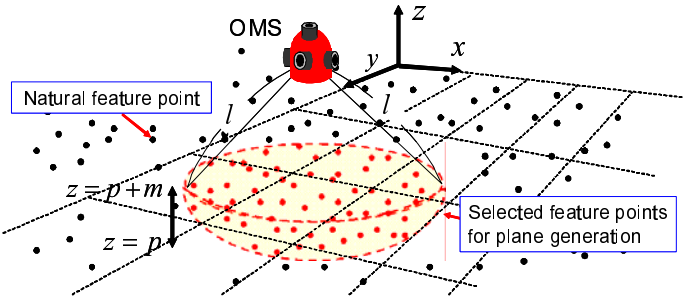


Fig. 2. Selection of feature points around missing region

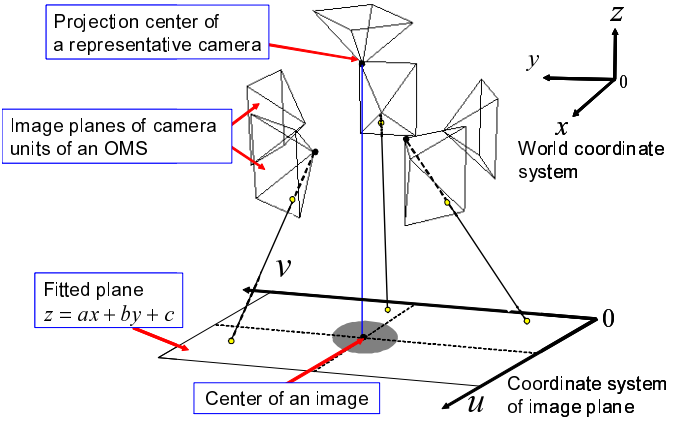


Fig. 3. Generation of an image projected on a plane

omnidirectional video to the estimated plane for each frame as shown in Fig. 3. Here, in order for a missing region to be the center of the projected image, an intersection point of the plane with the straight line that goes just under an OMS through the projection center of a representative camera of the OMS is set as the center of the image. Additionally, in order to prevent the rotation of the textures in projected image planes, the basis vectors $(\mathbf{u}, \mathbf{v})$ of the image in the world coordinate system are set so as to satisfy the following equation.

$$\mathbf{u} \cdot \mathbf{y} = 0, \tag{2}$$

where $\mathbf{y}$ is one of the basis vectors of the world coordinate system.

2.3 Inpainting a Missing Region Based on Energy Minimization

A missing region in each frame is successively inpainted by applying an energy minimization method to each image projected on a plane with a missing region generated by the method described in the previous section. In the following,

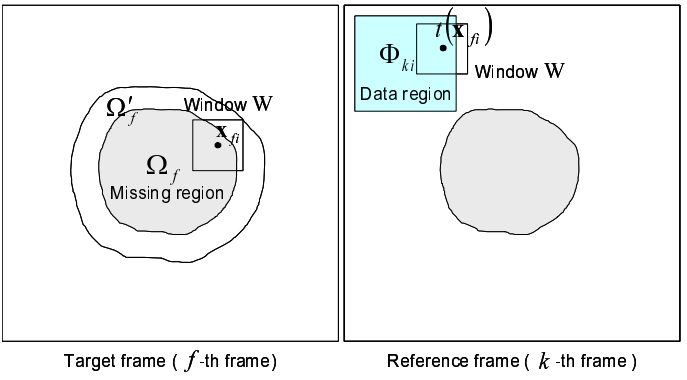


Fig. 4. Missing and data regions in projected images for inpainting process

the definition of an energy function, a method determining a data region and a method minimizing the energy function are described.

Definition of an energy function. As shown in Fig. 4, a missing region in the projected image of the f-th frame (target frame) is inpainted using an energy function based on the similarity of textures between region Ω'_f including missing region Ω_f in the f-th frame and data region Φ_{ki} in the k-th frame (reference frame) ($k \neq f$). Here, Ω'_f is the expanded area of the missing region Ω_f in which there is a central pixel, $\mathbf{x}_{fi}$, of a square window W overlapping region Ω_f and each data region Φ_{ki} corresponding to each pixel $\mathbf{x}_{fi}$ in the f-th frame is individually determined. Energy function E is defined as the weighted sum of SSD (Sum of Squared Differences) between the textures around pixel $\mathbf{x}_{fi}$ in region Ω'_f and $t(\mathbf{x}_{fi})$ in data region Φ_{ki}.

$$E = \sum_{\mathbf{x}_{fi} \in \Omega'_f} w_{\mathbf{x}_{fi}} SSD(\mathbf{x}_{fi}, t(\mathbf{x}_{fi})), \tag{3}$$

where $w_{\mathbf{x}_{fi}}$ is the weight for pixel $\mathbf{x}_{fi}$ and is set as 1 if $\mathbf{x}_{fi}$ is inside of region $\Omega'_f \cap \overline{\Omega_f}$ because pixel values in this region are fixed; otherwise $w_{\mathbf{x}_{fi}} = g^{-d}$ (d is the distance from the boundary of Ω_f and g is a constant) because pixel values around the boundary have higher confidence than those in the center of the missing region.

$SSD(\mathbf{x}_{fi}, t(\mathbf{x}_{fi}))$, which represents the similarity of textures around pixel $\mathbf{x}_{fi}$ and $t(\mathbf{x}_{fi})$, is defined as follows:

$$SSD(\mathbf{x}_{fi}, t(\mathbf{x}_{fi})) = \sum_{\mathbf{q} \in W} \{I(\mathbf{x}_{fi} + \mathbf{q}) - \alpha_{\mathbf{x}_{fi} t(\mathbf{x}_{fi})} I(t(\mathbf{x}_{fi}) + \mathbf{q})\}^2, \tag{4}$$

where $I(\mathbf{x})$ represents the pixel value of pixel $\mathbf{x}$. $\alpha_{\mathbf{x}_{fi} t(\mathbf{x}_{fi})}$ is the intensity modification coefficient. Note that textures around a missing region may change due to the reflection of the light on the ground and the shadow of the camera and operator. Therefore, by using this coefficient, the brightness of textures in data

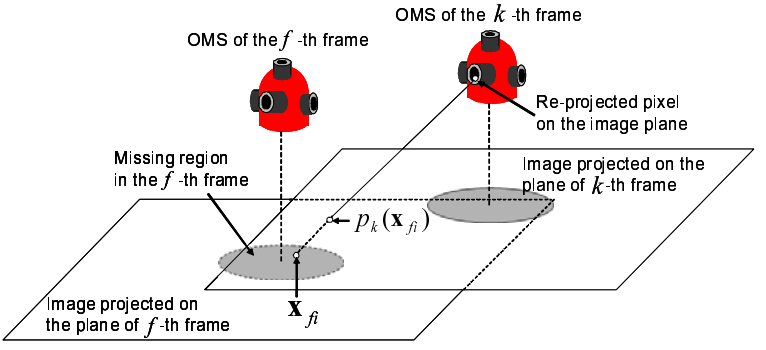


Fig. 5. Projection to the other frame

regions is adjusted to that in the missing region. In this research, $\alpha_{\mathbf{x}_{fi}t(\mathbf{x}_{fi})}$ is defined as the ratio of average pixel values around pixels $\mathbf{x}_{fi}$ and $t(\mathbf{x}_{fi})$ as follows:

$$\alpha_{\mathbf{x}_{fi}t(\mathbf{x}_{fi})} = \frac{\sqrt{\sum_{\mathbf{q}\in W} I(\mathbf{x}_{fi}+\mathbf{q})^2}}{\sqrt{\sum_{\mathbf{q}\in W} I(t(\mathbf{x}_{fi})+\mathbf{q})^2}}. \tag{5}$$

Determination of a data region. A data region in which position $t(\mathbf{x}_{fi})$ of the most similar texture pattern may exist is determined by using the position and posture of a moving OMS estimated in Section 2.1 and the planes generated in Section 2.2. In this research, appropriate textures for the missing region in the f-th frame are expected to be captured in the frames other than the target frame by assuming that the omnidirectional video is captured while moving. Additionally, the parameters of the plane and the position and posture of the OMS in each frame are known. Therefore, regions in which the most similar pattern exists can be determined in the frames other than the target frame by using the geometric relationships of a moving camera system and the ground. Also, an appropriate frame is determined considering the resolution of the similar texture pattern. In the following, we describe the way to determine a region and a frame that are used as a data region for the energy minimization process described in the following section.

First, the 3-D coordinate of pixel $\mathbf{x}_{fi}$ in the target (f-th) projected image is re-projected on the image plane of a camera unit of the OMS in the k-th frame. Then, the pixel coordinate $p_k(\mathbf{x}_{fi})$ of the intersection of the k-th projected image with the straight line that goes through the re-projected pixel on the image plane of the camera unit and pixel $\mathbf{x}_{fi}$ on the f-th projected image is calculated as shown in Fig. 5. In a similar way, pixel coordinate $p_k(\mathbf{x}_{fi})$ in each frame k corresponding to pixel $\mathbf{x}_{fi}$ is calculated. Next, a frame is selected considering the position of $p_k(\mathbf{x}_{fi})$ in a projected image and the difference of frames between the target and the reference frames. In projected images, the resolution of texture becomes lower the farther a pixel is from the center of the image because textures

of objects remote from the camera become small in input images of an OMS. In order to prevent the generation of blurred textures, textures near the center of the image should be used as samples for inpainting. In addition, it is highly possible that temporally close frames have similar brightness of textures. Therefore, the appropriate frame $s(\mathbf{x}_{fi})$ is selected from candidate frames $\mathbf{K} = (k_1, \ldots, k_n)$ by the following equation.

$$s(\mathbf{x}_{fi}) = \underset{k \in \mathbf{K}}{\operatorname{argmin}}(\|p_k(\mathbf{x}_{fi}) - \mathbf{x}_{center}\| + \lambda|k - f|), \tag{6}$$

where candidate frames $\mathbf{K}$ are picked up so that the fixed range of the texture around $p_k(\mathbf{x}_{fi})$ does not include the missing region. $\mathbf{x}_{center}$ is the central pixel in the k-th planar projected image and λ is the weight for the difference of frames. Finally, fixed square area S whose center is pixel $p_{s(\mathbf{x}_{fi})}(\mathbf{x}_{fi})$ is set as a data region $\Phi_{s(\mathbf{x}_{fi})i}$, which is used for the energy minimization process described in the following section. In a similar way, each data region $\Phi_{s(\mathbf{x}_{fi})i}$ corresponding to each pixel $\mathbf{x}_{fi}$ in expanded missing region Ω'_f is individually determined.

Energy minimization. Energy function E in Eq. (3) is minimized by using a framework of greedy algorithm in a similar way to [13]. In our definition of energy E, the energy for each pixel can be treated independently if pattern pairs $(\mathbf{x}_{fi}, t(\mathbf{x}_{fi}))$ can be fixed and the change of coefficient $\alpha_{\mathbf{x}_{fi}t(\mathbf{x}_{fi})}$ in the iterative process of energy minimization is very small. Thus, we repeat the following two processes until the energy converges: (i) search for the most similar pattern keeping pixel values fixed, and (ii) perform a parallel update of all pixel values keeping pattern pairs fixed.

In process (i), data region Φ_{ki} $(k = s(\mathbf{x}_{fi}))$ is searched for position $t(\mathbf{x}_{fi})$ of the most similar pattern keeping pixel values $I(\mathbf{x}_{fi})$ fixed. $t(\mathbf{x}_{fi})$ is determined as follows:

$$t(\mathbf{x}_{fi}) = \underset{\mathbf{x} \in \Phi_{ki}}{\operatorname{argmin}}(SSD(\mathbf{x}_{fi}, \mathbf{x})). \tag{7}$$

In process (ii), all pixel values $I(\mathbf{x}_{fi})$ are updated in parallel so as to minimize the energy keeping the similar pattern pairs fixed. In the following, the method for calculating pixel values $I(\mathbf{x}_{fi})$ is described. First, energy E is resolved into element energy $E(\mathbf{x}_{fi})$ for each pixel $\mathbf{x}_{fi}$ in missing region Ω_f. Element energy $E(\mathbf{x}_{fi})$ can be expressed in terms of the pixel values of $\mathbf{x}_{fi}$ and $f(\mathbf{x}_{fi} + \mathbf{q}) - \mathbf{q}$, coefficient α as follows:

$$E(\mathbf{x}_{fi}) = \sum_{\mathbf{q} \in W} w_{(\mathbf{x}_{fi}+\mathbf{q})} \{I(\mathbf{x}_{fi}) - \alpha_{(\mathbf{x}_{fi}+\mathbf{q})t(\mathbf{x}_{fi}+\mathbf{q})} I(t(\mathbf{x}_{fi} + \mathbf{q}) - \mathbf{q})\}^2. \tag{8}$$

The relationship between energy E and element energy $E(\mathbf{x}_{fi})$ for each pixel can be written as follows:

$$E = \sum_{\mathbf{x}_{fi} \in \Omega} E(\mathbf{x}_{fi}) + C. \tag{9}$$

C is the energy of pixels in region $\Omega'_f \cap \overline{\Omega_f}$, and is treated as a constant because pixel values in the region and all pattern pairs are fixed in process (ii). Therefore,

by minimizing element energy $E(\mathbf{x}_{fi})$ respectively, total energy E can be minimized. Here, if it is assumed that the change of $\alpha_{\mathbf{x}_{fi}t(\mathbf{x}_{fi})}$ is much smaller than that of pixel value $I(\mathbf{x}_{fi})$, by differentiating $E(\mathbf{x}_{fi})$ with respect to $I(\mathbf{x}_{fi})$, each pixel value $I(\mathbf{x}_{fi})$ in missing region Ω_f can be calculated in parallel as follows:

$$I(\mathbf{x}_{fi}) = \frac{\sum_{\mathbf{q}\in W} w_{(\mathbf{x}_{fi}+\mathbf{q})}\alpha_{(\mathbf{x}_{fi}+\mathbf{q})t(\mathbf{x}_{fi}+\mathbf{q})}I(t(\mathbf{x}_{fi}+\mathbf{q})-\mathbf{q})}{\sum_{\mathbf{q}\in W} w_{(\mathbf{x}_{fi}+\mathbf{q})}}. \tag{10}$$

In addition, a coarse-to-fine approach is also employed for energy minimization. Concretely, an image pyramid is generated and processes (i) and (ii) are repeated from higher-level to lower-level layers successively. This makes it possible to decrease computational cost and avoid local minima.

2.4 Generation of an Omnidirectional Video Using Inpainted Images

An omnidirectional video without invisible areas is generated by re-projecting the projected images inpainted in the previous section onto spherical panoramic images with a missing region. Concretely, first, the coordinate of the intersection of the plane with the straight line that goes through the projection center of a camera unit and each pixel in the missing region in the spherical panoramic image is calculated. Next, the pixel value of the calculated coordinate in the projected image is copied to the panoramic image.

3 Experiments

In this section, the effectiveness of the proposed method is demonstrated by inpainting a missing region caused by the blind side of an OMS and generating an omnidirectional video without invisible areas. In the following, the experiment of inpainting for images projected on images is described and a prototype telepresence system using the omnidirectional video without invisible areas is presented.

3.1 Inpainting a Missing Region in an Omnidirectional Video

In this experiment, we used Ladybug [14] as an OMS that is composed of 6 camera units and an omnidirectional image sequence (300 frames) is captured. Figure 6 shows the 1st frame of 6 image sequences captured with Ladybug. The position and posture of Ladybug and the positions of natural feature points were obtained by SFM [12] described in Section 2.1. A missing region in each projected image is determined by manually specifying the region in 6 images of the first frame. In addition, a blind region in the projected image is also specified as the missing region.

First, as shown in Fig. 7, images projected on planes were generated by the method described in Section 2.2. The resolution of a projected image was set

Fig. 6. 1st frame of input image sequence obtained by 6 camera units

1st frame 51st frame 101st frame

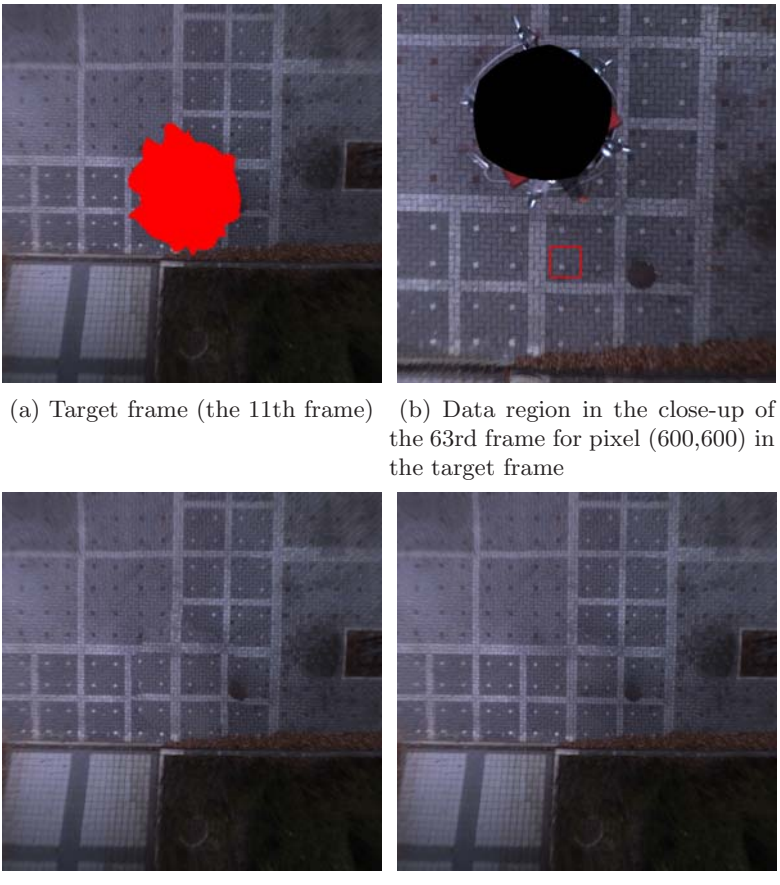
Fig. 7. Images projected on planes

as 1200 × 1200 pixels. Round black regions in the images are missing regions caused by the blind side of Ladybug. As shown in these figures, textures of tiles on the ground are uniform regardless of the position of pixels and textures of the same objects do not rotate in each frame. As a result, appropriate projected images used for inpainting were generated.

Next, a missing region in each projected image was inpainted. Figure 8 shows the experiment of inpainting for the projected image of the 11th frame. Figure 8(a) shows the target 11th frame in which the missing region is specified and Fig. 8(b) shows the data region in the close-up of the 63rd frame corresponding to pixel (600,600) in the target frame. Figure 8(c) shows the result by projecting pixel values in other frames onto the missing region in the target frame using the position and posture of Ladybug and the generated plane without the inpainting process. From this figure, the geometrical and optical disconnect of textures in the boundary of the missing region appears. We consider this is because of the errors of the estimation of camera parameters by SFM and errors of plane fitting. On the other hand, in the resultant image by the proposed method as shown in Fig. 8(d), textures continuously connect on the boundary and plausible textures are generated in the missing region. Fig. 9 shows the inpainted images corresponding to Fig. 7. In each frame, the missing region is successfully inpainted.

3.2 Omnidirectional Telepresence without Invisible Areas

In this experiment, the effectiveness of the proposed method is demonstrated by making the telepresence system using an omnidirectional video in which missing



(a) Target frame (the 11th frame)

(b) Data region in the close-up of the 63rd frame for pixel (600,600) in the target frame

(c) Result by projection using the plane

(d) Result by the proposed method

Fig. 8. Comparison of results by projection using a plane and proposed method

regions are filled in with inpainted images shown in the previous section. Figure 10 shows the omnidirectional panorama image without invisible areas generated by projecting the inpainted image (Fig. 9) onto the panoramic image (2048 ×1024 pixels). By using the panoramic image as input, we built an omnidirectional telepresence system. Figure 11 shows examples of user's views in the telepresence system. By comparison of the left and right images in Fig. 11, we can confirm that realistic sensation is drastically increased by the proposed method.

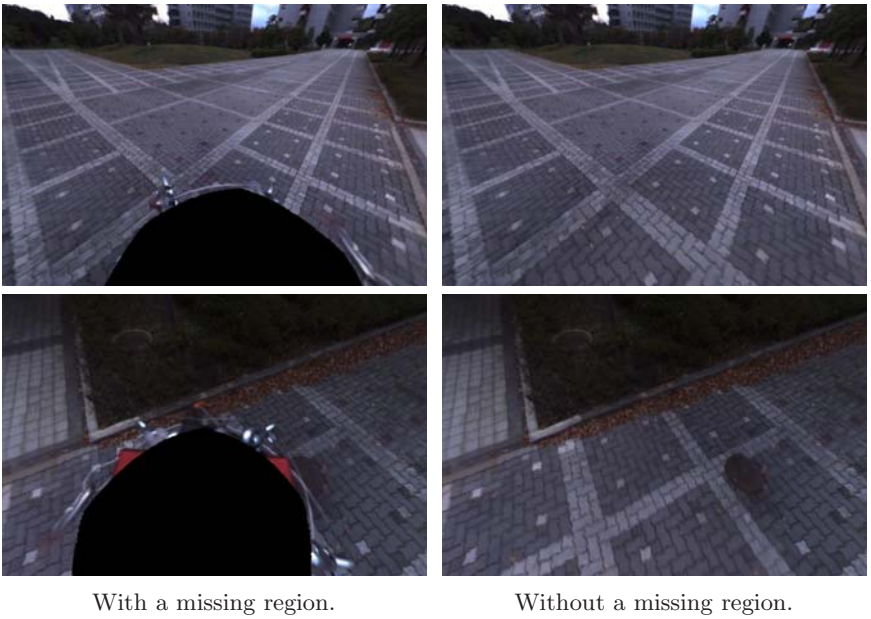
4 Conclusion

In this paper, we have proposed a method that generates an omnidirectional video without invisible areas by compensating for the change in the appearance of textures caused by the camera motion and determining a data region

1st frame 51st frame 101st frame

Fig. 9. Inpainted projected images (Corresponding to Fig. 7)

Fig. 10. Filled panoramic image of 1st frame (Corresponding to Fig. 1)



With a missing region. Without a missing region.

Fig. 11. Looking around using omnidirectional video

considering the camera motion and the shape of the scene around the missing region. In experiments, missing regions in images projected on planes were successfully inpainted and the omnidirectional telepresence without missing regions was achieved. In future work, we will perform experiments with various scenes. In addition, the proposed method will be evaluated quantitatively by using virtual environments.

References

1. Ikeda, S., Sato, T., Yokoya, N.: Immersive Telepresence System with a Locomotion Interface Using High-resolution Omnidirectional Videos. In: Proc. IAPR Conf. on Machine Vision Applications, pp. 602–605 (2005)
2. Hori, M., Kanbara, M., Yokoya, N.: Novel Stereoscopic View Generation by Image-Based Rendering Coordinated with Depth Information. In: Proc. Scandinavian Conf. on Image Analysis, pp. 193–202 (2007)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image Inpainting. In: Proc. ACM SIGGRAPH 2000, pp. 417–424 (2000)
4. Criminisi, A., Perez, P., Toyama, K.: Region Filling and Object Removal by Exemplar-Based Inpainting. IEEE Trans. on Image Processing 13, 1200–1212 (2004)
5. Komodakis, N., Tziritas, G.: Image Completion Using Global Optimization. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 442–452 (2006)
6. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.: Full-Frame Video Stabilization with Motion Inpainting. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(7), 1150–1163 (2006)
7. Jia, J., Tai, Y., Wu, T., Tang, C.: Video Repairing under Variable Illumination Using Cyclic Motions. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(5), 832–839 (2006)
8. Shen, Y., Lu, F., Cao, X., Foroosh, H.: Video Completion for Perspective Camera Under Constrained Motion. In: Proc. IEEE Int. Conf. on Pattern Recognition, pp. 63–66 (2006)
9. Patwardhan, K., Sapiro, G., Bertalmio, M.: Video Inpainting Under Constrained Camera Motion. IEEE Trans. on Image Processing 16, 545–553 (2007)
10. Wexler, Y., Shechtman, E., Irani, M.: Space-Time Completion of Video. Trans. on Pattern Analysis and Machine Intelligence 29, 463–476 (2007)
11. Cheung, V., Frey, B., Jojic, N.: Video epitomes. In: Proc. IEEE Conf. on Computer Vision and Patern Recognition, pp. 141–152 (2005)
12. Sato, T., Ikeda, S., Yokoya, N.: Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 326–340. Springer, Heidelberg (2004)
13. Kawai, N., Sato, T., Yokoya, N.: Image Inpainting Considering Brightness Change and Spatial Locality of Textures and Its Evaluation. In: Proc. Pacific-Rim Symp. on Image and Video Technology, pp. 271–282 (2009)
14. Point Grey Research Inc.: Ladybug, http://www.ptgrey.com/products/spherical.asp

Accurate and Efficient Cost Aggregation Strategy for Stereo Correspondence Based on Approximated Joint Bilateral Filtering

Stefano Mattoccia, Simone Giardino, and Andrea Gambini

Department of Electronics Computer Science and Systems (DEIS)
Advanced Research Center on Electronic Systems (ARCES)
University of Bologna, Viale Risorgimento 2, 40136 - Bologna, Italy
stefano.mattoccia@unibo.it, simone.giardino@hotmail.it,
gambini00@gmail.com

Abstract. Recent local state-of-the-art stereo algorithms based on variable cost aggregation strategies allow for inferring disparity maps comparable to those yielded by algorithms based on global optimization schemes. Unfortunately, thought these results are excellent, they are obtained at the expense of high computational requirements that are comparable or even higher than those required by global approaches. In this paper, we propose a cost aggregation strategy based on joint bilateral filtering and incremental calculation schemes that allow for efficient and accurate inference of disparity maps. Experimental comparison with state-of-the-art techniques shows the effectiveness of our proposal.

1 Introduction and Related Work

Stereo algorithms aim at inferring disparity maps by processing images of the same scene from two or more cameras.. This topic was exhaustively surveyed in [10,2] and according to [10] most algorithms consist of four steps: *matching cost computation*, *cost aggregation*, *disparity computation* and *disparity refinement*.

Although cost aggregation is mandatory for local approaches in order to increase the signal to noise ratio this methodology is also frequently adopted by global (or semi-global) approaches [20,18,17,6]. An ideal (fronto-parallel) cost aggregation strategy, to deal with depth discontinuities and ambiguous regions (low textured areas, repetitive patterns, etc), should modify its support at each position according to image content to include only those points with the same (unknown) disparity. Although this behavior is far from ideal, state-of-the-art cost aggregation strategies, recently surveyed and evaluated [14,4], deploying the simple local Winner Takes All (WTA) disparity selection approach allow for obtaining disparity maps comparable to those yielded by algorithms based on schemes that use more complex reasoning. Unfortunately, the execution time is often comparable. Moreover, the results of state-of-the-art cost aggregation strategies are severely affected by image noise. In this paper we propose a cost aggregation strategy that combines the effectiveness of recent local approaches based on adaptive weights with the efficiency and the robustness of conventional local approaches. Although we assume binocular rectified stereo pairs, the proposed method can be extended in a straightforward manner to three or more rectified cameras.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 371–380, 2010.

Bilateral filtering is a non-iterative feature-preserving image smoothing technique [11] that due to its relevance in computer vision, computer graphics and image processing has recently gained a lot of attention [8,7,19,1]. The idea behind bilateral filtering is to jointly and independently enforce a geometric (*spatial filter*) and a color proximity constraint (*range filter*). Given an image I, the value assigned to each point p of the filtered image $\hat{I}$ is a weighted convolution with points q_i in its neighbor $S(p) \subset I$ (a square *support* region centered in p) according to weighting functions related to the spatial distance D_s between p and q_i and a distance D_c in the color space between $I(p)$ and $I(q_i)$.

$$\hat{I}(p) = \frac{\sum_{q_i \in S(p)} W_S(p, q_i) \cdot W_C(I(p), I(q_i)) \cdot I(q_i)}{\sum_{q_i \in S(p)} W_S(p, q_i) \cdot W_C(I(p), I(q_i))} \tag{1}$$

The denominator acts as normalization factor and the two weighting functions W_s and W_c, respectively, assign higher values to points *closer* to the central point p and to points with color intensity *similar* to $I(p)$. Typically, the weights W_s and W_c are assigned according to Gaussian functions, respectively, with variance γ_s and γ_c. The distance between the coordinate points and between triplets in the color space are often computed according to the ℓ_2 norm. Although bilateral filtering has proven to be a very effective technique it is computationally demanding. For this reason, recently, several approximated techniques aimed at reducing its running time have been proposed [8,7]. According to a recent study [14] specifically focused on evaluating and benchmarking state-of-the-art cost aggregation strategies for stereo correspondence, algorithms belonging to the *adaptive weights* [19,12] category dramatically outperformed other approaches in terms of accuracy. In the Adaptive Weight (AW) approach [19] the weight assigned to each point within the support is obtained by applying two independent bilateral filters in the neighborhood of each potential correspondence. Given a point p_r in the reference image I_r and a potential correspondence point p_t in the target image I_t: the weights assigned to each point of the support S(p_r,p_t) are computed by combining (multiplying) the weights that would be yielded by the two independent bilateral filters (with the same parameters γ_s and γ_c) applied to p_r and p_t. The cost of the correspondence $C(p_r, p_t)$ between p_r and p_t is the weighted sum of the TAD (Truncated Absolute Differences) scores within the support normalized by the weights (1).

$$C(p_r, p_t) = \frac{\sum_{\substack{q_{r_i} \in S(p_r) \\ q_{r_i} \in S(p_t)}} W_S(p_r, q_{r_i}) \cdot W_C(I_r(p_r), I_r(q_{r_i})) \cdot W_S(p_t, q_{t_i}) \cdot W_C(I_t(p_t), I_t(q_{t_i})) \cdot TAD(q_{r_i}, q_{t_i})}{\sum_{\substack{q_{r_i} \in S(p_r) \\ q_{r_i} \in S(p_t)}} W_S(p_r, q_{r_i}) \cdot W_C(I_r(p_r), I_r(q_{r_i})) \cdot W_S(p_t, q_{t_i}) \cdot W_C(I_t(p_t), I_t(q_{t_i}))} \tag{2}$$

The weighting function is Gaussian and authors use the ℓ_2 norm in the CIELAB color space for the range filters. This cost aggregation strategy provides excellent results within a WTA framework and has also been successfully adopted within global optimization frameworks [18].

However, it has been shown [12] that under certain circumstances (i.e. along depth discontinuities, low-textured regions, high-textured regions and repetitive patterns) the spatial filter embodied in [19] can lead to wrong correspondences and for these reasons a further segmentation-based constraint was introduced. Therefore, the Segment Support (SS) [12] approach when it computes the weights associated to the two distinct bilateral filters assigns weight 1 to those points belonging to the same segment of the central point and assigns the spatial weight of AW to those points outside the segment containing the central point. This method improves AW, but at the expense of almost doubling the execution time.

It is worth observing that supports for AW and SS are computed by means of a symmetric strategy that relies on both images. Unfortunately AW and SS are computationally very demanding and their execution time [14] are comparable or even worse than those required by global approaches (e.g. on Teddy, AW requires more than 18 minutes while SS requires more than 33 minutes). To reduce the computational complexity of the AW approach a simplified asymmetrical weight assignment strategy was proposed [4]. The weights were computed asymmetrically, according to the reference image only, and approximated by means of a two pass approach (the first pass along the horizontal scanline and the second pass along the perpendicular direction). These simplifications allowed for real-time GPU implementation that yields worse but reasonably accurate [4] disparity maps compared to AW. Another interesting approach based on asymmetrical weight assignment, referred to as SB, was also proposed [3]. This method appears to be a good trade-off between accuracy and computational efficiency deploying segmentation and assigning weights according to the reference image only. Finally, according to the evaluation provided in [14], among effective cost aggregation strategies Variable Windows (VW) [15] deserves particular attention. In fact, although VW is significantly less accurate than AW and SS its execution time is significantly reduced (it takes 26 seconds on Teddy). The efficient aggregation strategy deployed by VW is completely different from the adaptive weight approaches described so far since in VW the weights are always set to 1 while the size of the square support is selected according to three criteria. The best support is selected evaluating for each predefined square region the cost function and its variance. Moreover, to deal with low-textured regions, a biasing term is used to favor large windows. Massive deployment of the *integral image* technique [16] allows notable computational efficiency.

2 Proposed Cost Aggregation Strategy

As already pointed out by the authors [19], the most ambiguous correspondences are set by AW when the support becomes too small. Experimental results show that this behavior mainly occurs in two circumstances: a) when the supports are within regions that are highly textured b) when the support contains uniform regions with pixel intensity similar to the central pixel (not necessarily completely uniform regions). Although case a) seems intrinsically related to the method since it is likely that in highly textured regions several pixels will have different intensity when compared to the central pixel, in both cases pixels with intensity similar to the central pixel should provide the *cue* for setting unambiguous correspondences. In the two cases depicted in Figure 1, an ideal

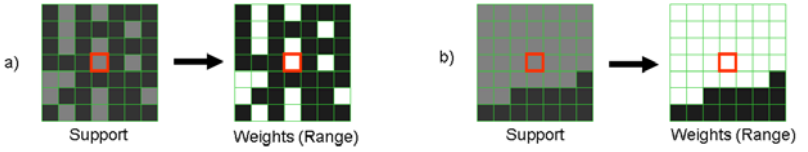


Fig. 1. Case a) and b): weights assigned by an ideal range filter

range filter[1] would assign higher weights (in white) to points with intensity similar to the central one and lower weights to the pixels with different intensity.

When higher weights are assigned to the pixels with intensity similar to the central point (combined with the weights assigned based on the same rationale by the other range filter applied in the other image), the spatial filters and the matching cost would then allow for potentially setting unambiguous correspondences. Hence, points with intensity similar to the central one play a major role in the AW approach. Analyzing the behavior of AW we observed that the origin of the discrepancy with the ideal case could be ascribed to noise and how weights are computed by means of the two independent range filters. In fact, when computing the weight for two pixels with similar color intensity (i.e. when the ℓ_2 norm $\| \Delta \| \rightarrow 0$) the exponential function embodied in the range filter becomes very sensitive to image noise since derivative is very high. Under these circumstances (i.e. when the ℓ_2 norm $\| \Delta \| \rightarrow 0$) image noise severely affects the weights assigned by means of the range filters.

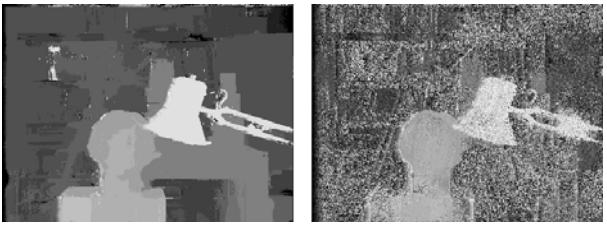


Fig. 2. Disparity maps yielded by AW: (Left) original Tsukuba stereo pair (Right) Tsukuba stereo pair corrupted by Gaussian noise ($\mu = 0$, $\sigma = 1.275$ pixels) independently applied to the three color channels

In Figure 2 we report the disparity maps yielded by AW on the original Tsukuba stereo pair and on the same stereo pair corrupted by Gaussian noise. The figure seems to confirm our analysis; the effects of noise are particularly prominent in the regions with similar color intensity. To deal with this problem the proposed cost aggregation strategy, referred to as Fast Bilateral Stereo (FBS), embodies a simple but effective noise regularization stage for the range filter. Moreover, as the major aim of this research activity was the development of an effective cost aggregation strategy that could

[1] For simplicity we consider greyscale images and a single range filter applied to the support of a single image.

fit within an efficient computational framework we combined the efficiency of the traditional and local approaches for stereo correspondences [14] with a symmetric adaptive weights strategy based on two independent spatial and range filters applied on a regular block basis. Given two points $p_r \in I_r$ and $p_t \in I_t$, for which a correspondence had to be evaluated, and the associated supports $S(p_r) \subset I_r$ and $S(p_t) \subset I_t$ both of size $W \times W$ we partitioned the two supports $S(p_r)$ and $S(p_t)$ in $\frac{W}{w} \times \frac{W}{w}$ regular blocks of size $w \times w$ as shown in Figure 3[2]. At each block of the two supports S_r, S_t we independently assigned two weights according to a spatial filter and a range filter.

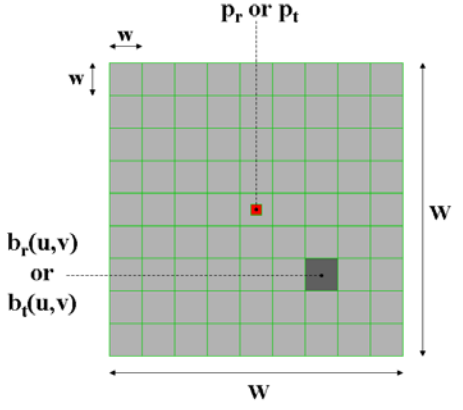


Fig. 3. Proposed partitioning scheme of the two supports $S_r \subset I_r$ and $S_t \subset I_t$

For what concerns the spatial filter, at each point within $w \times w$ block we assigned the spatial weight of the central point of the block according to the W_s function in (1). For what concerns the range filter we assigned at each $w \times w$ block a single weight according to a simple but effective strategy aimed at increasing noise robustness. Assuming that pixels within each block are spatially correlated, the average values of the intensities within the block provide a means to decrease the variance by a factor w. Clearly, when this hypothesis is not verified the averaged value induces a bias that might lead to a non perfect localization of discontinuities. However, with small w the hypothesis that pixel within the $w \times w$ block are correlated often holds and averaging is a simple, yet effective, strategy to reduce image noise. To partially deal with this issue we assigned at each block $b_r(u, v)$ (with $u \in \frac{W}{w}, v \in \frac{W}{w}$) within the support $S(p_r)$ of the reference image the weight

$$W_C\left(I_r\left(p_r\right), \overline{I}_r\left(b_r(u,v)\right)\right) = \exp\left(-\frac{\left\|I_r\left(p_r\right) - \overline{I}_r(b_r(u,v))\right\|}{\gamma_c}\right) \tag{2}$$

with $\overline{I}(b_r(u, v))$ representing the average value of pixels within block $b_r(u, v)$. Using small w allows for reducing the variance within each block as well as the maintenance of an accurate localization with respect to the central point p_r. Experimentally we found

[2] Although not strictly necessary, for simplicity, W mod w is assumed to be 0.

that $w = 3$ provides optimal results on the Middlebury dataset. Obviously, the same strategy is applied to point p_t of the target image I_t.

Once we obtained the block-based weights by means of the two spatial filters and range filters, we combined the weights to obtain a symmetric block-based weighted support and computed the matching cost using a TAD cost function on a pixel basis. That is, at each block a single weight is assigned, but each point within the block is evaluated according to the pixel-wise matching cost. Similarly to (1), the matching cost is normalized by the weights. It is worth observing that cost computation and block averaging can be efficiently computed by means of incremental calculation schemes [16] typically deployed by conventional local approaches. Moreover, compared to (1), the number of range and spatial filters is intrinsically reduced by factor $w \times w$. To further reduce the computational requirements spatial and range weights are stored in look-up tables. Finally, it is worth observing, that for $w = 1$ our computational framework is equivalent to [19].

3 Experimental Results

This section aims at assessing the performance of the proposed approach within a framework specifically focused on the evaluation of state-of-the-art cost aggregation strategies [14]. The disparity maps yielded by the considered cost aggregation strategies on the four images of the Middlebury dataset [9] (Tsukuba, Venus, Teddy and Cones) were obtained by means of a simple WTA strategy without any post processing filtering and without enforcing the left-right consistency constraint. Nevertheless, although the focus here is on the evaluation of the raw cost aggregation strategy and all the considered approaches do not deal explicitly with occlusions, for completeness, we have also report the ALL parameter so as to allow a direct comparison with other approaches on the Middlebury evaluation site. We have reported in Table 1 the results obtained by FBS (parameters $W = 39$, $w = 3$,$\gamma_s = 14$, $\gamma_c = 23$ and TAD threshold 53), by our implementation of the AW approach (referred to as AW* with optimal parameters $W = 35$, $\gamma_s = 31$, $\gamma_c = 13$ and TAD threshold 40) and by the five top performing state-of-the-art cost aggregation strategies [12,3,19,5,15] according to [14][3].

It is worth noting that these results differ from those published in [14] (concerned with SAD cost function) because, for fairness, we deployed the original cost functions originally proposed by the authors of each paper (see [13] section "*Original*"). Here we stress the fact that we were interested in evaluating the performance of the raw cost aggregation strategies, and the results for AW and SS available on the Middlebury evaluation site include post processing steps that are not specified. For all approaches the execution time is concerned with the Teddy stereo pair and for the proposed approach also includes initialization of look-up tables. As FBS has two parameters for the support (W and w) we used $W = 39$, similar to those deployed for the adaptive weight

[3] For [12,3,19,5,15] the optimal parameters found in [14] were deployed, which are available at www.vision.deis.unibo.it/spe/data/parameters.pdf. The execution time for SB was obtained by deploying a much faster segmentation approach compared to the results reported in [14]. For FBS we found the optimal parameters minimizing the NOCC+DISC error on the whole dataset.

Table 1. Accuracy according to the Middlebury web site [9] and (in boldface) according to [14]. The table reports the accuracy of the proposed FBS approach and the five top performing [14] state-of-the-art approaches [12,3,19,5,15]. The disparity maps tagged with symbol † are available in [13] - section '*Original*'. Table also reports the execution time (Intel Core Duo 2.14 GHz processor) concerned with the Teddy stereo pair.

	Tsukuba			Venus			Teddy			Cones			Time
	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC	sec
FBS$_{39(3)}$	**2.95**	4.75	**8.69**	**1.29**	2.87	**7.62**	**10.71**	19.8	**20.82**	**5.23**	15.3	**11.34**	32
SS† **[12]**	**2.15**	4.04	**7.22**	**1.38**	3.0	**6.27**	**10.5**	19.7	**21.2**	**5.83**	16.4	**11.8**	2358
SB† **[3]**	**2.25**	2.86	**8.87**	**1.37**	2.31	**9.4**	**12.7**	20.1	**24.8**	**11.1**	19.2	**20.1**	2
AW*[19]	**3.33**	5.25	**8.87**	**2.02**	3.61	**9.32**	**10.52**	19.7	**20.84**	**3.72**	14.3	**9.37**	3226
AW† **[19]**	**4.66**	6.68	**8.25**	**4.61**	6.18	**13.3**	**12.7**	21.6	**22.4**	**5.5**	16.0	**11.9**	1221
Rel† **[5]**	**5.08**	6.94	**17.9**	**3.92**	5.5	**13.9**	**18.9**	27.0	**29.9**	**11.3**	20.7	**18.3**	803
VW† **[15]**	**3.12**	4.86	**12.4**	**2.42**	3.87	**13.3**	**17.7**	25.9	**25.5**	**21.2**	29.6	**27.3**	26

Table 2. Accuracy for FBS and adaptive weights approach (AW*) on stereo pairs corrupted by Gaussian noise ($\mu = 0, \sigma = 0.255, 1.275, 2.55, 12.75$ pixels) independently applied to the three color channels of the original stereo pairs.

	Noise	Tsukuba			Venus			Teddy			Cones		
	σ	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC	NOCC	ALL	DISC
FBS$_{39(3)}$	0.255	**7.58**	9.32	**14.83**	**14.13**	15.48	**17.83**	**15.42**	24.00	**26.13**	**6.67**	16.60	**13.48**
AW*	0.255	**21.34**	22.94	**25.65**	**24.61**	25.81	**27.54**	**19.98**	28.16	**30.71**	**7.30**	17.55	**14.68**
FBS$_{39(3)}$	1.275	**34.26**	35.44	**36.81**	**45.15**	45.97	**41.58**	**47.85**	53.09	**54.00**	**31.11**	38.42	**38.34**
AW*	1.275	**52.93**	53.81	**49.89**	**56.08**	56.73	**54.74**	**51.28**	56.23	**60.19**	**35.72**	42.80	**41.38**
FBS$_{39(3)}$	2.55	**53.94**	54.70	**54.52**	**68.13**	68.57	**62.87**	**68.56**	71.62	**69.43**	**56.50**	61.03	**59.57**
AW*	2.55	**66.32**	66.87	**63.07**	**74.76**	75.13	**70.26**	**73.56**	76.19	**77.73**	**64.55**	68.36	**65.21**
FBS$_{39(3)}$	12.75	**71.78**	72.15	**74.82**	**85.69**	85.84	**83.30**	**92.30**	93.00	**91.31**	**91.69**	92.41	**91.20**
AW*	12.75	**78.61**	77.50	**78.81**	**87.95**	88.09	**85.10**	**94.65**	95.14	**94.17**	**94.07**	94.59	**93.01**

approach, and $w = 3$. Therefore, our proposal in Table 1 is referred to as FBS$_{39(3)}$. With respect to the NOCC and DISC errors, we noticed that our proposal has an accuracy comparable to the best performing cost aggregation strategies, SS and AW (both implementations) being close in most cases to the results yielded by the best one (SS). It is also worth pointing out that SS and AW run is minutes on the Teddy stereo pair (e.g. SS is almost 40 minutes, AW is about 20 minutes) while FSB takes only 32 seconds. The table also shows that FBS significantly outperforms the accuracy of the VW and Rel approaches on the whole dataset. For SB, we noticed that, in most cases, it is outperformed by FBS (in particular, on Cones, NOCC and DISC errors are about 50 %). Nevertheless, SB is significantly faster than FSB (it takes only 2 seconds on Teddy).

Figure 4 shows the disparity maps for the considered approaches. From the figure, we noticed that, compared to AW, our approach allows for reducing several errors (region in front of the camera in Tsukuba, slanted surfaces in Venus, roof in Teddy). Depth maps yielded by FBS are, in general, less noisy compared to AW. Although not reported here

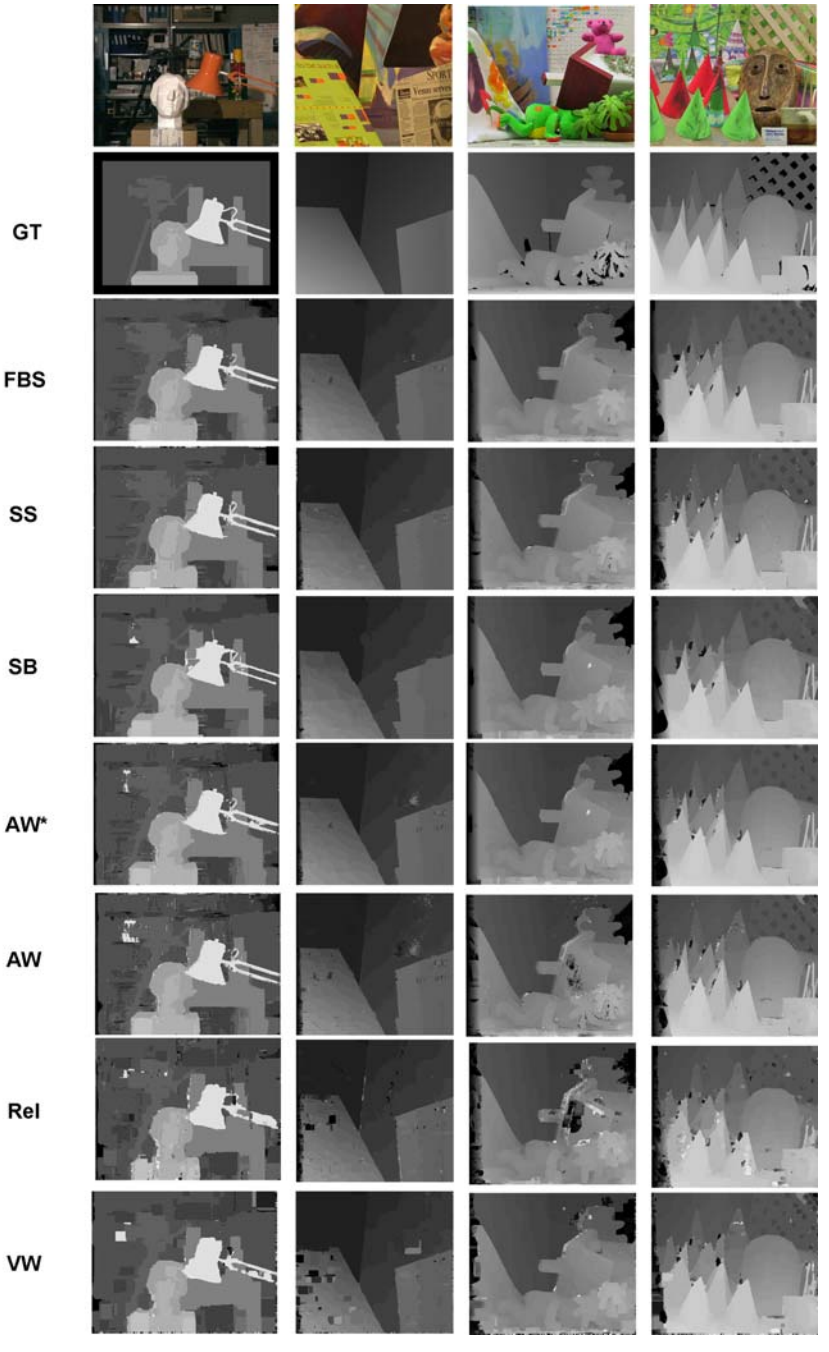


Fig. 4. From top to bottom: reference image, groundtruth (GT), disparity map yielded by the proposed FBS approach, SS [12], SB [3], our implementation of AW [19], original implementation of AW [19], Rel [5] and VW[15].

for the lack of space[4], increasing w decreases accuracy but improves efficiency: in fact ($W = 39$) on Teddy FBS takes 14 sec (4 sec on Tsukuba) with $w = 5$, 9 sec (2 sec on Tsukuba) with $w = 7$ and 5 sec (1 sec on Tsukuba) with $w = 9$. This highlights an interesting behavior of the proposed approach: by modifying w one can trade accuracy for speed and vice versa. This might be interesting in certain applications (e.g. *robot picking*, face detection and recognition) where accurate disparity maps are required only when objects are close to the camera.

To prove the effectiveness of the noise reduction technique embodied in our proposal, we report in Table 2 the results obtained by our proposal and by our implementation of the adaptive weights technique (AW*) on stereo pairs corrupted by Gaussian noise (mean value $\mu = 0$ and variance (in pixels) $\sigma = 0.255, 1.275, 2.55, 12.75$) independently applied to the three color channels of the original stereo pairs. Although both approaches have poor results with higher noise levels, Table 2 reports that for the whole dataset the noise regularization step embodied in the range filter calculation of our proposal is always notably more effective than adaptive weights. For both approaches we deployed the optimal parameters described for Table 1.

4 Conclusions

We have proposed a cost aggregation strategy that combines the efficiency of traditional local algorithms with the accuracy of state-of-the-art approaches. The weight computation strategy proposed deploys a simple, but effective, noise regularization step that allows for improving the accuracy of the original AW approaches and, exploiting efficient incremental calculation schemes, for obtaining a disparity map at a small fraction of the time required by state-of-the-art approaches. Experimental results within a framework specifically aimed at evaluating the performance of state-of-the-art cost aggregation strategies for stereo correspondence confirm the effectiveness of our proposal. Future work is aimed at exploiting the block-based framework proposed to deal with photometric distortions that typically arise in real applications. We are also interested in deploying the cost aggregation strategy proposed within global or semi-global optimization frameworks.

References

1. Ansar, A., Castano, A., Matthies, L.: Enhanced real-time stereo using bilateral filtering. In: 3DPVT 2004, pp. 455–462 (2004)
2. Brown, M.Z., Burschka, D., Hager, G.D.: Advances in computational stereo. IEEE Trans. Pattern Anal. Mach. Intell. 25(8), 993–1008 (2003)
3. Gerrits, M., Bekaert, P.: Local stereo matching with segmentation-based outlier rejection. In: Proc. CRV 2006, p. 66 (2006)
4. Gong, M., Yang, R.G., Liang, W., Gong, M.W.: A performance study on different cost aggregation approaches used in real-time stereo matching. Int. Journal Computer Vision 75(2), 283–296 (2007)

[4] Additional experimental results are available at:
www.vision.deis.unibo.it/smatt/fast_bilateral_stereo.htm

5. Kang, S.B., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. In: Proc. CVPR 2001, pp. 103–110 (2001)
6. Mattoccia, S., Tombari, F., Di Stefano, L.: Stereo vision enabling precise border localization within a scanline optimization framework. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 517–527. Springer, Heidelberg (2007)
7. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. Int. Journal Computer Vision 81(1), 24–52 (2009)
8. Porikli, F.M.: Constant time o(1) bilateral filtering. In: CVPR 2008, pp. 1–8 (2008)
9. Scharstein, D., Szeliski, R.: http://vision.middlebury.edu/stereo/
10. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. Journal Computer Vision 47(1/2/3), 7–42 (2002)
11. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV 1998, pp. 839–846 (1998)
12. Tombari, F., Mattoccia, S., Di Stefano, L.: Segmentation-based adaptive support for accurate stereo correspondence. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 427–438. Springer, Heidelberg (2007)
13. Tombari, F., Mattoccia, S., Di Stefano, L., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence, www.vision.deis.unibo.it/spe/SPEHome.aspx
14. Tombari, F., Mattoccia, S., Di Stefano, L., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. In: CVPR 2008, pp. 1–8 (2008)
15. Veksler, O.: Fast variable window for stereo correspondence using integral images. In: Proc. CVPR 2003, pp. 556–561 (2003)
16. Viola, P., Jones, M.J.: Robust real-time face detection. Int. Journal of Computer Vision 57(2), 137–154 (2004)
17. Wang, L., Liao, M., Gong, M., Yang, R., Nister, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: Proc. 3DPVT 2006, pp. 798–805 (2006)
18. Yang, Q., Wang, L., Yang, R., Stewénius, H., Nistér, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(3), 492–504 (2009)
19. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. PAMI 28(4), 650–656 (2006)
20. Yoon, K.J., Kweon, I.S.: Stereo matching with symmetric cost functions. In: Proc. CVPR 2006, vol. 2, pp. 2371–2377 (2006)

Detecting Critical Configurations for Dividing Long Image Sequences for Factorization-Based 3-D Scene Reconstruction

Ping Li[1], Rene Klein Gunnewiek[2], and Peter de With[3]

[1] Eindhoven University of Technology
p.li@tue.nl
[2] Philips Research Europe
rene.klein.gunnewiek@philips.com
[3] CycloMedia Technology B.V.
P.H.N.de.With@tue.nl

Abstract. The factorization 3-D reconstruction method requires that all feature points must occur in all images in a sequence. A long sequence has to be divided into multiple subsequences for partial reconstructions. This paper proposes an algorithm for dividing a long sequence for factorization-based Structure and Motion (SaM). First, we propose an Algorithm for Detecting a few Critical Configurations (ADCC) where Euclidean reconstruction degenerates. The critical configurations include: (1) coplanar 3-D points, (2) pure rotation, (3) rotation around two camera centers, and (4) presence of excessive noise and outliers in the measurements. The configurations in cases of (1), (2) and (4) will affect the rank of the scaled measurement matrix (SMM). The number of camera centers in case of (3) will affect the number of independent rows of the SMM. By examining the rank and the row space of the SMM, we detect the above-mentioned critical configurations. With the proposed ADCC algorithm, we are able to divide a long sequence into subsequences such that a successful 3-D reconstruction can be obtained on each subsequence with a high confidence. Experimental results on both synthetic and real sequences demonstrate the effectiveness of the proposed algorithm for an automatic 3-D reconstruction using the factorization method.

1 Introduction

The factorization-based 3-D reconstruction method [1] requires that all feature points must occur in all frames of an image sequence. A long sequence has to be divided into multiple short subsequences in order to track a sufficient of number of feature points in the subsequence for 3-D reconstruction. However, though short sequences usually have sufficient feature points, the camera disparity may be insufficient due to the limited number of images (e.g. the camera may undergo pure rotation), which leads to a failure of a factorization-based 3-D reconstruction. Long sequences usually have sufficient disparity. However, the number of feature points that can be tracked is usually limited, which also leads to a failure. Thus, a tradeoff has to be made between the number of feature points and the number of images. Besides, Euclidean scene information can never be recovered by any algorithm under so-called *critical motions* and *critical surfaces* [1,2]. For example, if all detected feature points are coplanar, or all cameras have

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 381–394, 2010.

the same center, projective reconstruction using the factorization method will not be possible. Such critical configurations where Euclidean reconstruction degenerates need also to be detected while we divide a long sequence. Furthermore, 3-D reconstruction is sensitive to noise and outliers, especially when the configuration is 'close' to a critical configuration [3]. The presence of one single outlier may deteriorate the whole SaM process. It is useful that the impact of the noise and outliers on the 3-D reconstruction process can be measured. This paper proposes an algorithm for dividing long image sequences while considering critical configurations, noise and outliers, and the tradeoff on the number of feature points.

Critical motions and surfaces have been rigorously investigated in literature. Assuming that the focal lengths are the only unknown parameters, a complete categorization of critical motions is given in [2]. Ref. [3] extended this work by relaxing the constraints on the intrinsic parameters. The critical motions under different calibration constraints (zero skew, unit aspect ratio, vanishing principal point) are derived. Some particular critical configurations that frequently occur in practice are discussed. Though the problem of critical motions and surfaces have been extensively studied in literature, we found little work on the detection of them. Related work on detection of the degenerate configurations for estimating the fundamental matrix is reported in [4]. For breaking the long sequence for factorization-based projective reconstruction, we found related work [5], where a quantitative measure is proposed for dividing a long sequence based on measuring the number of the feature points, the homography error, and the distribution of the feature points. This paper contributes in two ways. First, an algorithm is proposed to detect the critical configurations resulting from (1) pure rotation, (2) coplanar 3-D points, and (3) rotation around two camera centers. Second, an algorithm is proposed to divide a long image sequence, which balances the number of the feature points and the length of the subsequences for a successful factorization-based 3-D reconstruction.

2 Factorization-Based 3-D Reconstruction

2.1 Notations

Assume a set of n 3-D points projected onto m perspective cameras. Let $\mathbf{X}_j = (X_j, Y_j, Z_j, 1)^T$ with $j = (1, \cdots, n)$ be the homogenous coordinates of the 3-D points. Let $\mathbf{P}_i = \mathbf{K}_i\mathbf{R}_i[\mathbf{I} - \mathbf{C}_i]$ with $i = (1, \cdots, m)$ be the camera projection matrices, where $\mathbf{K}_i$ is the intrinsic camera matrix, $\mathbf{R}_i$ is the camera orientation matrix and $\mathbf{C}_i$ is the camera center. The 2D projection of 3-D point $\mathbf{X}_j$ in image i can be computed by $\lambda_{ij}\mathbf{x}_{ij} = \lambda_{ij}(u_{ij}, v_{ij}, 1)^T = (\alpha_i\mathbf{P}_i)(\beta_j\mathbf{X}_j)$, where $\mathbf{x}_{ij} = (u_{ij}, v_{ij}, 1)^T$ is the homogeneous coordinate of the 2D projection; α_i and β_j are two arbitrary non-zero scaling factors; λ_{ij} is the projective depth, which becomes the true optical depth if $\alpha_i = \beta_j = 1$. Rewriting above equation in matrix form, we obtain

$$\mathbf{W}_s = \begin{pmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{pmatrix} = \begin{pmatrix} \alpha_1\mathbf{P}_1 \\ \vdots \\ \alpha_m\mathbf{P}_m \end{pmatrix} \begin{pmatrix} \beta_1\mathbf{X}_1 \cdots \beta_n\mathbf{X}_n \end{pmatrix} = \mathbf{P}\mathbf{X}, \tag{1}$$

where $\mathbf{W}_s$ is the $3m \times n$ SMM, $\mathbf{P}$ is the $3m \times 4$ Euclidean motion matrix and $\mathbf{X}$ is the $4 \times n$ Euclidean shape matrix.

2.2 Matrix Rank-r Factorization

Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition of a matrix $\mathbf{W}$. The best approximating matrix $\mathbf{W}_r$ to $\mathbf{W}$ under the Frobenius norm with a rank $\leq r$ is computed by $\mathbf{W}_r = \mathbf{U}\mathbf{\Sigma}_r\mathbf{V}^T$, where $\mathbf{\Sigma}_r$ is obtained from the diagonal matrix $\mathbf{\Sigma}$ by keeping the first r largest singular values [2]. Thus, $\mathbf{W}$ can be approximated using the product of two rank-r matrices as $\mathbf{W} \approx \mathbf{W_r} = \mathbf{P_r}\mathbf{X_r}$, where $\mathbf{P}_r$ and $\mathbf{X}_r$ can be computed as $\mathbf{P}_r = \mathbf{U}\mathbf{\Sigma}_r^{'}$ and $\mathbf{X}_r = \mathbf{\Sigma}_r^{''}\mathbf{V}^T$ ($\mathbf{\Sigma}_r^{'}$ and $\mathbf{\Sigma}_r^{''}$ are any two diagonal matrices of a rank of r that satisfy $\mathbf{\Sigma}_r = \mathbf{\Sigma}_r^{'}\mathbf{\Sigma}_r^{''}$). The above process of approximating $\mathbf{W}$ by two rank-r matrices is referred to as **rank-r factorization** in this paper. Obviously, if the rank of $\mathbf{W}$ is larger than r, some nonzero singular values are discarded during factorization, which results in an inaccuracy of the approximation. Thus, the accuracy of the rank-r factorization can be measured using the 'rank-r-ness' κ_r of $\mathbf{W}$, which is defined as

$$\kappa_r = (1 - \sigma_{r+1}/\sigma_r) \times 100\%, \tag{2}$$

where σ_r is the r-th largest singular value. A large κ_r implies that the rank of $\mathbf{W}$ is close to r, since σ_{r+1} is negligible in this case.

In our proposal, κ_r together with the re-projection error, which will be defined below, are used to determine the rank of the SMM. Note that κ_r alone does not give much confidence on the rank of the SMM. For example, when the rank of $\mathbf{W}$ is smaller than r, κ_r may have a large value even though both σ_{r+1} and σ_r are very small.

2.3 Rank-r Iteration: Projective Reconstruction Using Separate Bundle Adjustment

The task of factorization-based projective reconstruction is to recover the projective depths λ_{ij}, so that $\mathbf{W}_s$ can be factorized into a $3m \times 4$ projective motion matrix $\hat{\mathbf{P}}$ and a $4 \times n$ projective shape matrix $\hat{\mathbf{X}}$, which can be upgraded afterwards to Euclidean matrices[1]. The accuracy of the projective reconstruction can be measured by the re-projection/residual error that is computed as

$$E = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left\{\left(u_{ij} - \frac{\hat{\mathbf{P}}_{i(1)}^T\hat{\mathbf{X}}_j}{\hat{\mathbf{P}}_{i(3)}^T\hat{\mathbf{X}}_j}\right)^2 + \left(v_{ij} - \frac{\hat{\mathbf{P}}_{i(2)}^T\hat{\mathbf{X}}_j}{\hat{\mathbf{P}}_{i(3)}^T\hat{\mathbf{X}}_j}\right)^2\right\}}{2 \times m \times n}}. \tag{3}$$

In the above, $\hat{\mathbf{P}}_{i(1)}^T$, $\hat{\mathbf{P}}_{i(2)}^T$ and $\hat{\mathbf{P}}_{i(3)}^T$ are the row vectors of $\hat{\mathbf{P}}_i$. The algorithms using bundle adjustment for projective reconstruction solve $\hat{\mathbf{P}}$ and $\hat{\mathbf{X}}$ by minimizing the above re-projection error. When eliminating the scale factors in Eq. (3), we obtain

$$u_{ij} - \frac{\hat{\mathbf{P}}_{i(1)}^T\hat{\mathbf{X}}_j}{\hat{\mathbf{P}}_{i(3)}^T\hat{\mathbf{X}}_j} = 0 \quad \text{and} \quad v_{ij} - \frac{\hat{\mathbf{P}}_{i(2)}^T\hat{\mathbf{X}}_j}{\hat{\mathbf{P}}_{i(3)}^T\hat{\mathbf{X}}_j} = 0. \tag{4}$$

[1] $\hat{\mathbf{P}}$ and $\mathbf{P}$ are related by an unknown 4×4 projective transformation $\mathbf{H}$ as $\mathbf{P} = \hat{\mathbf{P}}\mathbf{H}$, and $\mathbf{X} = \mathbf{H}^{-1}\hat{\mathbf{X}}$.

Given the above two equations, we can formulate linear equations to solve for $\hat{\mathbf{X}}_j$ and $\hat{\mathbf{P}}_i$ by alternatively holding $\hat{\mathbf{P}}$ and $\hat{\mathbf{X}}$ constant, as proposed in the Weighted Iterative Eigen (WIE) algorithm [6], which is introduced below.

(a) Assuming $\lambda_{ij} = 1$, factorize $\mathbf{W}_s$ into two rank-4 matrices $\hat{\mathbf{P}}^{(0)}$ and $\hat{\mathbf{X}}^{(0)}$ using rank-4 factorization.
(b) Given $\hat{\mathbf{P}}^{(k)}$ and $\hat{\mathbf{X}}^{(k)}$ (k denotes the iteration count), update $\hat{\mathbf{X}}_j^{(k+1)} (\forall j = 1, \ldots, n)$ by minimizing the residual error of the $2m$ number of linear equations derived from Eq. (4).
(c) Given $\hat{\mathbf{P}}^{(k)}$ and $\hat{\mathbf{X}}^{(k+1)}$, update $\hat{\mathbf{P}}_i^{(k+1)} (\forall i = 1, \ldots, m)$ by minimizing the residual error of the $2n$ number of linear equations derived from Eq. (4).
(d) Iterate (b) and (c) until the projective depths $\lambda_{ij}^{(k+1)}$ converge[2] or a maximum number of iterations have been executed.

Derivation of the linear equations from Eq. (3) in Steps (b) and (c) can be found in [6].

In the above algorithm, $\mathbf{W}_s$ is factorized into two rank-4 matrices. However, as discussed in Section 2.2, we can factorize $\mathbf{W}_s$ into two matrices of an arbitrary rank r. In that case, we will obtain a $3m \times r$ matrix $\hat{\mathbf{P}}^{(0)}$ and a $r \times n$ matrix $\hat{\mathbf{X}}^{(0)}$ in Step (a), which is refined later in Steps (b), (c) and (d). We refer to such process as **rank-r iteration**. It is observed that the rank-r iteration converges quickly if $\mathbf{W}_s$ is indeed of rank r, while the iteration may not converge if the rank of $\mathbf{W}_s$ differs from r. This property is exploited in our proposed algorithm to determine the rank of the SMM, as will be discussed in Section 3.

2.4 Factorization-Based Self-calibration

Given the projective reconstruction $(\hat{\mathbf{P}}_i, \hat{\mathbf{X}}_j)$, one has to compute the 4×4 transformation $\mathbf{H}$ that relates the projective motion and shape with Euclidean motion and shape by

$$\alpha_i \mathbf{P}_i = \hat{\mathbf{P}}_i \mathbf{H} \quad \text{and} \quad \beta_j \mathbf{X}_j = \mathbf{H}^{-1} \hat{\mathbf{X}}_j. \tag{5}$$

In this section, we briefly introduce the factorization-based self-calibration technique proposed in [1], which is used in this paper. We represent $\alpha_i \mathbf{P}_i$ by

$$\alpha_i \mathbf{P}_i = [\mathbf{M}_i \ \mathbf{T}_i], \text{ where} \tag{6}$$

$$\mathbf{M}_i = \alpha_i \mathbf{K}_i \mathbf{R}_i = [\mathbf{m}_{xi} \ \mathbf{m}_{yi} \ \mathbf{m}_{zi}]^T \text{ and } \mathbf{T}_i = -\alpha_i \mathbf{K}_i \mathbf{R}_i \mathbf{C}_i = [T_{xi} \ T_{yi} \ T_{zi}]^T. \tag{7}$$

Knowing $\hat{\mathbf{P}}$, $\hat{\mathbf{X}}$ and $\mathbf{W}_s$, the task of self-calibration is to determine $\mathbf{H}$ by imposing constraints on $\mathbf{R}_i$ and $\mathbf{K}_i$. With the assumption that the principal point is at the origin, the aspect ratio equals unity and the skew equals zero, it can be easily verified that

$$|\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \text{ and } \mathbf{m}_{xi}\mathbf{m}_{yi}^T = \mathbf{m}_{xi}\mathbf{m}_{zi}^T = \mathbf{m}_{yi}\mathbf{m}_{zi}^T = 0. \tag{8}$$

[2] The projective depths $\lambda_{ij}^{(k+1)} = \hat{\mathbf{P}}_{i(3)}^{T(k+1)} \hat{\mathbf{X}}_j^{(k+1)}$ are considered converged if the relative change of the projective depths between two consecutive iterations is smaller than a threshold.

Thus, we obtain 4 linear constraints on $\mathbf{M}_i\mathbf{M}_i^T$ from each camera, which are referred to as *calibration constraints*. We will show below how the calibration constraints are used to solve for $\mathbf{H}$. Let $\mathbf{H} = [\mathbf{A}\ \mathbf{B}]$, where $\mathbf{A}$ is a 4×3 matrix and $\mathbf{B}$ is a 4-vector, from Eqs. (5) and (6), we obtain $\mathbf{M}_i = \hat{\mathbf{P}}_i\mathbf{A}$ and $\mathbf{T}_i = \hat{\mathbf{P}}_i\mathbf{B}$. Thus, we have

$$\mathbf{M}_i\mathbf{M}_i^T = \hat{\mathbf{P}}_i\mathbf{A}\mathbf{A}^T\hat{\mathbf{P}}_i^T = \hat{\mathbf{P}}_i\mathbf{Q}\hat{\mathbf{P}}_i^T. \tag{9}$$

Through Eq. (9), the 4 linear constraints (Eq. (8)) on $\mathbf{M}_i\mathbf{M}_i^T$ are transferred to 4 linear constraints on the 10 elements of the 4×4 symmetric matrix $\mathbf{Q}$. For m cameras, linear least squares solution of $\mathbf{Q}$ can be computed from $4m + 1$ linear equations[3]. $\mathbf{A}$ is then computed by rank-3 decomposition of $\mathbf{Q}$. Please refer to [1] for details on how $\mathbf{B}$ is solved. Obviously, at least 9 independent linear equations are required for solving the 10 elements of $\mathbf{Q}$ (with one element fixed to 1). This implies that we require at least 3 distinct cameras for factorization-based self-calibration under the above-mentioned calibration assumptions, as will be proven below. The following proof is found by the author and forms an alternative to existing work [2,3,7].

Proof: From Eqs. (5)(6) and (7), we have $\hat{\mathbf{P}}_i = \mathbf{M}_i[\mathbf{I}\ -\mathbf{C}_i]\mathbf{H}^{-1}$. Suppose cameras $\mathbf{P}_p$ and $\mathbf{P}_q$ have the same center, i.e., $\mathbf{C}_p = \mathbf{C}_q$, it can be verified that

$$\hat{\mathbf{P}}_p = \mathbf{M}_p\mathbf{M}_q^{-1}\hat{\mathbf{P}}_q. \tag{10}$$

Substitute Eq. (10) into $\mathbf{M}_p\mathbf{M}_p^T = \hat{\mathbf{P}}_p\mathbf{Q}\hat{\mathbf{P}}_p^T$, we obtain $\mathbf{M}_q\mathbf{M}_q^T = \hat{\mathbf{P}}_q\mathbf{Q}\hat{\mathbf{P}}_q^T$. Thus, $\mathbf{P}_p$ and $\mathbf{P}_q$ actually provide the same set of calibration constraints. □

In practice, due to the inaccuracy of the calibration constraints (e.g. the principal point is not exactly located at the origin), errors in the measurements and the degeneracy of the configurations, more cameras are usually required for self calibration. One contribution of this paper is that an algorithm is proposed to count the number of distinct camera centers to ensure sufficient calibration constraints, as will be discussed in Section 3.

3 The Proposed Algorithms

In the following, each of the 4 critical configurations presented in the abstract is investigated, and referred to as C1-C4. After that, the algorithms for counting the number of distinct camera centers and detecting the critical configurations are presented.

C1. *Coplanar 3-D points*: From Eq. (1), we see that $\mathbf{W}_s$ is of rank 3, since there are only 3 independent columns in $\mathbf{X}$, and consequently in $\mathbf{W}_s$.

C2. *Pure rotation*: $\mathbf{W}_s$ is of rank 3, since there are only 3 independent rows in $\mathbf{W}_s$, as shown from the following proposition. **Proposition 1:** Let $\mathbf{v}_i = [\lambda_{i1}\mathbf{x}_{i1}, \ldots, \lambda_{in}\mathbf{x}_{in}]$ be the partial SMM for camera i, the 3 row vectors of $\mathbf{v}_i$ are linearly dependent on the 3 row vectors of $\mathbf{v}_j$ iff cameras i and j have the same center. *Proof*: From Eq. (1), we have:

$$\mathbf{v}_i = \alpha_i\mathbf{K}_i\mathbf{R}_i[\beta_1(\mathbf{X}_1 - \mathbf{C}_i), \ldots, \beta_n(\mathbf{X}_n - \mathbf{C}_i)],$$
$$\mathbf{v}_j = \alpha_j\mathbf{K}_j\mathbf{R}_j[\beta_1(\mathbf{X}_1 - \mathbf{C}_j), \ldots, \beta_n(\mathbf{X}_n - \mathbf{C}_j)].$$

[3] One extra equation is obtained by requiring that the scale factor of the first camera α_1 equals unity, i.e., $\mathbf{m}_{z1}\mathbf{m}_{z1}^T = 1$.

If $\mathbf{C}_i = \mathbf{C}_j$, then $\mathbf{v}_j = (\mathbf{K}_j\mathbf{R}_j\mathbf{R}_i^T\mathbf{K}_i^{-1}\alpha_j/\alpha_i)\mathbf{v}_i = \mathbf{M}\mathbf{v}_i$, where $\mathbf{M}$ is a 3×3 non-singular matrix. If $\mathbf{C}_i \neq \mathbf{C}_j$, representing $\mathbf{v}_j$ by $\mathbf{M}\mathbf{v}_i$ is generally not possible. □

C3. *Rotation around two camera centers*: $\mathbf{W}_s$ has rank 4 in this case. We have to ensure at least 3 distinct camera centers for a successful factorization-based self-calibration.

C4. In the presence of excessive noise or outliers, the rank of the SMM will deviate from 3 or 4. Consequently, large re-projection errors and low rank-r-ness are expected for both the rank-3 and the rank-4 iterations. In practice, this case rarely occurs, since most outliers can be rejected using constraints such as the Epipolar constraint.

Counting of the number of distinct camera centers will be discussed in Section 3.1, while the dection of configurations C1, C2, C3 and C4 will be discussed in Section 3.2.

3.1 Algorithm for Counting Distinct Camera Centers (ACCC)

As discussed in Section 2.4, at least three distinct cameras are required for the factorization-based self-calibration. To measure the 'closeness' between two cameras, we define it as the normalized distance between the 'average' camera and object:

$$D = \frac{\|\mathbf{X}_a - (\mathbf{C}_i + \mathbf{C}_j)/2\|}{\|\mathbf{C}_i - \mathbf{C}_j\|}, \tag{11}$$

where $\mathbf{C}_i$ and $\mathbf{C}_j$ are the two camera centers, $\mathbf{X}_a$ is the geometry center of the set of the 3-D points. Apparently, the smaller the D, the closer the object with respect to the cameras, and thus the more distinct the cameras are located. Thus, for a successful factorization-based Euclidean reconstruction, we need to make sure that the closeness metrics D of at least 3 camera pairs are below certain threshold. However, the problem is that D cannot be computed prior to the Euclidean reconstruction. As a solution, Proposition 1 suggests that the closeness between two cameras $\mathbf{C}_i$ and $\mathbf{C}_j$ can be measured by the difference between $\mathbf{v}_i$ and $\mathbf{v}_j$, which leads to the following ADCC algorithm. Step1: randomly pick a $\mathbf{v}_i$ from $\mathbf{W}_s$. Step2: check if the row space of $\mathbf{v}_j(\forall j \neq i)$, denoted $RS(\mathbf{v}_j)$, can be spanned by $RS(\mathbf{v}_i)$. This can be done by checking if the maximum angle deviation between the individual row vectors of $\mathbf{v}_j$ and their corresponding projections in $RS(\mathbf{v}_i)$ is below a threshold. The maximum deviation d is found by

$$d = \text{asin}(\max_{k=1}^{3}(\|\mathbf{v}_{j(k)}^T - \mathbf{J}\mathbf{v}_{j(k)}^T\|/\|\mathbf{v}_{j(k)}^T\|)) \times 180/\pi, \tag{12}$$

where $\mathbf{J} = [\mathbf{v}_i^T(\mathbf{v}_i\mathbf{v}_i^T)^{-1}\mathbf{v}_i]$ is the projection matrix that projects a vector into $RS(\mathbf{v}_i)$, and $\mathbf{v}_{j(k)}^T$ is the kth row vector of $\mathbf{v}_j$. Step3: if d is smaller than a given threshold T_d, cameras i and j are considered to have the same center. Repeat the above steps until all cameras are clustered with distinct camera centers.

3.2 Algorithm for Detecting Critical Configurations (ADCC)

Based on the above discussions, we propose the following ADCC algorithm.

S1. Perform both the *rank-3* and the *rank-4* iterations. If both iterations fail to converge[4], we conclude that the the rank of the SMM is larger than 4 and the measurements contain either outliers or excessive noise. If both iterations converge, or only the rank-3 iteration converges, we conclude that either the 3-D points are coplanar or the camera undergoes pure rotation. If only the rank-4 iteration converges, we proceed to Step S2.
S2. Count the number of camera centers using the ACCC algorithm. If there are more than two camera centers, we proceed to further processing.

The essence of the above algorithm is that we require a high rank-4-ness of the SMM (indicated by κ_3, κ_4, E_3 and E_4), and more than two camera centers for factorization-based 3-D reconstruction. The configurations that do not satisfy these necessary conditions are considered critical and should be omitted for subsequent processing.

3.3 Discussion on Detection of Critical Configurations

We now give an analysis on the possible false detections by the proposed ADCC algorithm. There are mainly two flaws that may lead to a false detection by ADCC. *Reason 1*: As pointed out in [8], there is no theoretical justification that the iterative projective reconstruction methods such as those in [9,10] will converge to sensible results, even when the data is free of noise. This holds for the WIE algorithm, where the iteration may not converge to useful results with or without noise. Consequently, relying on the results of WIE iteration for determining the rank of the SMM is not trustable. *Reason 2*: The presence of the noise or outliers may completely mask the degeneracy. For example, when outliers are present in coplanar 3-D points, the 'coplanar points' may become non-coplanar. Due to the above two reasons, ADCC may result in the following two false detections. (a) The SMM is detected as 'noisy'. However, the actual rank is 3 or 4, when WIE does not converge (Reason 1). (b) The rank of SMM is detected as 4. However, the actual rank is 3, when the rank of the SMM increases from 3 to 4 due to the presence of noise or outliers (Reason 2).

Though lacking a theoretical justification, we argue that the probability that Case (a) occurs is low because of the good convergence capability of the WIE algorithm, as demonstrated by the experimental results in Section 4. As for the probability that the rank of the SMM increases from 3 to 4 due to the presence of noise or outliers, we do not have a statistical measurement. However, no existing method is able to completely avoid such a false detection. After all, the following two grounds give a good justification of the ADCC algorithm. First, similar to curve fitting where a high-order curve cannot be well fit by a low-order curve, it is not possible to well approximate a high-rank SMM using a lower-rank matrix. Thus, as long as the rank-r iteration produces a small residual, we can safely conclude that the rank of the SMM is maximally r. Second, a high rank-r-ness κ_r on top of a small residual error further increases the confidence that the rank of the SMM is r, since σ_{r+1} is negligible compared with σ_r. In the cases when

[4] The convergence of the rank-r iteration is judged by both the residual error and the rank-r-ness of the resulting SMM. For example, for the real data, rank-r iteration can be considered converged when the residual error is below one pixel and the resulting rank-r-ness is above 90%.

both iterations converge to small residuals, the lower-rank model is selected, as in S1 of Section 3.2.

As discussed in [11,12], degeneracy detection can be tackled using the model-checking approach, where the best geometric model is selected using some scoring criteria considering both the *goodness of the fit* of the model to the observed data and the *model complexity*. A good model should not only produce small residual, but also should have a low complexity. In the following, the relation between ADCC and the model-checking approach is discussed.

In our experiments, the G-AIC [11] of Kanatani is computed for performance evaluation, which is computed by

$$\text{G-AIC} = \hat{J} + 2(Nd + p)\varepsilon^2, \text{ with } \varepsilon^2 = \frac{\hat{J}}{cN - p}, \tag{13}$$

where $\hat{J}$ is the residual error, N is the number of data measurements (e.g. the number of feature points), d is the dimension of the model, p is the number of the parameters of the parameterized model, c is the number of the constraints provided by one observed data sample, and ε^2 is the estimated noise level. Eq. (13) implies that a good model should not only produce small residual $\hat{J}$, but also should have a low complexity d and p.

Comparing with ADCC, $\hat{J}$ is equivalent to the residual E, which describes how well the model fits to the data. However, the second term of the right-hand side of Eq. (13), which describes the model complexity, is simply represented by the rank of the SMM in ADCC. Furthermore, instead of giving a score of the goodness of the fit and the model complexity, ADCC employs a hard decision. That is, *a low-rank approximation is preferred as long as the resulting residual is below a statistically meaningful threshold*. There are two reasons that the geometric model selection criteria such as G-AIC cannot be directly used in our problem. (1) The WIE algorithm does not guarantee that the rank-4 iteration will always produce a smaller residual than the rank-3 iteration. (2) Though it was demonstrated that G-AIC works well for curve fitting, the weighting between the residual and the model complexity in G-AIC may not be optimal for our problem. Experimental results on this can be found in Section 4.

3.4 Algorithm for Dividing Long Image Sequence (ADLS)

The actual number of feature points and images required for reconstruction depends on compounding factors including the scene shape, the camera positions, the noise levels, distribution of the feature points and so on, which are referred to as 'criticalness' of the configuration in this paper. The challenges to determine the optimal division of a long sequence include: (1) we lack metrics to quantitatively measure the above-mentioned criticalness; (2) the number of the feature points decreases with the length of the subsequence. We can not obtain a large number of feature points and a large number of images at the same time. From the proposed ADCC algorithm, it is observed that the criticalness of configuration can be measured by parameters κ_3,κ_4,E_3, E_4 and $\#cc$, where $\#cc$ is the number of distinct camera centers. If self-calibration has been done, the residual error ξ of the $4m + 1$ linear equations in Section 2.4 can also be used as a metric. Thus, the number of feature points $\#fp$ can be determined based on the values

of κ_3,κ_4,E_3, E_4, $\#cc$ and ξ. This leads to the proposed ADLS algorithm. The steps of the ADLS algorithm are based on iteratively computing the next subsequence, as described below:

S1. Specify a minimum number of feature points and a minimum number of images, and obtain a few subsequences with varying number of images. Perform ADCC for each subsequence.
S2. If the subsequence is detected as non-critical, perform the self-calibration procedure.
S3. Among all non-critical subsequences that satisfy the necessary conditions specified by the ADCC algorithm, choose the subsequence that gives the smallest residual error ξ.

It is important to note that the proposed ADLS algorithm provides only necessary conditions for Euclidean reconstruction, because the proposed ADCC algorithm is not able to detect other critical configurations than C1-C4. For example, if the camera undergoes a pure translation without rotation, Euclidean reconstruction will degenerate. However, the proposed metrics will not be able to detect it. Furthermore, using algebraic error ξ as a selection criteria lacks the geometric meaning and is not optimal.

4 Experimental Results

We have tested the ADCC algorithm on both the synthetic and the real sequences shown in Fig. 1. In the experiments, we assume the principal point is at the origin, the aspect ratio equals unity and the skew is zero. Only the focal lengths are assigned varying values.

4.1 Detecting Pure Rotation and Coplanar 3-D Points

Fig. 1(a) shows the synthetic house that is used for our experiments, where we observe 15 points in general positions and 10 cameras pointing towards the house. Knowing the coordinates of all 15 points, we project them onto the 10 cameras. Afterwards, Gaussian noise[5] is added to each coordinate of the 2D projections. Euclidean reconstruction is then performed on the noisy 2D projections. The accuracy A of the reconstruction is measured by the relative difference between the recovered focal lengths and the assigned focal lengths, and is computed by $A = 1/m \sum_{i=1}^{m}(|f^i - s \times f_r^i|/f^i) \times 100\%$, where f^i is the preset focal length for the i-th camera, f_r^i is the recovered focal length, s is the scale factor computed as $s = (\sum_{i=1}^{m} f^i)/(\sum_{i=1}^{m} f_r^i)$.

Table 1 shows the experimental results of ADCC algorithm on both the synthetic and real sequences, where the column A denotes the 'accuracy' of the factorization-based 3-D reconstruction and is assigned one of the following five values. 'D' is assigned when

[5] Gaussian noise of 4 different magnitudes is added to each coordinate of the 2D projections. The 4 standard deviations of the Gaussian noise are set to $0.002w$, $0.001w$, $0.0005w$ and $0.0002w$, respectively, where $w = 14$ is the width of the synthetic image. To simulate the influence of the outliers, we assume that 1% of the feature points are outliers, of which the magnitude is computed as $0.05w$.

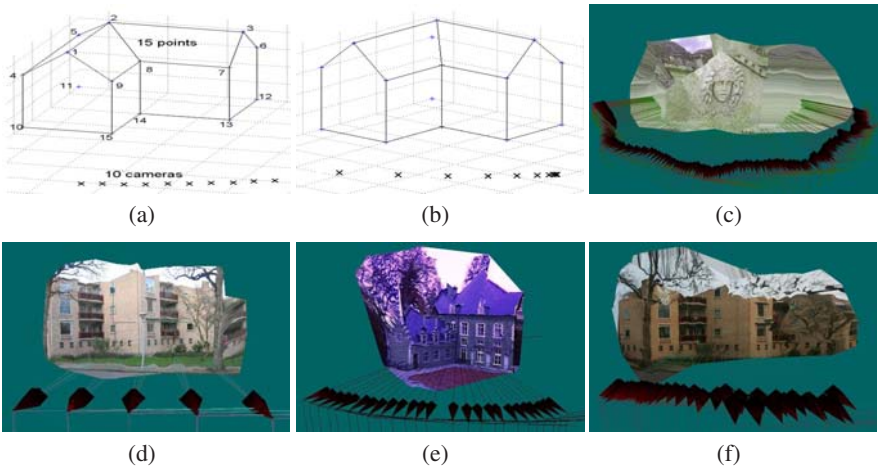


Fig. 1. Test sequences: **(a)** synthetic house with 10 cameras equally displaced; **(b)** synthetic house with 10 cameras displaced with varying distances; **(c)** *medusa* with 195 images from [13]; **(d)** *house1* with 20 images taken from 5 viewpoints (4 images for each viewpoint); **(e)** *castle* with 26 images from [13]; **(f)** *house2* with 38 images taken from slightly displaced viewpoints.

the configuration is detected as 'degenerate' or 'N' is assigned when the configuration is detected as 'noisy'. If neither 'D' or 'N' is detected, it is assigned the value of the accuracy A for synthetic sequence, and Success (S) and Fail (F) of the factorization-based 3-D reconstruction for real sequences (judged by visual inspection of the reconstruction results). The 'X' in the table means that the value is not applicable.

Rows *b0-b_o* and *c0-c_o* of Table 1(a) show the results of the detection of pure rotation and coplanar 3-D points in the presence of nose and outliers. As discussed in Section 3, if all 3-D points are coplanar, or all cameras have the same center, the SMM will be of rank 3. From rows *b0-b4* and *c0-c4* of Table 1(a), we observe that the rank-3 iteration converges with a small E_3 and a large κ_3 in all experiments. Thus, the proposed ADCC algorithm successfully detects all degenerate configurations resulting from pure rotation and coplanar 3-D points in the presence of noise of varying levels. Rows *b_o* and *c_o* represent the cases when outliers are present. In experiment *b_o*, the rank of SMM remains at 3 despite the presence of outliers, which leads to the convergence of the rank-3 iteration.

Row *c_o* represents a case of false detection, where the rank of the SMM increases from 3 to 4 due to the presence of outliers, which leads to the convergence of the rank-4 iteration. Thus, outliers may mask out the degeneracy and lead to the fail of the proposed algorithm. Fortunately, as discussed in C4 in Section 3, outliers can be easily removed using, for example, the Eipolar constraint. Thus, such false detection rarely occurs in practice. Rows *h4, m3* and *m4* in Table 1(b) correspond to the cases of pure rotation for real sequences, which are successfully detected by the ADCC algorithm.

Table 1(b) also shows the G-AIC scores that are computed using Eq. (13) with the following parameters: $N = nm$ (n points tracked along m images), $d = r$ (rank of the SMM), $p = m + n$ (number of projective depths) and $c = 1$ (one point provides

Table 1. Results of detecting critical configurations by ADCC. **(a)**: results on synthetic data. *a0-a_o*: results on counting the number of camera centers; *b0-b_o*: results on detection of pure rotation; *c0-c_o*: results on detection of coplanar 3-D points; The '0' in '*a0*' refers to noise-free data, '1' in '*a1*' means that the std. dev. υ of the noise $\upsilon = 0.0002w$, '2' means $\upsilon = 0.0005w$, '3' means $\upsilon = 0.001w$, '4' means $\upsilon = 0.002w$. The 'o' in '*a_o*' means that there are outliers added to the synthetic data, the same holds for *b0-b_o*, *c0-c_o*. **(b)**: results on real data.

exp	rank-3 ite. E_3	rank-3 ite. κ_3 (%)	rank-4 ite. E_4	rank-4 ite. κ_4 (%)	$\#cc$	A
a0	0.1	89.7	3e-5	100	7	0.05
a1	0.1	89.7	0.003	99.5	7	0.5
a2	0.1	89.6	0.007	99.2	9	1.8
a3	20	38.1	0.01	98.9	9	2.6
a4	0.1	85.9	0.03	98.9	10	7.9
a_o	70	95.8	4	21.4	X	N
b0	3e-7	100	4e-7	70	X	D
b1	0.003	99.9	0.002	99.7	X	D
b2	0.008	99.7	0.006	99.1	X	D
b3	0.02	99.4	6	23.3	X	D
b4	0.03	98.9	8	34.4	X	D
b_o	0.04	98.2	0.1	89.5	X	D
c0	2e-6	100	2e-7	72.2	X	D
c1	0.003	99.6	0.003	99.6	X	D
c2	0.007	99.4	5	61.9	X	D
c3	0.01	99.1	7	48.3	X	D
c4	0.03	98.3	0.06	88.4	X	D
c_o	0.1	96.2	0.03	97.8	5	24

(a) Results on synthetic sequence (Fig. 1(b))

exp	#fm	rank-3 ite. E_3 (pix)	rank-3 ite. κ_3 (%)	rank-3 ite. GAIC3 ($\times 10^3$)	rank-4 ite. E_4 (pix)	rank-4 ite. κ_4 (%)	rank-4 ite. GAIC4 ($\times 10^3$)	#cc	A
h1	20	2.8	96.1	350	0.23	96.9	1.9	5	S
h2	12	2.5	95.8	480	0.27	95.4	3.2	3	S
h3	8	3.3	96.4	450	0.24	96.7	2.2	2	D
h4	4	0.25	99.9	4	0.19	65.7	3.2	X	D
m1	181	3.8	95.7	580	0.31	96.5	4.2	18	S
m2	16	1.5	98	810	0.26	96.6	12	5	S
m3	11	0.54	99.4	74	0.25	88.7	9.6	X	D
m4	8	0.29	99.8	10	0.25	84	8.1	X	D
g1	24	6.6	93.4	2400	0.27	98.9	4.6	24	S
g2	12	6.1	91.8	2300	0.26	98.4	4.9	12	S
g3	8	4.9	92.6	1500	0.29	94.1	14	8	S
g4	4	3	95.3	420	0.24	96.1	3.3	4	F

h1-h4: results on *house1* (Fig. 1(d))
m1-m4: results on *medusa* (Fig. 1(c))
g1-g4: results on *castle* (Fig. 1(e))

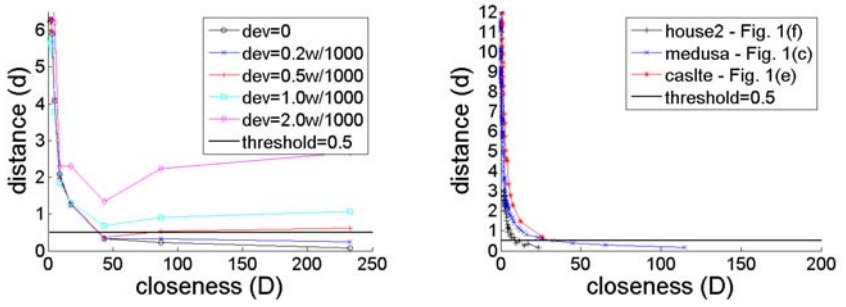
(b) Results on three real sequences

only one constraint). As we see from Table 1(b), the G-AIC scores for rank-4 iterations are consistently smaller than rank-3 iterations, which are not correct for *h3, h4, m3* and *m4* in Table 1(b). The weight between the residual and the model complexity in G-AIC needs to be optimized if it is to be applied. Furthermore, as shown in rows *b3-b_o* and *c1-c4* in Table 1(a), the rank-4 iteration does not always produce smaller residual than the rank-3 iteration. This prevents a direct use G-AIC in our problem.

4.2 Counting Distinct Camera Centers

The ACCC algorithm proposed in Section 3.1 is based on the Proposition 1, which suggests that the maximum angle deviation d between $\mathbf{v}_i$ and $\mathbf{v}_j$ will be small if the $\mathbf{C}_i$ and $\mathbf{C}_j$ is close (with a large closeness D). In this section, we first verify the validity of this proposition, and then ACCC is used to count the camera centers.

Figs. 2 shows the d-D curves obtained in our experiments on both synthetic and real data, where we observe that d monotonously decreases with D for all three real sequences and for the synthetic sequence when $\upsilon < 0.0005w$. This justifies the use of



(a) $d - D$ curves for the synthetic sequence (Fig. 1(b)) with noise of varying levels.

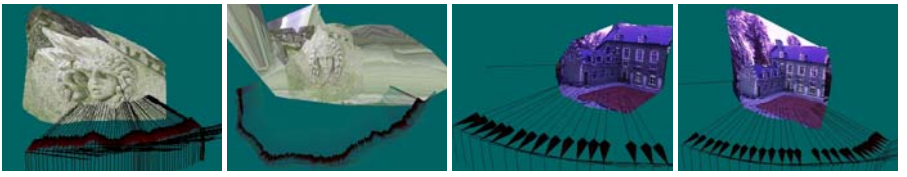
(b) $d - D$ curves for three real sequences.

Fig. 2. $d - D$ curves for synthetic and real sequences

deviation d for inferring the closeness D in the proposed ACCC algorithm. Note that d and D are computed using Eq. (12) and Eq. (11), respectively.

The threshold T_d for grouping the close cameras in the ACCC algorithm is empirically set a value of 0.5 for all experiments. From Figs. 2(a) and 2(b), we note that $D < 40$ if $d > 0.5$, for all real sequences and for the synthetic sequence when $\upsilon < 0.0005w$. With the chosen T_d, our experimental results show that ACCC is able to detect the $\#cc$ robustly. Column $\#cc$ of rows *a0-a4* in Table 1(a) shows the detected $\#cc$ for the synthetic sequence depicted in Fig. 1(b), where we see that 10 cameras are arranged with varying distances between each other. With the rightmost camera taken as the reference camera, the closeness D between each neighboring camera and the reference camera is shown in Fig. 2(a) as the x-coordinates of the data samples. From rows *a0-a2* of Table 1(a), we see 7 camera centers are counted for the first 2 experiments where $\upsilon < 0.0005w$. In these 2 experiments, the 3 rightmost cameras in Fig. 1(c) are grouped with the same center. This is what we expected, since their corresponding D is larger than 40, as seen from Fig. 2(a). The detected $\#cc$ is not correct under large noise levels (Exps. *a2-a4*), because in this case ACCC is no longer able to distinguish the real cause of the large value of d, i.e., whether it stems from the 'distinctness' of the cameras or from the large noise in the measurements. Fortunately, for the real data, the noise levels, which to some extent can be measured by the E_4, are mostly smaller than $0.0005w$ (equivalently around 0.5 pixels for 1024×768 images), as can be observed from our experimental results on real sequences as shown in Table 1(b) and Fig. 2(b).

Rows *h1-h4* in Table 1(b) show the results on the *house1* sequence. In the table, '#fm' denotes the number of images. As we see from the Exp. *h1* in Table 1(b), the rank-3 iteration fails to converge while the rank-4 iteration converges, and the $\#cc$ is correctly counted as $\#cc = 5$, as depicted in Fig. 1(b). Exp. *h2* corresponds to a case with 3 camera centers. Exp. *h3* corresponds to a case of rotation around only 2 camera centers, and Exp. *h4* corresponds to a case of pure rotation. The ADCC correctly counted the number of camera centers in all experiments. Rows *m1-m4* show the experimental results on the *medusa* sequence. In contrast with the *house1* sequence where the camera centers are widely displaced (with a closeness $D < 30$), some adjacent cameras in *medusa* are very close. Using the threshold $T_d = 0.5$, some neighboring cameras are grouped



(a) (b) (c) (d)

Fig. 3. Examples of reconstructed sparse 3-D models: **(a)** 3-D model reconstructed from sseq *s1* in Table 2(b); **(b)** merged 3-D model for *medusa*; **(c)** 3-D model reconstructed from sseq *t1* in Table 2(a); **(d)** merged 3-D model for *castle*.

with the same center though they are actually slightly displaced. That explains why the 181 cameras depicted in Fig. 1(c) are counted as only 18 in Exp. *m1*. In Exp. *m2*, 16 images are used and 5 camera centers are counted, which lead to a successful Euclidean reconstruction. Exp. *m3* corresponds to a degenerate case of rotation around 2 camera centers, and Exp. *m4* corresponds to a case of pure rotation. ADCC successfully detects the critical configurations resulting from insufficient camera disparity. Experiments on the *castle* sequence (rows *g1-g4* in Table 1(b)) show the similar results.

4.3 Dividing Long Image Sequences

Tables 2(a) and 2(b) show the results of the ADLS algorithm on the *castle* (Fig. 1(e)) and *medusa* (Fig. 1(b)) sequences for an automatic factorization-based SaM. From the tables, we observe that the *castle* sequence is divided into 2 subsequences and the *medusa* is divided into 4 subsequences. Euclidean reconstruction is successful for all subsequences judged by visual inspection. The reconstructed 3-D models of the subsequences *t1* and *s1* and the merged 3-D models for the complete *castle* and *medusa* sequences are depicted in Fig. 3. Note that direct factorization-based SaM for *medusa* is not possible since no feature points can be tracked from frame 0 to frame 193. For *castle*, we do track 134 feature points from frame 0 to 25. However, the sequence is detected by the ADCC algorithm as containing outliers. Using ADLS, both sequences are automatically divided into multiple subsequences for individual partial reconstructions, which are thereafter merged into a common coordinate system. The Euclidean reconstruction on long image sequences then becomes possible (the algorithm for merging the partial reconstructions is not presented in this paper).

Table 2. Results on dividing *castle* and *medusa* by the ADLS algorithm

sseq	frames	E_3	κ_3	E_4	κ_4	#cc
t1	0-15	20.8	96.0	0.43	99.0	16
t2	15-25	9.2	96.6	0.30	98.4	11

(a) Results for *castle*

sseq	frames	E_3	κ_3	E_4	κ_4	#cc
s1	0-60	6.3	97.8	0.39	97.6	6
s2	60-124	3.6	98.2	0.34	98.2	5
s3	124-171	12.6	96.1	0.46	98.2	7
s4	171-193	26.7	96.5	0.36	99.3	8

(b) Results for *medusa*

5 Conclusion

We have presented an algorithm for dividing a long image sequence into multiple subsequences for factorization-based 3-D reconstruction, with the consideration of degenerate configurations, noise and outliers and camera disparities. First, a quantitative metric is proposed to measure the closeness of two cameras based on the linear dependency between the row spaces of two corresponding partial scaled measurement matrices (SMMs). Second, an algorithm is proposed to estimate the rank of the SMM by analyzing both the residual error of the projective reconstruction and the singular values of the resulting SMM. By analyzing the row space and the rank of the SMM using a few simple but effective metrics, the critical configurations including coplanar 3-D points, pure rotation and rotation around two camera centers are successfully detected. Our experimental results on both synthetic and real sequences demonstrate that the algorithm is able to robustly detect the mentioned critical configurations. The algorithm provides a practical solution for an automated processing of the factorization-based 3-D reconstruction from long image sequences.

References

1. Han, M., Kanade, T.: A perspective factorization method for euclidean reconstruction with uncalibrated cameras. J. Visual. Comput. Animat. (2002)
2. Sturm, P.F.: Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. Image and Vision Computing 20(5-6), 415–426 (2002)
3. Kahl, F.: Critical motions and ambiguous eucldian reconstructions in auto-calibration. In: Proc. Int. Conf. Computer Vision, vol. 1, pp. 469–475 (1999)
4. Torr, P., Zisserman, A., Maybank, S.: Robust detection of degenerate configurations whilst estimating the fundamental matrix. Int. J. Computer Vision 71(3), 312–333 (1998)
5. Jung, Y.Y., Hwang, Y.H., Hong, H.K.: Frame grouping measure for factorization-based projective reconstruction. In: Proc. ICPR, vol. 4, pp. 112–115 (2004)
6. Chen, Q., Medioni, G.: Efficient iterative solution to m-view projective reconstruction problem. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 55–61 (1999)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
8. Oliensis, J., Member, S.: Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. IEEE TPAMI 29(12), 2217–2233 (2007)
9. Mahamud, S., Hebert, M.: Iterative projective reconstruction from multiple views. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. II, pp. 430–437 (2000)
10. Heyden, A., Berthilsson, R., Sparr, G.: An iterative factorization method for projective structure and motion from image sequences. Image and Vision Computing 17, 981–991 (1999)
11. Kanatani, K.: Uncertainty modeling and model selection for geometric inference. IEEE Trans. Pattern Analysis and Machine Intelligence 26(10), 1307–1319 (2004)
12. Torr, P.: An assessment of information criteria for motion model selection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 47–52 (1997)
13. castle sequence, http://www.cs.unc.edu/~marc

Scene Gist: A Holistic Generative Model of Natural Image

Bolei Zhou[1] and Liqing Zhang[2]

[1] MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems, and Department of Biomedical Engineering, Shanghai Jiao Tong University, No.800, Dongchuan Road, Shanghai, China
zhoubolei@gmail.com

[2] MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems, and Department of Computer Science and Engineering, Shanghai Jiao Tong University, No.800, Dongchuan Road, Shanghai, China
zhang-lq@cs.sjtu.edu.cn

Abstract. This paper proposes a novel generative model for natural image representation and scene classification. Given a natural image, it is decomposed with learned holistic basis called *scene gist* components. This gist representation is a global and adaptive image descriptor, generatively including most essential information related to visual perception. Meanwhile prior knowledge for scene category is integrated in the generative model to interpret the newly input image. To validate the efficiency of the scene gist representation, a simple nonparametric scene classification algorithm is developed based on minimizing the scene reconstruction error. Finally comparison with other scene classification algorithm is given to show the higher performance of the proposed model.

Keywords: image representation, natural image statistics, scene classification.

1 Introduction

One of the extraordinary capabilities of the human visual system is its ability to rapidly group elements from a complex natural scene into the holistic and semantic percept. The studies of cognitive psychology have shown that human can recognize the category of natural scene in less than 150ms when a novel scene image is presented [1,2].

The gist of a novel scene is recognized at a single glance, independent of its spatial complexity. How is this remarkable feat accomplished? One prominent view of scene recognition is based on the idea that a scene is organized from a collection of objects. This notion depicts visual processing as a hierarchical organization of local modules of increasing complexity(gradually from edge to shape, object, then to global scene percept)[3]. On the other hand, psychological results suggest that a scene may be initially represented as a global entity and segmentation of region or object appears at a later stage after the formation of scene gist [2,4].

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 395–404, 2010.

Motivated by the psychological evidence of scene gist components existed in the visual percept of natural image, we propose a novel holistic representation for natural image. The scene gist representation is an *global* image descriptor, adapted to natural image statistics to realize a most compressive encoding. Moreover, the marked performance in the scene classification task proves the superiority of this scene gist representation.

1.1 Related Works

Image representation or descriptor is of fundamental importance to the research of computer vision. It directly deals with organization of pixels, and plays a key role for extracting feature for later processing like feature classification and object recognition. Standard image descriptors, such as SIFT [5], bank of Gabor wavelet and image pyramid [6], have been widely used in feature extraction. In recent years, there have appeared another kind of *adapted* image descriptors drawing our attention. Early work on natural image statistics reveals that the natural image signal is highly non-gaussian and contains much information redundancy[7]. This leads to the headway in the Independent Component Analysis [8] or sparse coding [9] for natural image representation. The adapted basis share the similar response properties to the simple cell in the primary visual cortex. Summarily, the central concept of efficient coding is straightforward: if we want to efficiently capture the feature and reduce the redundancy, the image representation should reflect intrinsic structural properties of natural image [8].

Recent works on scene image modeling are mainly based on local approach, such as bag of words model like pLSA [10] and LDA [11], those methods are mainly through the hierarchical organization of local information to formulate the percepts of scenes. On the other hand, psychological results indicate that human visual system is more likely to rely on global approach to recognize the category of scenes [12]. Oliva et al[4] propose a global representation called Spatial Envelope. However, their holistic modeling of scene is only based on the amplitude of Fourier component coefficients for gray image, which ignores the influence of color information on the visual scene perception [13], and their model is not generative.

Fig. 1. Example images from 8 scene categories in the dataset [4] we work on

Fig.1 shows the example scene images from the scene dataset[4] our model is implemented on. The rest of the paper is organized as follows. Section 2 extends our scene gist model in detail. In Section 3, the scene gist representation is applied to the scene classification experiment, and comparison is given. Section 4 concludes this holistic generative model, and discusses its subspace property related to image manifold research.

2 Scene Gist Generative Model

Psychological study [1] has indicated that human visual system integrates enough information for the category of a scene in about 150ms. What is the underlying computational mechanism in the visual cortex? We consider this problem from the view of signal analysis and reconstruction. A discrete-time input signal $\mathbf{x}$ can be holistically viewed as an N×1 column vector in $\mathbb{R}^N$(we treat a scene image data by vectorizing it into a long one-dimensional vector). To sense and extract the gist of scenes for visual system is like a dynamic filtering and reconstruction process between input signal $\mathbf{x}\in \mathbb{R}^N$ and intrinsic signal representation $\mathbf{s}\in \mathbb{R}^M$, that is:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{1}$$

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{s} \tag{2}$$

where filter basis $\mathbf{W}$ is a projection from the image pixel space $\mathbb{R}^N$ to a representational space $\mathbb{R}^M$, the reconstruction basis $\mathbf{A}$(if $\mathbf{W}$ is full-rank square matrix, $\mathbf{A}$=$\mathbf{W}^{-1}$) recovers the image pixels from a given representational space, and $\hat{\mathbf{x}}$ is the reconstructed image signal.

Our generative model learns nearly-optimized holistic components $\mathbf{A}$ and $\mathbf{W}$ to represent the natural image. When the image signal is projected on these basis to enable M≪N, and to minimize $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$, then, we term $\mathbf{s}$ as the scene gist representation, $\mathbf{A}$ and $\mathbf{W}$ as scene gist components. Before further expending our scene gist representation, two important issues on information redundancy should be discussed, which constitute a theoretical foundation for our model.

2.1 Information Redundancy Revisited

Efficient coding is a general framework under which many mechanisms of our visual processing can be interpreted. Barlow [14] first proposed the efficient coding hypothesis for the purpose of visual processing as to removes information redundancies in the sensory input. The research of natural image statistics [7] also indicates that there is a large amount of redundancy in the visual signal. To sum up, two kinds of information redundancy exist in visual signal processing:

Perceptual redundancy. Perceptual redundancy relates to the human prior knowledge about the visual world. In a sense, prior knowledge is the redundant

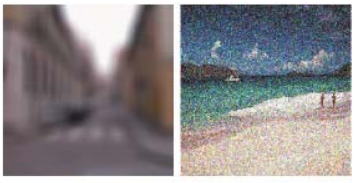


Fig. 2. Natural scene images are heavily blurred and noised respectively, we can recognize them because of prior knowledge for street and coast(better view in color edition)

information that should be suppressed by a coding system [14]. However, human visual system relies heavily on the prior knowledge to interpret the input signal, especially when the input signal is incomplete or noisy. Fig.2 shows two blurred and heavily noised natural scene images, we still can easily recognize them as street and coast because of our prior knowledge on the spatial layout of two scene categories.

Our scene gist model would integrate the prior knowledge in the learned gist components **A** and **W**, to help encode and interpret newly input signal.

Computational redundancy. The other kind is computational redundancy. One of the fundamental problems in computer vision is the curse of dimensionality. The high dimensionality of image weakens the performance of algorithms like object recognition [15]. Hopefully despite of the high dimensionality of image, there are two regularities of natural image that could be utilized. First, natural images are usually embedded in a relatively low dimensional subspace of images, and there are common spatial patterns along the ensemble of the same scene category [16]. Second, for the specific purpose of visual task such as the scene classification, the necessary scale of images might be low.

Fig.3 demonstrates those two regularities of natural image. Fig.3a is the chart of procedure we take: first downsample original 256×256 natural image to 32×32, then resize it to 256×256, so that information of the resized image is equal to the 32×32 image. And Fig.3b shows the result for example images from 8 different categories, orderly, coast, mountain, forest, open country, street, inside city, tall building and highway (refer to Fig. 1 for the example images with original scale). We can see that the low 32×32 scale preserves enough information for recognition. In Fig. 3d, we average 200 natural scene images from 8 different category ensembles separately, arranged in the same order as Fig.3b, we could still recognize the category of those image even the images are heavily blurred and averaged. This example illustrates that from image regularity we still could distinguish the certain semantic information such as the scene category. Moreover, when we average images from two or more different ensembles of scene category together, like in Fig.3c, there is no statistical regularities among the random averaged images, that is to say, the statistical regularities only exist within the

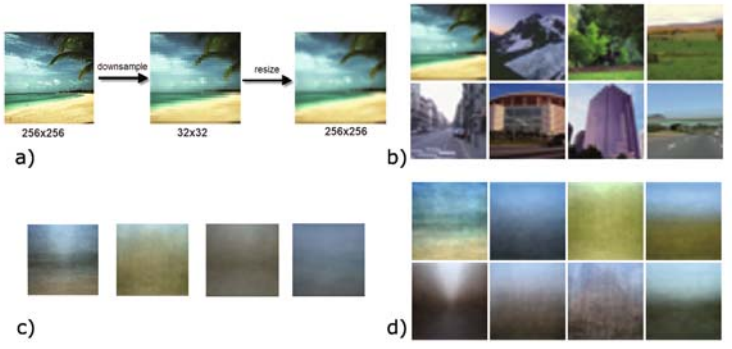


Fig. 3. (a)downsample procedure. (b)32×32 scale images still preserve enough information for scene recognition. (c)average images from different scene category, statistical regularities is not recognizable. (d)average images from same scene category, there exists clearly statistical regularities.(better view in color edition)

same scene category. Those statistical regularities are learned in our model as scene gist components.

2.2 Learning Gist Component

Since we have demonstrated that there are statistical regularities within scene images sampled from same category ensemble, our assumption is that if those regularities were learned as prior knowledge, we could construct highly efficient representation for natural scenes.

Let $\mathbf{X}=[\mathbf{x}^1,\mathbf{x}^2,\dots]$ be the matrix of images from the ensemble of one scene category, $\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\dots]$ be the filter basis, $\mathbf{W}_T$ be the first T rows of $\mathbf{W}$, and $\mathbf{A}_T$ be the first T columns of $\mathbf{A}$. Since given the number of T, the optimal $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$ should minimizes the reconstruction error,

$$\tilde{\mathbf{W}} = \arg\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{A}_T(\mathbf{W}_T\mathbf{X})\|_F, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm, defined as: $\|\mathrm{Y}\|_F = \sqrt{\mathrm{Tr}(\mathrm{Y}^\top\mathrm{Y})}$. According to Eckart-Young theorem [17], the optimal solution to Eq .3 is the PCA basis of the sample matrix, that is, $\mathbf{W}$ is ensemble of the eigenvectors for sample covariance matrix, and $\mathbf{A}=\mathbf{W}^{-1}$. Given the threshold T, and a scene image $\mathbf{x}$, then

$$\hat{\mathbf{s}} = \mathbf{W}_T\mathbf{x} \tag{4}$$

we can reconstruct the scene image to minimize the reconstruction error by:

$$\hat{\mathbf{x}} = \mathbf{A}_T\hat{\mathbf{s}} \tag{5}$$

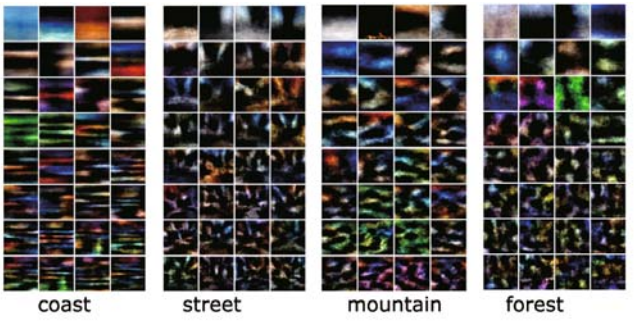


Fig. 4. First 32 Gist components $\mathbf{A}_T^i$ from 4 scene categories i

We learn gist components $\mathbf{A}_T$ and $\mathbf{W}_T$ respectively from 8 category ensembles based on PCA components.[1] In Fig.4, we show first 32 gist components $\mathbf{A}^i$ from 4 scene categories *i*. Obviously it reveals that the gist components have holistic spatial property for corresponding scene categories(refer the example scene images in Fig.1), we can see that the gist components are the holistic components for every scene category.

Because of the adaptive property and orthogonality of PCA basis, the energy of the image signal focuses on the first few principal basis. Fig.5 illustrates that two coast images are projected to the coast PCA basis, the amplitude of coefficients is focused on first few gist components, and the series of small images below the horizontal-axle are the reconstructed images by increasing the threshold T, from 5, 10, 20, 50, 100, 200, 500. Empirically with T around 200, the perceptual loss can be hardly perceived.

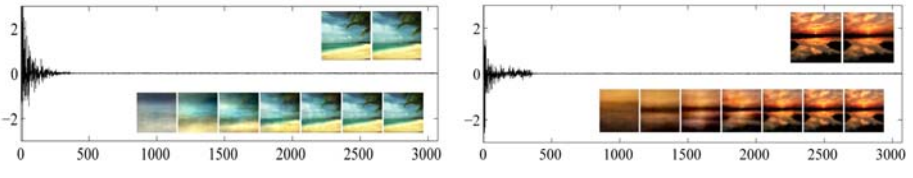


Fig. 5. Two coast images are projected to the 3072 PCA basis. The vertical axle is the coefficient of every PCA basis. The two images above the horizontal axle are the original image and the downsampled image. The series of scene image below is reconstructed by tuning threshold T, from 5,10,20,50,100,200 to 500.

[1] We learn PCA components respectively on down-sampled 32×32 RGB images from different category ensembles, so that learned PCA components is 32×32×3=3072 dimensional.

2.3 Discriminative Property of Gist Components

We have learned the gist component pair ($\mathbf{W}_T^i$,$\mathbf{A}_T^i$) for different scene categories i, then, what is the difference between ($\mathbf{W}_T^i$,$\mathbf{A}_T^i$) and ($\mathbf{W}_T^j$,$\mathbf{A}_T^j$), while i$\neq$ j? It is found that the gist components have discriminative sparse property between different categories: When one scene image is projected to the PCA components from the same scene category, the coefficients of gist components appear to be sparse(focused on first few components), otherwise are not. Fig. 6 gives two examples: a coast scene image is projected to gist components of mountain, and the other one is a mountain scene image projected to gist components of coast, in both conditions the coefficients of basis are not sparse. Then, threshing the coefficients would bring on the reconstructed image signals both great energy and perceptual loss, as shown in Fig. 6.

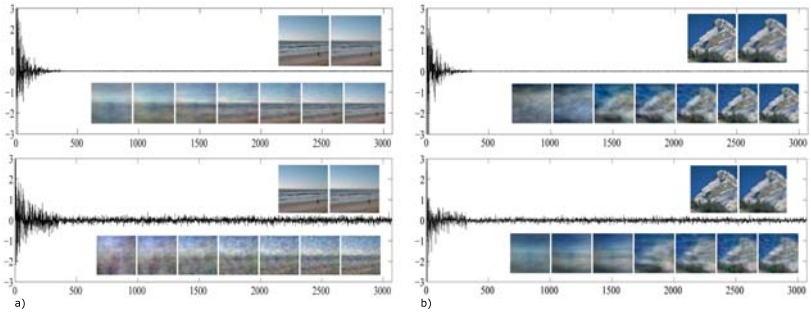


Fig. 6. a) and b) Threshing the gist component coefficients would lead to great signal energy and perceptual loss if scene image is projected to gist components of other scene category

In the following Experiment section, we apply this discriminative property of gist components to develop a simple nonparametric classification algorithm, based on minimizing the scene reconstruction loss.

3 Experiment

Scene classification task is to assign each test image to one category of scene, Fig.1 shows the example scene images from the dataset[4], which includes 8 categories of scenes. The performance is illustrated by a confusion table, and overall performance is measured by the average value of the diagonal entries of the confusion table, see Fig. 7 and Table 2.

3.1 Gist Subspace Classification Method

For each scene category i, we have learned gist components $\mathbf{W}_T^i$ and $\mathbf{A}_T^i$ on corresponding scene category ensembles. Then given one scene image $\mathbf{x}\in \mathbb{R}^N$,

$\mathbf{s}_i$=$\mathbf{W}_T^i\mathbf{x}$, $\hat{\mathbf{x}}_i$=$\mathbf{A}_T^i\mathbf{s}_i$. Relying on the discriminative property of gist components shown in Section 2.3, we can classify $\mathbf{x}$ by assigning it to the scene category i that minimizes the reconstruction error between $\mathbf{x}$ and $\hat{\mathbf{x}}_i$:

$$\min_i \varepsilon_i(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}_i\|_2 = \|\mathbf{x} - \mathbf{A}_T^i(\mathbf{W}_T^i\mathbf{x})\|_2, \tag{6}$$

Algorithm below summarizes the complete scene classification procedure.

Table 1. Gist Subspace classification algorithm

Algorithm : Scene classification

1:**Input**: k pair of gist components ($\mathbf{W}_T^1$,$\mathbf{A}_T^1$),($\mathbf{W}_T^2$,$\mathbf{A}_T^2$),. . . ,($\mathbf{W}_T^k$,$\mathbf{A}_k^1$) for k scene categories, where $\mathbf{W}_T^i \in \mathbb{R}^{T\times N}$,$\mathbf{A}_T^i \in \mathbb{R}^{N\times T}$. And a test scene image $\mathbf{x}\in \mathbb{R}^N$.
2:Compute the gist representation $\mathbf{s}_i$=$\mathbf{W}_T^i\mathbf{x}$, for every pair of components i.
3:Compute the reconstruction errors $\varepsilon_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{A}_T^i(\mathbf{W}_T^i\mathbf{x})\|_2$, for every pair of components i.
4:**Output**: identity($\mathbf{x}$)=$\arg\min_i \varepsilon_i(\mathbf{x})$

For the sake of comparison with other methods, we learn our gist components of each categories from downsampled 48×48 gray images, which ignore the influence of color information, then the threshold T is set empirically as 150, so that the learned gist components $\mathbf{W}_{150}^i \in \mathbb{R}^{150\times 2304}$,$\mathbf{A}_{150}^i \in \mathbb{R}^{2304\times 150}$. Experiment has been repeated ten times with different 200 (75% of each scene category ensemble) randomly selected images for learning scene gist components and the entire ensemble of images for test images for scene classification.

3.2 Result

Fig. 7 shows the confusion table for scene classification. The average performance is 88.75%. In Table 2, we compare our algorithm with other two methods[4,10],

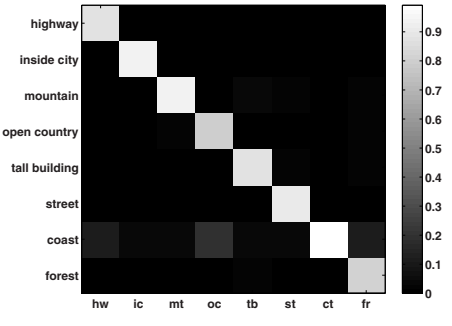


Fig. 7. Confusion table of scene classification. The average performance is 88.75%.

Table 2. Comparison of our algorithm with other scene classification methods

Method	Performance
Gist Subspace	88.75%
Spatial Envelope[4]	83.75%
pLSA[10]	86.65%

from which we can see our Gist Subspace classification method achieves the best performance. This experiment demonstrates the computational efficiency of the scene gist representation.

4 Discussion and Conclusion

The scene gist components $\mathbf{A}_T$ and $\mathbf{W}_T$ include the prior knowledge through the learning process. We conjecture those adapted components may act like the specialized neurons responsible for encoding spatial layout of the scene image in the visual cortex. Our work supports the assumption that there is close relationship between the natural image statistics and the neural representation[7].

On the other hand, there is in-depth implication beyond the discriminative property of gist components. The learning procedure for gist components is based on the Principal Component Analysis. Mathematically, PCA, as the classical linear technique for dimensionality reduction, is to discover the intrinsic structure of data lying on or near a *linear* low-dimensional subspace in the high-dimensional input space. So that different ensembles of scene category may be different subspaces $\mathbb{R}^M$ embedded in high dimensional space $\mathbb{R}^N$, where $M \ll N$, and the different scene gist components approximate the unit basis spanning each subspace. When a natural image is projected to the subspace of the same scene category, energy of signal concentrates on first few basis, otherwise it does not(refer to Fig. 6). That is why the Gist Subspace classification algorithm works. Our findings correspond with previous work on manifold learning [18] that the natural images are embedded in a low-dimensional manifold. Our further work would focus on generalizing a perceptual-meaningful manifold structure for natural images.

To conclude, we propose a holistic generative model of natural image. The scene classification experiment demonstrates its efficiency on representing natural scene image. Moreover, its inner connection to the more general modeling of natural image is discussed.

Acknowledgements

The work was supported by the Science and Technology Commission of Shanghai Municipality (Grant No. 08511501701), the National Basic Research Program of China (Grant No. 2005CB724301) and the Okawa Foundation Research Grant Program, Japan. The first author would like to dedicate this work to his passed-away dear Yan Zhang, and also thank Xiaodi Hou, Qiaochu Tang and Shiteng Suo for their valuable discussion and comments.

References

1. Fabre-Thorpe, M., Delorme, A.: A limit to the speed of processing in ultra-rapid visual categorization of novel natural scenes. J. Cogn. Neurosci. 13(2), 171–180 (2001)
2. Oliva, A.: Gist of the scene. In: Itti, L., Rees, G., Tsotsos, J.K. (eds.) The Encyclopedia of Neurobiology of Attention, pp. 251–256. Elsevier, San Diego (2005)
3. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. W.H. Freeman, New York (1982)
4. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42, 145–175 (2001)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
6. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, Englewood Cliffs (2002)
7. Simoncelli, E.P., Olshausen, B.: Natural image statistics and neural representation. Annual Review of Neuroscience 24, 1193–1216 (2001)
8. Bell, A.J., Sejnowski, T.J.: The "independent components" of natural scenes are edge filters. Vision Res. 37(23), 3327–3338 (1997)
9. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583), 607–609 (1996)
10. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
11. Li, F.F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proc. CVPR, pp. 524–531 (2005)
12. Schyns, P.G., Oliva, A.: From blobs to boundary edges: Evidence for time and spatial scale dependent scene recognition. Psychological Science 5, 195–200 (1994)
13. Castelhano, M.S., Henderson, J.M.: The influence of color on the perception of scene gist. Journal of experimental psychology, Human perception and performance 34(3), 660–675 (2008)
14. Barlow, H.: Redundancy reduction revisited. Network 12(3), 241–253 (2001)
15. Riesenhuber, M., Poggio, T.: Models of object recognition. Nature neuroscience 3(suppl.), 1199–1204 (2000)
16. Torralba, A., Oliva, A.: Statistics of natural image categories. Network (Bristol, England) 14(3), 391–412 (2003)
17. Deprettere, E.F. (ed.): SVD and signal processing: algorithms, applications and architectures. North-Holland Publishing Co., Amsterdam (1988)
18. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)

A Robust Algorithm for Color Correction between Two Stereo Images

Qi Wang, Xi Sun, and Zengfu Wang*

Dept. of Automation, University of Science and Technology of China
crabwq@mail.ustc.edu.cn, snxi@mail.ustc.edu.cn,
zfwang@ustc.edu.cn

Abstract. Most multi-camera vision applications assume a single common color response for all cameras. However, significant luminance and chrominance discrepancies among different camera views often exist due to the dissimilar radiometric characteristics of different cameras and the variation of lighting conditions. These discrepancies may severely affect the algorithms that depend on the color correspondence. To address this problem, this paper proposes a robust color correction algorithm. Instead of handling the image as a whole or employing a color calibration object, we compensate for the color discrepancies region by region. The proposed algorithm can avoid the problem that the global correction techniques possiblely give bad correction results in local areas of an image. Many experiments have been done to prove the effectiveness and the robustness of our algorithm. Though we formulate the algorithm in the context of stereo vision, it can be extended to other applications in a straightforward way.

Keywords: Color correction, stereo images, OF-SIFT, mean-shift.

1 Introduction

Stereo vision has traditionally been, and continues to be, one of the most extensively investigated topics in computer vision. Generally, a vision algorithm employs two or more images to recover depth information of a specific scene. In the algorithms presented, some assumptions about the physical world and the image formation process are used explicitly or implicitly. For example, surfaces in the scene are assumed to be Lambertian ones whose appearance does not vary with viewpoint and the multiple cameras employed are assumed to have uniform properties. Based on these assumptions, numerous algorithms have emerged and the increased sophistication of newer algorithms is producing a commensurate improvement in their performance [1].

However, these assumptions are not always true in real applications. For example, one can obtain stereo images with consistent color appearance in some cases, such as

* Corresponding author: Zengfu Wang. Dept. of Automation, University of Science and Technology of China, Hefei, Anhui 230027, P.R.China. This work was supported by the National Natural Science Foundation of China (No.60875026).

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 405–416, 2010.

when the Benchmark images in the famous Middlebury dataset [2] are used to be stereo images, but it is not appropriate in a real application. In a real environment, significant luminance and chrominance discrepancies among different camera views often exist due to the dissimilar radiometric characteristics of different cameras — even of the same type, and the variation of lighting conditions. These discrepancies may severely affect the algorithms that depend on the color correspondence and these algorithms abound in stereo vision field [2]. So it is necessary to consider the color correction problem in such an application. Our aim in this paper is to present a color correction method to ensure the color consistency between multi-camera views.

Previous work aimed at color correction mainly falls into two categories: calibrating cameras in order to obtain some desired response and processing images after acquisition. A common approach taken toward the first category is to calibrate each camera independently through comparisons with known colors on a color calibration object [3, 4]. Though a feasible solution, it is indeed an inconvenient and complex procedure. Furthermore, once the system is moved to another environment, the whole procedure must be repeated again. The other category involves a large variety of techniques, such as histogram matching [5], multispectral imaging technique [6], energy minimization in camera view networks [7], dominant basic color mapping [8], general color transfer method [9], selective color correction [10], etc. These techniques can compensate for color discrepancies between two images by using the global color distribution information of the two images. The advantage of the techniques is that they don't require a standard reference object for color calibration. However, they sometimes give bad correction results in local areas of an image. Besides the two categories mentioned above, color correction has also been studied in the fields of printers, scanners and monitors [11, 12]. But few of the corresponding techniques developed have been extended to camera systems.

In this paper, we propose a new color correction method based on image segmentation and keypoint matching. The general idea of our method is, instead of handling the image as a whole or employing a color calibration object, we compensate for the color discrepancies region by region. Regions and color discrepancies are acquired by segmenting the reference image and by comparing the color information of matched keypoints extracted from the images respectively. Here two state-of-the-art techniques are employed. One is mean-shift based segmentation technique [13] and the other is SIFT keypoint extraction technique [14]. We also present a novel optical flow based algorithm for SIFT keypoint matching — OF-SIFT, which can greatly speed up the keypoints matching. Though we formulate the problem in the context of stereo vision, the proposed method can be easily extended to other applications.

The rest of this paper is organized as follows. Section 2 introduces our color correction algorithm. Section 3 gives a detailed description of OF-SIFT. Experimental results are shown in Section 4, and conclusions are finally given in Section 5.

2 Methodology

Our aim in this paper is to make the color appearances of two stereo images consistent with each other. The two images are respectively named as target image and source

image. We use the proposed method to adjust the source image so that it conforms to the target image in color appearance. Since our focus is on color correction procedure, we assume that the two images used in this paper are geometrically calibrated and the epipolar constraint is basically satisfied between them. This is an acceptable and reasonable assumption. For one thing, geometric calibration is a sophisticated technique [15] and can be done easily by using a known toolkit. For another, based on this assumption, we can keep our attention on the main problem of color correction.

In our procedure, we first extract SIFT keypoints from the two images and get matched pairs with OF-SIFT. Then the source image is segmented by mean-shift segmentation algorithm. The following color correction is performed region by region. For a given region in the source image, the color discrepancy is calculated by averaging the color discrepancies of matched pairs within it. If there are no matched pairs within the region, five keypoints will be added to the region and their corresponding matches in the target image will be identified. In the following, we will clarify our color correction algorithm step by step.

(1) Step 1: image acquisition. In the first place, two images, target image and source image, are acquired from two stereo cameras and we assume that they are geometrically calibrated. The source image is different from target image in color appearance for some known or unknown reasons.

(2) Step 2: color image transformation. Both of the acquired images, which are usually obtained in RGB color components, are transformed into the gray scale images and the HSI color images. Here, the gray scale images are used for SIFT keypoint extraction and the HSI color images are for the color correction.

(3) Step 3: SIFT keypoint extraction and matching. SIFT is one of the most widely used feature point detection technique in computer vision. The SIFT features are invariant to image scale and rotation, and can provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. Therefore, there can be hundreds of matched pairs, even though the source image differs greatly with the target image. However the original SIFT algorithm searches keypoint pairs in an exhaustive manner and is time-consuming. In order to speed up the process, we use the OF-SIFT algorithm we proposed, which will be discussed in Section 3, to find matched pairs.

(4) Step 4: segmenting source image and calculating color discrepancy. In this step, the source image is segmented by using mean-shift based segmentation algorithm. Most of the segmented regions have SIFT keypoints, whose counterpart matches are in the target image. These regions are called matched regions. But there are other regions that do not have any SIFT keypoints within them for the extracted SIFT keypoints are sparse and do not have a uniform distribution. In this case, they are called unmatched regions.

For a given matched region S, we calculate its color discrepancy with the target image by averaging the color discrepancies between the SIFT keypoints in this region and its corresponding matches in the target image. Note that both of the images are transformed into HSI ones. Therefore, the color discrepancies are computed on three channels separately. We use the following color correction function to correct every pixel's color information in the region,

$$C_{new}(i,j) = C_{old}(i,j) + \left(\sum_{x_s \in S} [CN(x_t) - CN(x_s)] \right) \Big/ keyNumInS . \quad (1)$$

where (i, j) is a pixel of the region S, $C_{old}(i, j)$ is the value of the original color component of the pixel (C can be H , S or I) , and $C_{new}(i, j)$ is the corresponding new color value after correction of $C_{old}(i, j)$, x_s is a SIFT keypoint in region S of the source image, x_t is its corresponding match in the target image, $keyNumInS$ is the number of keypoints in region S, and $CN(x_s)$ and $CN(x_t)$ are the mean values of the colors within a 3×3 neighbor of x_s and x_t respectively.

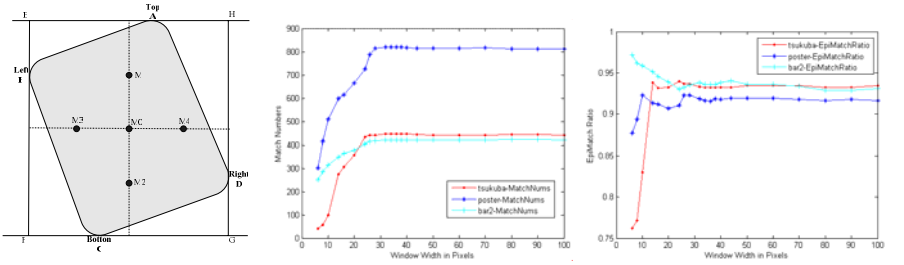
For a given unmatched region, we compute its color discrepancy by using five keypoints. Fig. 1(a) shows the detail of this procedure. Suppose the shadow area in Fig. 1(a) is an unmatched region to be processed. In the first place, we find the four boundary points of the region: Top A, Left B, Bottom C and Right D. Then a rectangle EFGH is formed based on the four boundary points. After that, perpendicular bisector of each edge is drawn and we get two line segments between the two pairs of opposite edges. At last, we trisect each line segments with three points and label them all in the figure. There are five points (M0, M1, M2, M3 and M4) in together because two centre points are coincident in position. Therefore, the unmatched region has five added points as keypoints. The added keypoints are located the way that we explained above because we want them to sample as much color information of the entire region as possible. In addition to what we have done, another existing SIFT keypoint outside the unmatched region is also needed as a reference point for the five added keypoints. It is identified by finding the nearest SIFT keypoint to the added keypoint M0. In that case, we can easily find corresponding points of the five added keypoints according to their relative position to the reference keypoint and the corresponding match of the reference keypoint in the target image. After that, we can compute the color discrepancy in the same way as matched regions.

(5) Step5: color correction region by region. For each region in the source image, we first calculate the color discrepancy according to the way discussed in Step 4. Then all the pixels belonging to this region are corrected according to the region's color discrepancy. This process is repeated until all the regions are processed.

3 OF-SIFT

Each SIFT keypoint is described by a 128 high dimensional vector. To get the best candidate match for a SIFT keypoint in the source image, the closest neighbor and the second-closest neighbor should be identified first. The closest neighbor is defined as the keypoint with minimum Euclidean distance in the target image and the second-closest is the one with the second closest distance. After that, a ratio of the closest distance to the second closest distance is calculated to determine whether to accept the closest neighbor as the correct match or discard it as a false match. This is the way SIFT algorithm takes to obtain the matched pairs.

However, an exhaustive search process is involved to establish each pair of matched keypoints. That means, for each keypoint in the source image, that we need to calculate the distances of the keypoint and all of the other keypoints in the target image to determine the closest and second-closest neighbors. Obviously, it is a time-consuming process. In order to speed up the processing, we present a new algorithm based on optical flow calculation – OF-SIFT.



(a) (b) (c)

Fig. 1. (a) is an illustration of how to add five keypoints for an unmatched region in step 4 of our color correction method. (b) and (c) are the keypoint matching performance varied with widow width in OF-SIFT.

The general idea of OF-SIFT is as follows. For each keypoint in the source image, we calculate its optical flow by Horn-Schunks algorithm [16]. According to the optical flow, we can get an estimation of the potential match's position in the target image. However, this estimation is not accurate enough because of the existence of noise and the algorithm itself. Subsequent keypoint matching is proceeded within a rectangle window centered at the estimated point. When making a decision on the true matched pair, a relaxed epipolar constraint should be satisfied. This means that when a keypoint in the source image lies in scan line i, its corresponding match in the target image should lie in scan line i-1, i or i+1. We check the matched pair this way because generally speaking, the calibration can not be so precise that the corresponding match lies exactly in the same scan line. Besides that, we also set a discrepancy threshold T ranged [0, 1] to control the acceptance level of color discrepancy (In our process, the HSI channels are all scaled to [0, 1]). If the color discrepancy of the matched pair is smaller than T, we accept them as a true matched pair. Otherwise, they are discarded as a false matched pair.

For the selection of the matching window size, we have the following considerations. Since the images in our experiment are geometrically well calibrated and we assume they satisfy the relaxed epipolar constraint, the height of the matching window is set to be 3 pixels. This is able enough to tolerate some errors in actual practice. As for the width of the matching window, we have done a lot of experiments to examine its effect on matching performance. All the 38 pairs of stereo images in our experiments are selected from the Middlebury dataset [2]. They are all well calibrated. We vary the window width to see its influence on the numbers of true matches and the true matching rates. In order to see the results clearly, graphs of only three pairs of images (Tsukuba, Poster and Bar2) are shown in Fig. 1(b, c). Other results that do not

appear in the Fig. are nearly the same as those of the three. The graph (b) in Fig. 1 shows that with the increase of window width, the total matched pairs increase accordingly. But when the width is larger than forty, the numbers remained unchanged. We can also find from graph (c) that the true matching rates vibrate violently when the width is small. However, when the width increases larger than forty, the curves stay flat. Therefore, a width of forty pixels is an appropriate selection for the matching window, because larger window will not improve the performance. Instead, it will cost much more processing time.

4 Experiments

Many experiments are done to examine the performance of OF-SIFT and the presented color correction algorithm. All the 38 pairs of stereo images mentioned in Section 3 are employed in our experiments. Due to space limitation, only results of the mostly used four pairs (Venus, Tsukuba, Teddy, Cones) are reported here. All the programs are run in a computer with Pentium 4 CPU 2.93GHz and 1G memory.

4.1 Performance of OF-SIFT

The original SIFT algorithm was implemented by the author in Matlab environment. In order to compare our OF-SIFT with SIFT fairly, all the programs are run in Matlab environment.

Table 1 lists the experimental results. The first column in the Table is the aspects to be compared. They are the true matched pairs, the matching operations (One matching operation means to calculate the Euclidean distance between two keypoints one time.), time consumed in the whole matching process, reduced matching operation and time compared with SIFT respectively. We can see clearly that our OF-SIFT can find more matched pairs than SIFT. That's because SIFT searches the matched keypoints in the entire target image. In order to avoid the false matches, the strict judgement is needed. On the contrary, OF-SIFT restricts the searching scope to a reasonable smaller window area and the candidates of the matched pairs obtained are more likely to be true. The Table also indicates that OF-SIFT costs far less matching operations, therefore far less time than SIFT. At least 97% matching operations and matching time can be saved. This is a critical advantage for real-time applications and it demonstrates the efficiency of OF-SIFT.

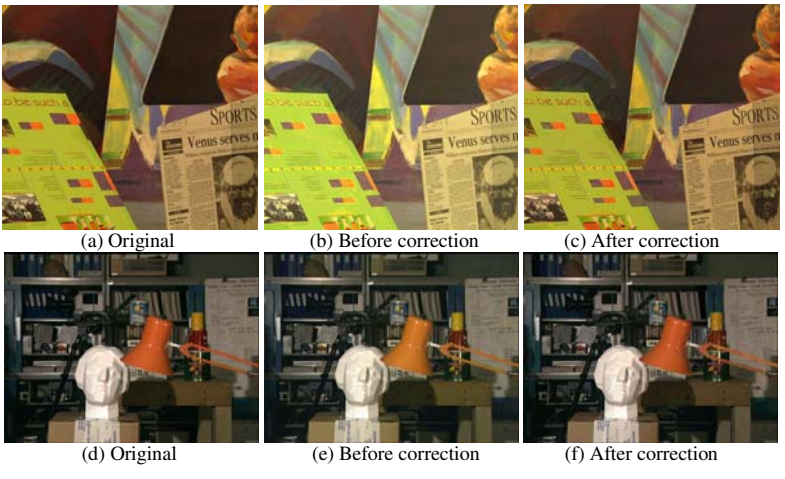
Table 1. Comparison of SIFT and OF-SIFT. For the consideration of limited paper length, results of only four images are reported here.

	Venus	Tsukuba	Teddy	Cones
SIFT/OF-SIFT True Matched Pairs	397/450	344/416	417/486	641/753
SIFT/ OF-SIFT Matching Operations	532170/6141	595968/7343	911028/7723	2202175/14046
SIFT/ OF-SIFT Matching Time (s)	4.697/0.060	5.119/0.111	8.271/0.070	21.368/ 0.120
Reduced Matching Operations (%)	98.84	98.76	99.15	99.36
Reduced Matching Time (%)	98.72	97.83	99.15	99.44

4.2 Performance of Color Correction Algorithm

In this part, we will examine the performance of our color correction algorithm. Image pairs for experiments are all from Middlebury dataset. In each pair, one image is adjusted to be different from the other in color appearance. The adjusted image is treated as source image and the other one as target image. Then we use our proposed algorithm to correct the adjusted image to be the same as the original one as possible. The reason for employing the Middlebury dataset and adjusting one image as the source image is based on the following considerations. Middlebury dataset is a publicly available and widely accepted dataset in stereo vision. All the image pairs are well calibrated and there is ground truth depth information for each pair. If we use these images to conduct our experiments, the acquired results can be objectively compared with the standard known information from the dataset. That will make our conclusion more convincing.

We adjust the images separately on three channels of H, S and I, using the Photoshop software. We select regions of different sizes in the image to adjust their colors. Different regions have different adjustments and the largest variation in pixel value from the original image is up to 25%. This means we set the threshold T as 0.25. We program our algorithm in Microsoft Visual Studio .NET 2003 environment and the experimental results are evaluated from the following three aspects.



(a) Original (b) Before correction (c) After correction

(d) Original (e) Before correction (f) After correction

Fig. 2. Illustration of color correction results for subjective evaluation. The first row lists Venus images and the second Tsukuba. Each row from left to right, is respectively the original image, adjusted image before correction and result image after correction.

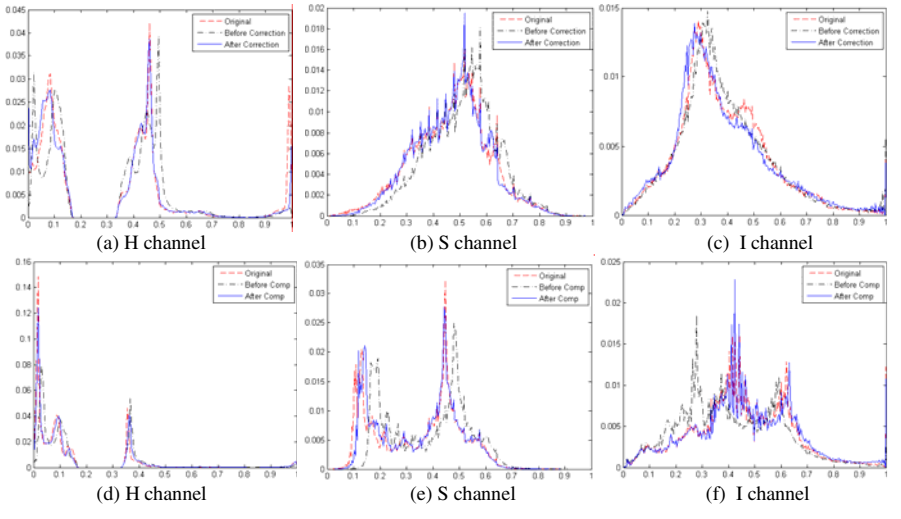
4.2.1 Subjective Evaluation

To compare color information qualitatively is a known hard problem. Objective evaluation usually gives a qualitative value to represent the goodness of the results. But it does not necessarily coincide with the human perception. This was noted by Cinque et al. [17]:"Although it would be nice to have a quantitative evaluation of performance given by an analytical expression, or more visually by means of a table

or graph, we must remember that the final evaluator is man." Therefore, we should assess our color correction results in a subjective manner. Fig. 2 displays the results of two pairs of images, Venus and Tsukuba. Each row from left to right, they are respectively the original image, adjusted source image before correction and the result image after correction. We can see clearly that the adjusted image differs greatly with the original one. Furthermore, since different regions of one image are adjusted differently, their color discrepancies are not the same. But after correction of our algorithm, the result images look nearly the same as the original one.

4.2.2 Histogram Evaluation

We also assess our color correction results in the form of histogram comparison. Color channels are compared separately. In order to see the difference clearly, we draw the histogram envelop curves of the original image, adjusted source image before correction and the result image after correction, together in one graph. The range of the three horizontal axes is scaled to [0, 1] and that of the vertical axes is normalized by the total pixel number of the image. Obviously from Fig. 3 we can see that, histograms of the Venus and Tsukuba result images have a greater resemblance with the original ones than that of the adjusted images.



(a) H channel (b) S channel (c) I channel

(d) H channel (e) S channel (f) I channel

Fig. 3. Illustration of color correction results for histogram evaluation. The first row lists Venus results and the second Tsukuba. Each row from left to right, they are respectively the histograms of H, S and I channels.

4.2.3 Stereo Vision Evaluation

We also evaluate our algorithm in the context of stereo matching. One state-of-the-art stereo matching algorithm, which is based on cooperative optimization [18] and is the best ranked algorithm [2], is employed in our experiment. The adjusted source image before correction is first used to calculate the disparity map. Then the result image is used the same way. At last, their results are compared with the ground truth disparity

map issued by the Middlebury website. Fig. 4 shows our experimental results of Venus and Tsukuba image pairs. We can find that the disparity maps calculated by the adjusted images have much noise and differ greatly with the original ground truth maps. But after correction, they are much smoother and show greater resemblance to the ground truth disparity maps. Due to the sake of exactly known disparity information, we can give a quantitative comparison of the color correction performance. We compare our calculated disparity maps with the ground truth ones. Pixels diverting from the ground truth values larger than one are treated as bad pixels. Then we compute the percentage of the bad pixels in the entire map. Table 2 shows the results, from which we can easily see that the error rates drop sharply after our color correction process. Besides, the poor performance of stereo algorithm confronting with the color discrepancy also reveals an existing problem that, some stereo matching algorithms neglect the color correction procedure. Although Middlebury website provides convenient stereo image pairs that are well calibrated as a platform to compare stereo matching algorithms, we may encounter the challenging problem in real applications that the images captured by different cameras may not coincide with each other in color appearance. In this case, even the best existing stereo matching algorithm may not work perfectly. This reflects the meaning of our work from another point of view.

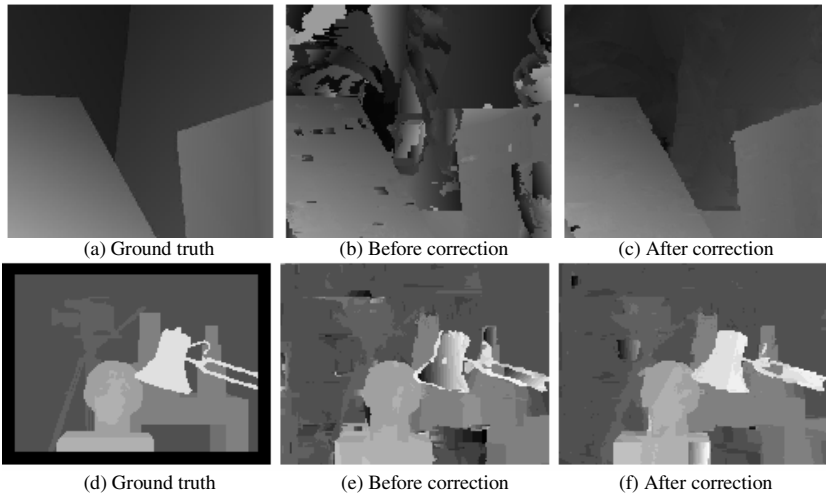


Fig. 4. Illustration of color correction results for stereo vision evaluation. The first row lists Venus disparity maps and the second Tsukuba. Each row from left to right, is respectively the disparity maps of original ground truth, calculated by using the adjusted image before correction and by the result image after correction.

4.3 Experiments under Extreme Conditions

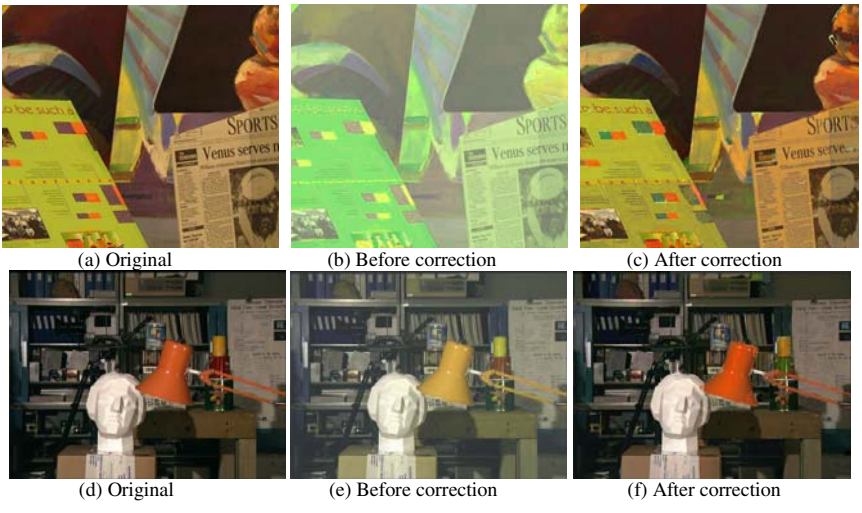
Experiments in Section 4.2 are conducted when the source and original images have a discrepancy up to 25% in color value. In this part, we evaluate our algorithm in the condition of greater discrepancy. Fig. 5 displays the results of Venus and Tsukuba image pairs. Different regions of source image have different adjustments and the

Table 2. Error rates comparison of disparity calculation before and after color correction process

	Error Rates of Different Images			
	Venus	Tsukuba	Teddy	Cones
Before Correction(%)	44.64	12.37	35.26	64.90
After Correction(%)	4.17	6.46	15.01	24.85

most salient variation in pixel value from the original image is up to 80%. That means we set the discrepancy threshold T as 0.8.

From the results, we can see that the adjusted source images look absolutely different from the original one. Fortunately, our algorithm works well enough to correct most of the regions to make them resemble to the original ones. But there are still some regions not properly corrected because of the inaccurate color discrepancies, which are resulted from the false matched pairs. Although the relaxed OF-SIFT can reduces false matches compared with SIFT, it can not eradicate mismatches from happening. Therefore, when there are regions with false matched pairs, their color correction results may not be correct.



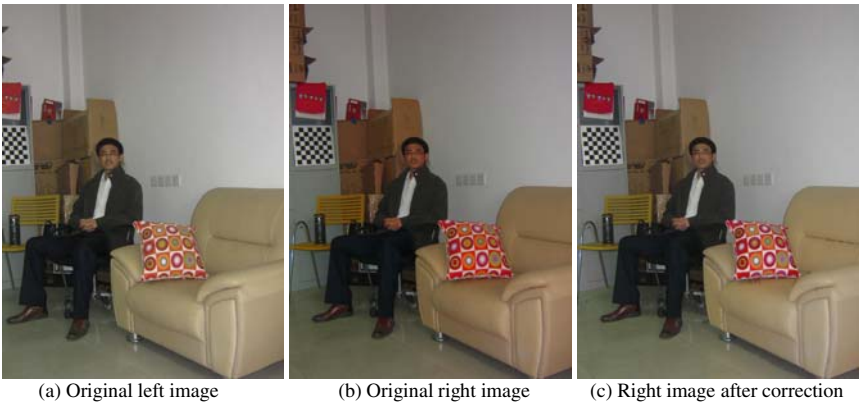
(a) Original (b) Before correction (c) After correction

(d) Original (e) Before correction (f) After correction

Fig. 5. Color correction results under extreme conditions. The first row lists Venus images and the second Tsukuba. Each row from left to right, is respectively the original image, adjusted image before correction and result image after correction.

4.4 Experiments on Real World Scene

The images used in the above experiments are all from Middlebury dataset. In this part, we present our results on real world scene. The stereo images are taken by two cameras, FinePix S5000, with different parameter settings. The original right image



(a) Original left image (b) Original right image (c) Right image after correction

Fig. 6. (a) and (b) are the original image pair taken from stereo cameras. (c) is the right image after color correction.

differs greatly in color appearance with the actual scene color. But after our correction process, their color consistency is improved a lot. Fig. 6 shows our results.

5 Conclusions

In this paper, we present a color correction algorithm to compensate for the color discrepancy between two stereo images. Instead of correcting the image in a global manner or employing a calibration object, our color correction process is conducted region by region. This makes our method more convenient and accurate. Many experiments have also been done to prove the efficiency and robustness of the proposed algorithm. But for the consideration of limited paper length, only a few of them are reported. The results of other image pairs are consistent with the conclusion.

We also present an optical flow based algorithm to speed up the SIFT keypoint matching process. This is indeed effective. However, SIFT keypoint extraction and mean-shift based segmentation are both time-consuming. How to find a more rapid as well as more robust keypoint extraction and image segmentation methods is a challenging work in the future. In addition, how to eradicate mismatches of SIFT keypoints is another problem to be researched.

Although we formulate the color correction problem in the context of stereo vision, the presented algorithm can be extended to other applications in a straightforward way. For example, if there are more than two cameras in a vision system, we chose one as the reference camera. Images captured by other cameras can be separately corrected according to the target image from the reference camera. Another example can be found in multi-view video coding. In this field, different coding schemes have been proposed which explore not only temporal correlation between subsequent frames but also the special correlation between neighboring camera views. Unfortunately, ununiform camera responses often exist. In this case, the presented algorithm can be helpful.

References

1. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-frame Stereo Corrspondence Algorithms. International Journal of Computer Vision 47(1/2/3), 7–42 (2002)
2. http://vision.middlebury.edu/stereo/
3. Ilie, A., Welch, G.: Ensuring Color Consistency across Multiple Cameras. In: Proc. Tenth IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 17–21 (2005)
4. Unal, G., Yezzi, A.: A Variational Approach to Problems in Calibration of Multiple Cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(8), 1322–1338 (2007)
5. Chen, Y., Cai, C., Liu, J.: YUV Correction for Multi-View Video Compression. In: Proc. 18th International Conference on Pattern Recognition (ICPR), vol. 3, pp. 734–737 (2006)
6. Cherdhirunkorn, K., Tsumura, N., Nakaguchi1, T., Miyake, Y.: Spectral Based Color Correction Technique Compatible with Standard RGB System. Optical Review 13(3), 138–145 (2006)
7. Yamamoto, K., Oi, R.: Color Correction for Multi-view Video Using Energy Minimization of View Networks. International Journal of Automation and Computing 5(3), 234–245 (2008)
8. Shangguan, L., Sun, J.: Multi-View Video Coding Using Color Correction. In: Workshop on Power Electronics and Intelligent Transportation System, pp. 149–152 (2008)
9. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color Transfer between Images. IEEE Computer Graphics and Applications 21(5), 34–41 (2001)
10. Inoue, A., Tajima, J.: Selective Color Correction for Arbitrary Hues. In: Proc. International Conference on Image Processing, vol. 3, pp. 38–41 (1997)
11. Bala, R., Sharma, G., Monga, V., Van de Capelle, J.-P.: Two-Dimensional Transforms for Device Color Correction and Calibration. IEEE Transactions on Image Processing 14(8), 1172–1186 (2005)
12. Kang, H.: Color Technology for Electronic Imaging Devices. SPIE-International Society for Optical Engineering (1997)
13. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
14. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
15. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, Englewood Cliffs (2002)
16. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Artificial Intelligence (1981)
17. Cinque, C., Guerra, C., Levialdi, S.: Reply: On the Paper by R. M. Haralick. CVGIP: Image Understanding 60(2), 250–252 (1994)
18. Wang, Z.F., Zheng, Z.G.: A Region based Stereo Matching Algorithm Using Cooperative Optimization. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)

Efficient Human Action Detection Using a Transferable Distance Function

Weilong Yang, Yang Wang, and Greg Mori

School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
wya16@sfu.ca, ywang12@cs.sfu.ca, mori@cs.sfu.ca

Abstract. In this paper, we address the problem of efficient human action detection with only one template. We choose the standard sliding-window approach to scan the template video against test videos, and the template video is represented by patch-based motion features. Using generic knowledge learnt from previous training sets, we weight the patches on the template video, by a transferable distance function. Based on the patch weighting, we propose a cascade structure which can efficiently scan the template video over test videos. Our method is evaluated on a human action dataset with cluttered background, and a ballet video with complex human actions. The experimental results show that our cascade structure not only achieves very reliable detection, but also can significantly improve the efficiency of patch-based human action detection, with an order of magnitude improvement in efficiency.

1 Introduction

This paper considers the problem of human action detection. Given a template video clip containing an actor performing a particular action, we would like to localize similar actions in our test videos. A closely related problem is action recognition, whose primary goal is to classify a video sequence into one of several pre-defined categories. The goal of action detection is distinct from that of action recognition – we would like to localize the specific position (in both time and space) of the target action in a video, rather than getting a single class label. In particular, we are interested in the scenario where the target action is specified using a *single video clip*. This is a natural and realistic scenario in many real-world applications, *e.g.*, surveillance, video retrieval, *etc.*

There is a large literature on action recognition and detection. Moeslund *et al.* [1] provide a survey of the literature on action recognition. We only give a brief review of the closely related work here. Niebles *et al.* [2] run an interest point detector over video sequences, then apply latent topic models to categorize and localize human actions. Ke *et al.* [3] apply the AdaBoost learning framework to the task of human action detection, and volumetric features are used for efficient video analysis. Laptev and Pérez [4] use a similar boosted space-time window classifier to localize human actions in movies. All these learning based

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 417–426, 2010.

approaches heavily rely on a large number of training examples. However, in many real-world applications, it is unrealistic to assume that we have access to a large amount of training data. For example, in the context of example-based video retrieval, we typically have only *one* short video clip submitted by the user.

One example of action detection with only one template is the work of Shechtman and Irani [5]. They compute the distance between two videos by exhaustively comparing patches centered around every space-time point. We use a similar patch based matching method, but we weight patches by their saliency, leading to a more efficient algorithm. In [6], Ke *et al.* propose a template matching scheme combined with a part based pictorial structure model to detect actions in crowded scenes with only one template. The limitation of this work is that one has to manually segment the parts (in space/time volumes), which can be time-consuming.

In our previous work [7], a patch based matching scheme is used for action recognition with a single clip as the template. We also propose a *transferable distance function* in [7] to weight those patches by their saliency. The transferable distance function is learnt from previously training sets, and can be applied to videos of new actions without further learning. The work presented here is based on [7]. However, in this paper, our main goal is to address human action detection, which does not require the pre-processing human detection and tracking step on test videos as [8,7]. The main contributions of this paper are two-fold, in addressing the efficiency issues. First, we propose a variant of the motion feature in Efros *et al.* [8] using a histogram representation. This feature representation can be computed efficiently using integral images. Second, we propose a cascade structure for action detection with only one template, which is based on the transferable distance learning framework of [7], and significantly boosts the efficiency of our approach.

2 Human Action Detection

Given a template action, the objective of human action detection is to localize all similar actions in test videos. In this paper, we choose the standard sliding-window approach, that is to slide the template action video clip T over all locations on the test video V. The distance between T and V at location l is denoted as $D(T, L)$, where L is the video segment of V centered around location l. An action is detected if the distance falls below a threshold. To compute the distance $D(T, L)$, we choose the patch-based action comparison approach proposed in [7]. However, we represent the motion feature using a histogram of four-channel motion flow, which enhances the efficiency of action detection.

2.1 Motion Feature

Our motion feature is a variant of the descriptor proposed by Efros *et al.* [8] which has been widely used in action recognition. First, we compute the optical

flow at each frame, then split the optical flow vector field F into the horizontal and vertical components, F_x and F_y. They are further half-wave rectified into four non-negative channels F_x^+, F_x^-, F_y^+, F_y^-. Then, those four channels are blurred using a Gaussian kernel.

One of the limitations of this four-channel descriptor is its large size. For a small 20×20 patch, the dimensionality of the four-channel descriptor is $4 \times 20 \times 20 = 1600$. The distance between two feature vectors cannot be computed efficiently with such a high dimensional feature. In this paper, we break the patch into 4×4 cells. Each cell is represented by a four-bin histogram, where each bin corresponds to one channel in the four-channel motion descriptor [8]. The value of each bin is the accumulation of the weighted votes of all pixels in the cell. In the end, we will obtain a feature vector with dimensionality only $4 \times 4 \times 4 = 64$. This motion feature is closely related to the histogram of optical flow used in [4]. The similarity between two feature vectors can be computed using normalized correlation or Euclidean distance. Moreover, to efficiently compute feature vectors, the *integral image* representation [9] is used for each histogram bin.

2.2 Patch Based Action Comparison

For the task of action detection, when using only one template, generalization is normally very difficult because of the intra-class variation among actors. In order to alleviate the effect of this variation, Ke *et al.* [6] manually break the template model into several parts over space and time. Instead, we use a simple patch-based approach that requires no manual interaction.

Following the work of [7], we compute distance $D(T, L)$ by comparing the patches from two video segments T and L. Each frame is decomposed into a number of 20×20 patches automatically, then $D(T, L)$ is computed as follows:

$$D(T, L) = \sum_{i=1}^{M} \sum_{s=1}^{S} \min_{r \in R_s} d(t_{is}, q_{ir}) \tag{1}$$

where t_{is} denotes the s-th patch on the template frame i, and q_{ir} denotes the r-th patch on the test frame i. R_s is the corresponding search region of s-th patch. M is the number of frames in a video segment. S is the total number of patches on each frame. $d(\cdot, \cdot)$ refers to the distance between two patches. For simplicity, we ignore the action speed variation between people, and directly correspond the frames from T to L in sequence. One could also apply dynamic programming based approaches to find the frame correspondence and thus alleviate the variation in speed.

3 Cascade Structure

As in most object detection tasks, *e.g.* face detection and car detection, human action detection is a *rare event detection*. Hence, when using a window-scanning approach, it is important to efficiently reject the majority of negative

sub-windows. Viola and Jones [9] proposed a cascade structure in the AdaBoost learning framework. Most of the negative sub-windows are rejected by simpler detectors efficiently, and then more complex detectors are applied to achieve low false positive rates. However, the training of boosted detectors requires a large number of both positive and negative training samples. In the case of human action detection, it is difficult and even impossible to collect such a large training set for any given action. In particular, in our scenario, only one template is available for each action category.

In order to build a cascade structure with only one template, we use the *transferable distance function learning* proposed in [7]. We first define the terminology we will use. The *source training set* denotes the large dataset we already have at hand, for example a standard benchmark dataset (*e.g.* KTH). The *template* denotes the video we use to detect an action in test videos. Note that the source training set does not contain the same action as the template. In this section, we will review the learning of the transferable distance function, then introduce the construction of the cascade structure.

3.1 Transferable Distance Function

This idea of knowledge transfer has been exploited in the context of object recognition and identification [10,11]. In particular, Ferencz *et al.* [10] propose to predict a patch's saliency for object identification by its visual feature called a *hyper-feature*. In human action recognition, we conjecture that there also exists a certain generic relationship between the saliency and the appearance of a patch [7]. For example, "stretched-arm-like" and "stretched-leg-like" patches are more likely to be salient than other patches. This generic relationship is "transferable", and we can employ this knowledge for patch weighting of unknown actions. In [7], we proposed the learning of a transferable distance function, which can extract generic knowledge of patch weighting from previous training sets, *e.g.* benchmark action datasets. When it is applied to unknown actions, the algorithm will look for salient patches and assign them high weights, that are also the parameters of the distance function for matching based recognition.

Given a patch i, the weight assigned to this patch is w_i, and we represent the hyper-feature of this patch as a $|V|$-dimensional vector $\mathbf{f}_i$ based on a codebook approach, where $|V|$ is the codebook size. The j-th element of $\mathbf{f}_i$ is set according to the distance between the feature vector of this patch and the j-th visual word. The feature vector of each patch consists of histogram of oriented gradient (HOG) [12] and patch positions. Please refer to [7] for more details. We assume that $\mathbf{f}_i$ and w_i have a linear relationship via the parameter $\mathbf{P}$:

$$w_i = \langle \mathbf{P} \cdot \mathbf{f}_i \rangle \tag{2}$$

Then we will have $\mathbf{w} = \mathbf{P}^T\mathbf{F}$, where each column of $\mathbf{F}$ denotes the hyper-feature vector of a patch, Each element of $\mathbf{w}$ denotes the weight of a patch. The objective is to learn $\mathbf{P}$ from the source training set. After the training, given any new action video, even if its action does not exist in the source training set, we can compute the weight of each patch of this video by Eqn. 2.

The learning of $\mathbf{P}$ follows the focal learning framework in [13]. The distance function obtained by $\mathbf{w} = \mathbf{P}^T\mathbf{F}$ will satisfy the constraints that the distance between dissimilar actions is larger than similar actions by the margin 1, that is $\langle \mathbf{w}_i \cdot (\mathbf{d}_{ij} - \mathbf{d}_{ik}) \rangle > 1$, $\langle \mathbf{P}^T\mathbf{F}_i \cdot (\mathbf{d}_{ij} - \mathbf{d}_{ik}) \rangle > 1$, where $\mathbf{d}_{ik}$ is the distance vector between the similar action i and k, and $\mathbf{d}_{ij}$ is the distance vector between the dissimilar action i and j. The weights are enforced to be non-negative, $\langle \mathbf{P} \cdot \mathbf{f}_m \rangle \geq 0$. For simplicity, we replace $\mathbf{d}_{ij} - \mathbf{d}_{ik}$ as $\mathbf{x}_{ijk}$. The max-margin optimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{P},\xi} \quad & \frac{1}{2}\|\mathbf{P}\|^2 + C\sum_{ijk}\xi_{ijk} \\ s.t.: \quad & \forall i,j,k: \quad \langle \mathbf{P}^T\mathbf{F}_i \cdot \mathbf{x}_{ijk} \rangle \geq 1 - \xi_{ijk} \\ & \forall m: \quad \langle \mathbf{P} \cdot \mathbf{f}_m \rangle \geq 0 \quad \forall i,j,k: \quad \xi_{ijk} \geq 0 \end{aligned} \tag{3}$$

where ξ_{ijk} is the slack variable and C is the trade-off parameter, similar to those in SVM. See [7] for more details about the solving of this optimization problem.

3.2 Construction of Cascade Structure

A key feature of the cascade structure is to use simpler but efficient detectors at the early stage to reject most negative sub-windows. The learnt distance function provides us a useful tool to obtain such a simple detector. After the learning on the source training set, we are able to compute the weights (*i.e.* saliency) of the patches on any given template action through Eqn. 2, and rank these patches by their saliency. At the early stage of the cascade structure, for the matching task, we can use only a subset of patches with high weights on the template video. For example, we can choose only two patches from each template frame with top-2 high wights at the first stage of the cascade structure. For a template video with 25 frames, only 50 patches are used at the first stage, so it could be very efficiently matched with all the sub-windows in test videos. The majority of negative sub-windows can be discarded after this stage. For the following stages, we can incrementally increase the number of patches utilized in the template video, and all patches will be used at the final stage in order to achieve an accurate matching. At the k-th stage of our cascade structure, distance $D^k(T, L)$ is computed as:

$$D^k(T, L) = \sum_{i=1}^{M} \sum_{s \in E_i^k} w_{is} \min_{r \in R_s} d(t_{is}, q_{ir}) \tag{4}$$

where E_i^k is the set of effective patches on the i-th frame at the k-th stage, and w_{is} is the weight assigned to the template patch t_{is}.

In the cascade structure of [9], the detection and false positive rates of each stage can be controlled using training and validation sets. However, in our scenario, only one template video is available for each action category, and there is no training dataset containing the same action as the template. Here we choose

a rather simple way to control the performance of each stage. The detection threshold of a stage is set so that a certain number of sub-windows with high matching distances will be discarded. The remaining sub-windows will be evaluated by the next stage of the cascade structure. An example of the cascade structure is given in Fig. 1. Note that it is possible that early stages of the cascade structure may have high false negative rates and thus decrease the performance of whole structure. However, the experimental results in Section 4.2 demonstrate our cascade structure achieves similar results to the direct scanning method without using a cascade, which implies the early stages of our cascade structure can reliably keep the true positive sub-windows.

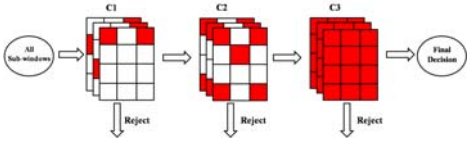


Fig. 1. An example of the cascade structure. The red patches are the effective patches on template frames. At the **C1** stage, the top-2 patches of each frame with high weights are used to match with the input sub-windows. At the **C2** stage, top-5 patches are used for matching. At the final stage, all patches are used.

4 Experiments

We evaluate our method on the cluttered human action dataset collected by Ke *et al.* [6], and a ballet video sequence. We first review the human action datasets, then present the experimental results.

4.1 Datasets

Weizmann Dataset [14]: The Weizmann human action dataset is a standard benchmark human action dataset. It contains 93 sequences of nine actors performing ten different actions. There are about $40-120$ frames for each sequences. This dataset is used as the source training set, so we choose the same *figure-centric* representation as [7]. After computing the motion feature, we crop each frame to 90×60 and put the human figure in the center of the frame.

Cluttered Human Action Dataset [6]: The cluttered human action dataset contains not only cluttered static backgrounds, but also cluttered dynamic backgrounds, such as moving cars and walking people. There are 48 videos containing 110 actions of interest. Each video contains approximately $300-800$ frames with resolution 120×160. Five types of actions are labeled: one-hand waving, two-hand waving, picking-up, pushing an elevator button, and jumping-jacks.

4.2 Experiments on the Cluttered Action Dataset

For human action detection on the cluttered dataset, we first choose one template video for each labeled action event. Except for the action of pushing an elevator button, we use the sequences of the actor *ido* from the Weizmann dataset as templates. For the action of pushing an elevator button, we choose the template provided by Ke *et al.* [6]. Note that this selection of template videos increases the difficulty of the task since the template and test videos are captured under different instructions. All template videos contains only $20-25$ frames, *i.e.* $1-1.5$ complete action cycles.

The figure-centric representation is applied to template videos and all template frames are normalized to 90×60. Representative frames of template videos are shown in Fig. 2. After computing motion features, each frame is decomposed into 40 patches. The size of a patch is 20×20 and the length of the stride is 10.

Fig. 2. Action detection examples on the cluttered action dataset. Representative frames of the template videos and the visualization of learnt weights are shown on the left. The left bottom corner shows the color bar for the visualization. Correct detection examples are shown on the right.

To meet the requirement of the transfer learning scenario, in our experiments, the source training set does not contain the action of the template video. For example, in the experimental step of jumping-jacks action, we remove the action of jumping-jacks from the Weizmann dataset. Then the remaining sequences form the source training set. After the training, we first compute hyper-features of the template video. Then, we can obtain the distance function of the template video through Eqn. 2. The detection of other actions follows the same experimental setup. Note that for the experiment of each action, the source training set does not contain the same action as template. The weights of the distance

function are visualized in Fig. 2. As we can see, the high weights (red patches) are assigned to the *salient* parts, such as the stretched-arm, and bent-back.

After training, we can build the cascade structure based on the learnt distance function. In the experiments, the cascade structure consists of four stages. At the first stage, there are only two effective patches on each template frame. At this stage, the template video is scanned across the test video. Subsequent locations are obtained by shifting the template video either 5 pixels along the x or y axis, or 5 frames along the time axis. Similar to [6], the template videos are matched with the test video under a fixed scale. The speed of this stage is 20 times faster than using all patches on the template video. After the first stage, 90% of the sub-windows are set to be rejected. The second stage has five effective patches on each frame, and 80% of the remaining sub-windows from last stage will be rejected. For the third stage, ten patches on each frame are effective and 80% of the sub-windows will be kept at this stage. All patches on the template video are effective at the final stage. These parameters of the cascade structure are all the same for the experiments of each action.

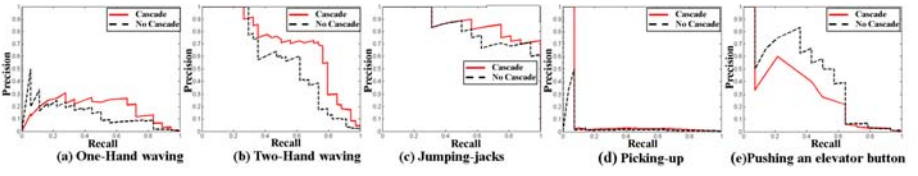


Fig. 3. Precision-Recall curves of the action detection on the cluttered action dataset

Similar to [6], we project the obtained three-dimensional distance map to a one-dimensional vector of score. Only the best detection is kept for each test frame. The Precision-Recall curves are generated by changing the detection threshold, as shown in Fig. 3. Since we choose a different way to scan the template over test videos, our results are not directly comparable with [6]. We admit this dataset is very difficult because of the cluttered background. However, by only using the motion cue, our method is still able to achieve very good performance for jumping-jacks, two-hand waving, and pushing an elevator button. Due to the large intra-class variation of actors performing the picking-up action, our method achieves very low detection rates on this action. One-hand waving is often confused with the two-hand waving and jumping-jacks and thus has a higher false positive rate. Example detections are shown in Fig. 2.

We give an example with more details in Fig. 4 about the detection of jumping-jacks in a video which contains some confusing actions, such as one-hand waving and two-hand waving. It is interesting to note that in the projected matching distance, the confusing actions cause very low matching distances but they are still much higher than the jumping-jacks action.

We also compare the results of using the distance function with and without thee cascade structure. As shown in Fig. 3, except for the action of pushing

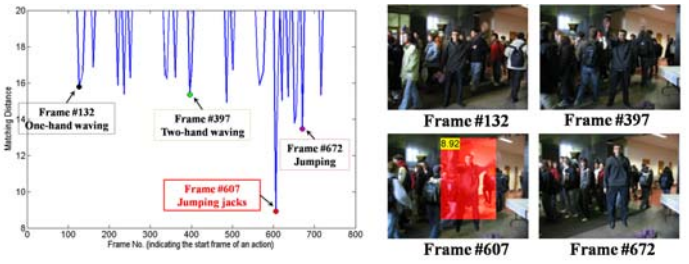


Fig. 4. (a) Projected matching distance of the detection of jumping-jacks. (b) Example detections. The true positives are highlighted in Frame #607, where the left corner is the matching distance. The rest frames are all true negatives.

an elevator button, our cascade structure achieves better accuracy. Moreover, the cascade structure is much more efficient. The methods are implemented in Matlab/MEX. With a 2.40GHz Intel processor, to scan a template video with 25 frames over a test video with 800 frames, the cascade structure only takes 30.3 seconds, but it takes 348.2 seconds without using the cascade. There is an order of magnitude improvement in efficiency by using the cascade structure.

4.3 Experiment on the Ballet Video

We apply our method to detect "spin" actions in a ballet video. Although this ballet video is very "clean", it contains more complex actions and two actors are performing the same actions in each frame. In addition, the actress wears a skirt and the appearance is very different to the template, which might cause difficulty for shape-based methods (*e.g.* [6]).

The Weizmann dataset serves as the source training set. The learnt weights on the template video are visualized in Fig. 5(a). Note that the actions in the

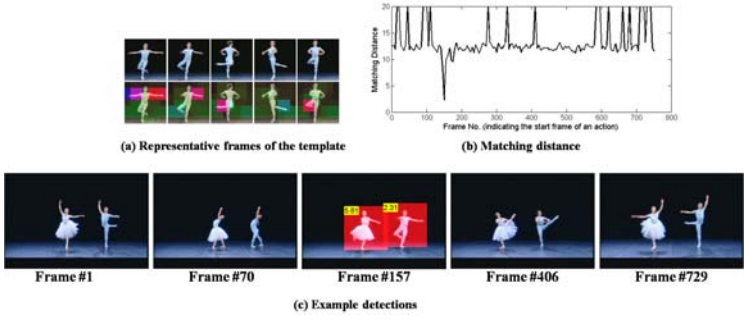


Fig. 5. (a) Representative frames of the template videos, and the visualization of learnt weights. (b) Projected matching distance. (c) Example detections. The true positives are highlighted in Frame #157, and the rest frames are all true negatives.

Weizmann dataset are distinctly different from the "spin" action of ballet. Our transferable distance function is still able to assign high weights onto the salient parts such as the stretched-arms and legs. After training, we scan the template over the test video using the cascade structure. The matching distances of correct detections for the actor and actress are 2.31 and 5.81 respectively. Although the matching distance for the actress is higher than the actor because of the clothing, these distances are still much lower than any other portion of the video.

5 Conclusion

In this paper, we have presented an efficient human action detection approach using only one template video. We have developed a histogram representation of the four-channel motion descriptors [8], which can be efficiently computed using integral images. Based on the learning of a transferable distance function [7], a cascade structure has been proposed. Experimental results show that our cascade structure achieves reliable detection results and improves the efficiency of the patch based action detection method significantly.

References

1. Moeslund, T., Hilton, A., Kruger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 103(2-3), 90–126 (2006)
2. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. IJCV 79(3), 299–318 (2008)
3. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: ICCV, vol. 1, pp. 166–173 (2005)
4. Laptev, I., Pérez, P.: Retrieving actions in movies. In: ICCV (2007)
5. Shechtman, E., Irani, M.: Space-time behavior based correlation. In: CVPR (2005)
6. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: ICCV (2007)
7. Yang, W., Wang, Y., Mori, G.: Human action recognition from a single clip per action. In: The 2nd International Workshop on Machine Learning for Vision-based Motion Analysis (2009)
8. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, pp. 726–733 (2003)
9. Viola, P., Jones, M.: Robust real-time face detection. In: IJCV (2004)
10. Ferencz, A., Learned-Miller, E., Malik, J.: Learning to locate informative features for visual identification. In: IJCV (2006)
11. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. PAMI 28(4), 594–611 (2006)
12. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: CVPR (2005)
13. Frome, A., Singer, Y., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS, vol. 19. MIT Press, Cambridge (2007)
14. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV (2005)

Crease Detection on Noisy Meshes via Probabilistic Scale Selection

Tao Luo, Huai-Yu Wu, and Hongbin Zha

Key Laboratory of Machine Perception (Ministry of Education),
Peking University, China
{luotao,wuhy,zha}@cis.pku.edu.cn

Abstract. Motivated by multi-scale edge detection in images, a novel multi-scale approach is presented to detect creases on 3D meshes. In this paper, we propose a probabilistic method to select local scales in the discrete 3D scale space. The likelihood function of local scale at each vertex is defined based on the minimum description length (MDL) principle. By introducing some prior knowledge, the optimal local scales are selected using Bayes rule. Therefore, the distribution of selected local scales is piecewise constant and discontinuity adaptive. The discrete 3D multi-scale representation of a given mesh can be constructed using an anisotropic diffusion method. With the selected scales, creases are traced by connecting the curvature extrema points detected on the mesh edges. Experimental results show that geometrically salient creases are well detected on noisy meshes using our method.

Keywords: minimum description length, Markov Random Field, multi-scale, crease detection.

1 Introduction

Feature line detection is an essential step in geometry processing, which has numerous applications in shape recognition, mesh simplification, mesh segmentation, non-photorealistic rendering, etc. Several methods for feature line detection have been proposed on polygonal and point-sampled surfaces [1] [2] [3] [4] [5] [6]. In different applications there exist a variety of feature lines, such as creases [1], contours and suggestive contours [7], apparent ridges [8], demarcating curves [9], etc. In this paper, we propose a method for detecting creases on triangular meshes. Creases are defined as the loci of points where the principal curvatures take extrema along their corresponding principal directions [1]. Specifically, creases include ridges and valleys, both of which are view-independent feature lines.

The general framework of crease detection in previous methods [1] [5] [6] includes estimating curvatures and curvature derivatives, tracing creases on mesh edges, and post-processing of creases. These methods put emphasis on reliable estimation of curvatures and curvature derivatives, which is crucial to detect creases accurately. Therefore, they are well experimented on noiseless meshes and achieve satisfying results. However, for the triangular meshes reconstructed

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 427–437, 2010.

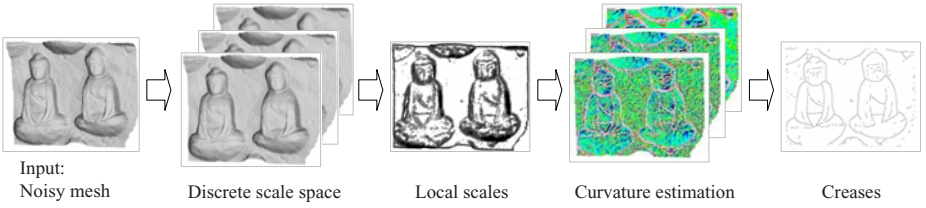


Fig. 1. Flowchart of our method

from raw scanning data, noise is inevitable due to the accuracy of laser scanner, perturbation and variation of reflectance property of the objects. Thus, previous methods may detect many redundant lines, which are hard to be distinguished from real creases.

To address this problem, we propose an approach to combining curvature information at multiple scales so that the redundant lines are eliminated to a large extent. The pipeline of our method is shown in Fig. 1. Given a noisy mesh as input, we first generate its discrete 3D multi-scale representation using an anisotropic diffusion method, which preserves the geometric features. Then, we implement a probabilistic method for local scale selection at each vertex to combine information in the discrete scale space. Based on the minimum description length (MDL) principle, a likelihood is defined at each vertex given a local scale. By introducing some prior knowledge about the distribution of local scales, we compute the posterior probabilities using Bayes rule with a prior Markov Random Field (MRF) model. The optimal local scale at each vertex is obtained by finding the scale associated with the maximum posterior probability, which makes the distribution of local scales piecewise constant and discontinuity adaptive. Finally, with the curvatures and curvature derivatives estimated at the selected local scales, creases are traced by connecting the curvature extrema points detected on the mesh edges.

Several methods for multi-scale feature extraction have been developed in [3] [10] [11]. In [3], a new technique for extracting line-type features on point-sampled geometry is presented. The key idea is to estimate the surface variation at multiple scales using the size of neighborhood as a discrete scale parameter. Feature weights are calculated to combine information at multiple scales. However, the thresholds need to be adjusted manually. In [10] [11], scale-dependent geometric features on triangular meshes are detected in the scale space of a 2D representation encoded by the surface normals. However, the required parametrization is time-consuming and induces distortion of the surface area. Moreover, the geometric features are blurred by Gaussian kernel convolution with 2D normal map. In [11], Novatnack et al. combine the shape features for each point by taking the maximum value across all scales weighted by the scale level in which the maximum detector response occurs. In [10], analogous to the automatic scale selection method proposed in [12], the scale is determined where the normalized feature response is maximized across a set of discrete scales. Recently, a multi-scale surface representation based on point samples is presented in [13], where local weighted least squares fitting is applied

to approximate shapes with different levels of smoothness. However, as the least squares filter with symmetric Gaussian kernel is isotropic, features can not be preserved in the scale space. To preserve the geometric features in the discrete scale space, a discrete 3D multi-scale representation of a triangular mesh is introduced in this paper based on the anisotropic diffusion method in Section 2. In Section 3, in order to combine information at multiple scales, we propose a probabilistic method for automatic scale selection so that the distribution of local scales is piecewise constant and discontinuity adaptive.

2 Multi-scale Representation of a Triangular Mesh

Motivated by edge detection in images, a multi-scale representation of a mesh is proposed for crease detection in this section. We introduce an anisotropic diffusion method in [14] to generate the discrete 3D scale space, which is efficient and feature-preserving.

2.1 Anisotropic Diffusion

As demonstrated in [15], anisotropic diffusion has good performance to generate multi-scale representation of images in 2D image processing. We implement an extension of anisotropic diffusion in images to generate the discrete scale space of a 3D mesh.

On a 3D mesh, the vertex-based anisotropic diffusion equation is expressed as,

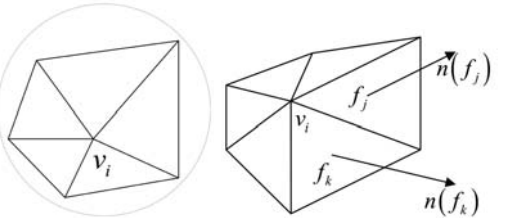
$$v_t = div(g(||F||)\nabla v) \ , \tag{1}$$

where F denotes the estimation of normal variations on the mesh, div the divergence operator and ∇ the gradient operator, respectively. The conduction coefficient $g(||F||)$ is a function of the estimated normal variations. The value of the function varies spatially in such a way to encourage smoothing in flat regions in preference to smoothing around the sharp edges, which leaves the geometric features preserved much better.

The face normal variation depicted in Fig. 2(b) is the key difference of our approach from [16]. When the mesh is tessellated irregularly, it is more reliable than vertex gradient magnitude therein. For each non-boundary vertex v_i on a mesh, $||F_i||$ is defined as,

$$||F_i|| = \frac{1}{d_i} \sum_{f_{j,k} \in v_i^*} \arccos < n(f_j), n(f_k) > \ , \tag{2}$$

where v_i^* is the one-ring neighborhood of the vertex v_i as shown in Fig. 2(a), f_j and f_k are adjacent faces in the one-ring neighborhood, d_i is the degree of v_i, $n(f_j)$ and $n(f_k)$ are the face normals.



(a) (b)

Fig. 2. (a) One-ring neighborhood of a vertex. (b) Illustration of normal variation.

To clamp the conduction coefficient $g(||F||)$ to the range $[0, 1]$, the following function is used,

$$g(x) = \frac{1}{1 + x^2/c^2} \ , \tag{3}$$

where c is a constant.

The vertex gradient operator is defined as,

$$\nabla v_i = \frac{v_j}{\sqrt{d_j}} - \frac{v_i}{\sqrt{d_i}}, \ v_j \in v_i^* \ . \tag{4}$$

With the vertex gradient (4) approximated by the finite difference of the vertex positions, the anisotropic diffusion (1) can be explicitly expressed as,

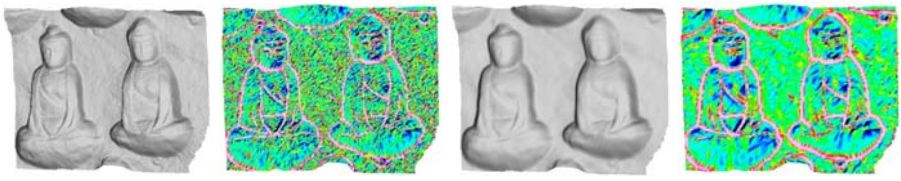
$$v_i \leftarrow v_i + \sum_{v_j \in v_i^*} \frac{1}{\sqrt{d_i}} \left(\frac{v_j}{\sqrt{d_j}} - \frac{v_i}{\sqrt{d_i}}\right)\left(g(||F_i||) + g(||F_j||)\right) \ , \tag{5}$$

where v_j is an adjacent vertex of v_i, d_i and d_j are the degrees of v_i and v_j respectively.

We implement the above anisotropic diffusion (5) on a noisy mesh iteratively. The discrete 3D scale space representation is constructed by a series of 3D shapes at different levels of smoothness. The number of iterations is used as the scale parameter. During this process, the correspondence of the mesh vertices between different scales is kept, which preserves the topological connectivity of meshes. Figure 3 shows a 3D model on which the anisotropic diffusion is applied 10 iterations. We can see that the geometric features are well preserved while the flat regions are smoothed. The discrete 3D multi-representation of the triangular mesh is illustrated in Fig. 4.

3 Probabilistic Estimation of Local Scale

With the aforementioned anisotropic diffusion method, we can efficiently generate the discrete 3D multi-scale representation of a triangular mesh. To automatically select a local scale for each vertex on the mesh, we present a probabilistic method based on the minimum description length (MDL) principle. By introducing prior knowledge, the local scales are selected using Bayes rule with a prior MRF model.



(a) (b) (c) (d)

Fig. 3. (a) A stone Buddha model. (c) The smoothed model after 10 iterations. (b) and (d) are curvatures estimated on (a) and (c) respectively.

3.1 Minimum Description Length

The basic idea of minimum description length (MDL) principle [17] states that, given a set of hypotheses H and data set D, we should try to find the hypothesis or combination of hypotheses in H that compress D most. Thus, the problem of selecting local scales can be regarded as finding an optimal distribution of local scales in the discrete 3D scale space that minimizes the description length of a given triangular mesh.

A discrete 3D multi-scale representation of triangular mesh M_0 can be obtained using the anisotropic diffusion in Sec. 2. At each scale t, the original mesh M_0 can be decomposed into a smoothed mesh M_t and a residual $\varepsilon_t = M_0 - M_t$. The description length is expressed as,

$$L(M_0 \mid t) = L(M_t) + L(\varepsilon_t) \ , \tag{6}$$

where $L(M_0 \mid t)$ denotes the description length of the original mesh at scale t, $L(M_t)$ and $L(\varepsilon_t)$ denote the description lengths of the smoothed mesh and the residual respectively.

The description length can be obtained by computing the two decomposed items individually. First, we consider the description length of the smoothed mesh in Equation (6). The larger the scale is, the more the original mesh is smoothed. Therefore, the description length $L(M_t)$ is inversely proportional to the scale t, that is, $L(M_t) \propto 1/t$. Then, the description length of the residual is related to noise on the original mesh. We define the residual at each scale as the distance between the positions of corresponding vertices, that is, $\varepsilon_t^2 = ||v_0 - v_t||^2$. Thus, the description length is proportional to the residual, that is, $L(\varepsilon_t) \propto \varepsilon_t^2$.

Given the local scale $t(v)$ at a vertex v of M_0, the description length is defined as,

$$L(M_0(v) \mid t(v)) = \frac{\lambda}{t(v)} + \varepsilon_{t(v)}^2(v) \ , \tag{7}$$

where λ is a constant.

Therefore, the likelihood at each vertex given a local scale t_k is defined as,

$$\hat{p}_k(v) = P(M_0(v) \mid t(v) = t_k) = \frac{1}{Z_v} e^{-(\frac{\lambda}{t_k} + \varepsilon_{t_k}^2(v))} \ , \tag{8}$$

where Z_v is a normalizing constant.

The local scale can be selected at each vertex by maximizing the likelihood in Equation (8). However, the distribution of selected local scales is incoherent spatially, as shown in Fig. 5(a).

3.2 Scale Selection with a Prior MRF Model

In order to find the optimal local scales on the triangular mesh, we introduce some prior knowledge about the distribution of local scales to compute a Bayesian estimator. The local scale field is assumed to be Markovian in the sense that the probabilistic dependencies are restricted to a neighborhood of each vertex on the triangular mesh. Thus, a prior Markov Random Field (MRF) model is used to find a piecewise constant and discontinuity-adaptive distribution of local scales.

The constraints about prior knowledge are expressed by means of a Gibbs distribution,

$$P_t(t) = \frac{1}{Z_t} e^{[-\beta \sum_{<v,u>} V(t(v),t(u))]} , \tag{9}$$

where $< v, u >$ denotes a pair of adjacent vertices in one-ring neighborhood, Z_t is a normalizing constant, β is a positive parameter, and $V(t(v), t(u))$ is a potential function.

To constrain the local scale field to be piecewise constant, the generalized Ising model [18] is used, whose potentials are given by,

$$V(t(v), t(u)) = \begin{cases} -1, & t(v) = t(u) \\ 1, & t(v) \neq t(u) \end{cases} . \tag{10}$$

Moreover, the local scale field is discontinuous when the geometric shape changes abruptly. Therefore, we define a function to describe this property, which is related to the variation between normals of adjacent vertices. The function $\alpha(v, u)$ is defined as,

$$\alpha(v, u) = \frac{1}{1 + \frac{\theta^2_{(v,u)}}{\eta}} , \tag{11}$$

where η is a constant, $\theta_{(v,u)}$ denotes the variation between the normals of vertex v and vertex u.

With the above prior distribution and the likelihood (8) defined in Sec. 3.1, the posterior distribution is computed according to Bayes rule,

$$P(t \mid M_0) = \frac{P_t(t) P(M_0 \mid t)}{P(M_0)} , \tag{12}$$

where $P(M_0)$ is a normalizing constant, and the likelihood $P(M_0 \mid t)$ is defined as,

$$P(M_0 \mid t) = \prod_v P(M_0(v) \mid t(v)) . \tag{13}$$

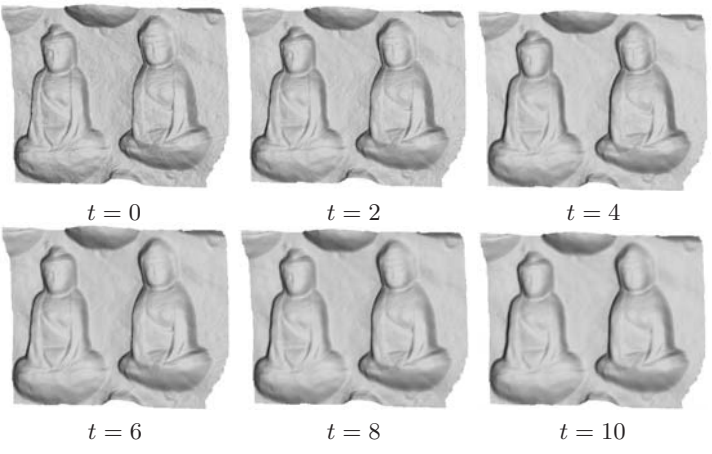


Fig. 4. The discrete 3D multi-scale representation of the Buddha model

The posterior marginal distribution can be approximated by minimizing an energy function [19], which is expressed as,

$$U(p) = \sum_{v} |p(v) - \hat{p}(v)|^2 + \sum_{<v,u>} \alpha(v,u)|p(v) - p(u)|^2 \,, \tag{14}$$

where $p(v)$ is the posterior marginal probabilities at vertex v, $\hat{p}(v)$ is the likelihood, and $\alpha(v,u)$ describes the discontinuity between vertices pair $< v,u >$ as defined in (11). This energy function indicates that the posterior marginal distributions should be similar to the likelihood of the observations and should keep piecewise constant and discontinuity adaptive across the mesh vertices.

The minimization of the energy function $U(p)$ is equivalent to the solution of decoupled systems of linear equations. The optimal local scale is selected as the scale associated with the maximum posterior marginal probability at each vertex, which is given by,

$$\begin{aligned} t^*(v) &= t_{k_{\max}} \,, \\ k_{\max} &= \underset{k}{\operatorname{argmax}}\, p_v(k) \,. \end{aligned} \tag{15}$$

In this way, the optimal local scale is selected at each vertex on the triangular mesh. The distribution of local scales is piecewise constant and discontinuity adaptive due to the introducing of prior knowledge, as shown in Fig. 5(b).

4 Multi-scale Approach to Crease Detection

After generating the discrete 3D multi-scale representation of a noisy mesh in Sec. 2, we can combine information at different scales selected in Sec. 3 to detect creases. Creases, including ridges and valleys, are defined as the loci of points

(a) (b)

Fig. 5. (a) Local scales with maximum likelihood estimator. (b) Local scales with Maximizer of the Posterior Marginals.

where the principal curvatures take extrema along their corresponding principal directions [1].

Similar to the general framework of crease detection in [1], our multi-scale approach also includes estimating curvatures and curvature derivatives, and tracing creases on mesh edges. Nevertheless, we combine information at multiple scales when estimating curvatures and curvature derivatives. The method proposed in [20] is employed to estimate the curvatures and curvature derivatives on the mesh models across all the scales in the discrete 3D scale space. It is a finite difference approach which can be seen as an extension of a common method for estimating per-vertex normals. For more details, we refer to [20]. With the probabilistic selection of local scale in Sec. 3, the curvature information is available at selected local scales. Finally, creases are traced by connecting crease points detected at the zero-crossing points of first-order curvature derivatives on mesh edges. The creases detected using multi-scale approach are shown in Fig. 6(d)and the creases detected directly (e.g. [20]) in Fig. 6(b).

5 Experiments

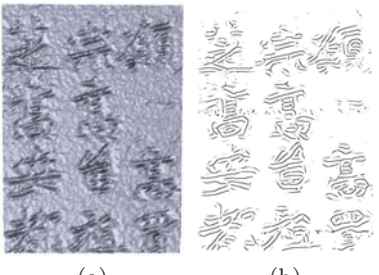
We apply our method on a stone Buddha model scanned by VIVID 910, with $36,503$ vertices and $72,227$ triangles. As mentioned above, noise exists on the model due to the accuracy of the scanner and the reflectance of stone material, which can be seen in the curvature map shown in Fig. 3(b). Therefore, if the previous method (e.g. [20]) is directly applied on the Buddha model, redundant detection occurs due to the inherent noise in the raw scanning data, as shown in Fig. 6(b).

In contrast to previous methods, we first generate the discrete 3D multi-scale representation of the given Buddha model using the anisotropic diffusion method described in Sec. 2. In Fig. 4, a series of 3D models at different levels of smoothness are plotted, where t denotes the number of iterations. The discrete 3D scale space is formed by these models, where the geometric features are preserved across the scales as shown in Fig. 3(b) and Fig. 3(d). Then, the probabilistic method in Sec. 3 is employed to determine the local scale at each vertex. By introducing some prior knowledge about the distribution of local scales on the triangular mesh, we estimate the scale at each vertex using Bayes rule with a

(a) (b) (c) (d)

Fig. 6. (a) A stone Buddha model. (b) Creases detected directly. (c) Creases at a single scale $t = 2$. (d) Creases detected using our approach.

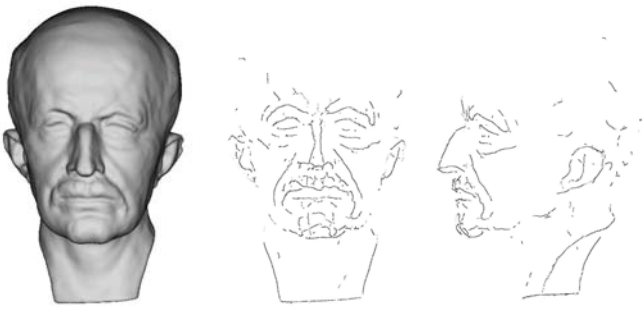
prior Markov Random Field model. As demonstrated in Fig. 5(b), the distribution of local scales is piecewise constant and discontinuity adaptive, compared with the local scales selected by the maximum likelihood estimator shown in Fig. 5(a). Finally, with the selected local scales, we combine curvature information on all models in the discrete scale space. As described in Sec. 4, creases are traced by connecting creases points detected at the zero-crossing points of first-order curvature derivatives on mesh edges. It can be seen that the redundant lines are eliminated evidently by combining multiple scales in Fig. 6(d), in comparison with the creases (valleys only) detected directly in Fig. 6(b) and the creases (valleys only) detected at a single scale in Fig. 6(c). Compared with the method in [14], the scales are selected automatically without interactive adjustment of thresholding parameters therein, while the detected result is comparable using our method.



(a) (b)

Fig. 7. (a) A stele model. (b) Non-photorealistic rendering.

Our method is also implemented on a stele model shown in Fig. 7(a), which consists of 19,991 vertices and 39,416 triangles. By combining information at multiple scales, both the ridges and valleys are detected on the mesh. They are rendered with a non-photorealistic style in Fig. 7(b). Similar to the method proposed in [9], our method can also be used as assistance for archeologists to draw "line drawing". Furthermore, the results demonstrate that our multi-scale method is robust to noise, which can be applied to raw scanning data directly.



(a) (b) (c)

Fig. 8. (a) The Max Planck model. (b) and (c) show two different views of creases detected using our approach.

For comparison with the methods proposed in [10] [11], we apply our multi-scale approach on the Max Plunck model in Fig. 8(a). The detected creases (ridges only) are shown at two different views in Fig. 8(b) and Fig. 8(c), which are comparable with the results in [10] [11]. Moreover, our method is more efficient in computation without the time-consuming parametrization required in [10] [11].

6 Conclusions

In this paper, we present a probabilistic approach to selecting local scales in a discrete 3D scale space. First, the discrete 3D multi-scale representation of a given triangular mesh can be efficiently constructed using an anisotropic diffusion method. Then, a probabilistic method for local scale selection is proposed based on the minimum description length (MDL) principle. With the local scales selected using Bayes rule, we detect creases by combining information at multiple scales. Therefore, our method can evidently reduce redundant lines and well preserve geometric features in comparison with previous methods.

Our method improves the results of crease detection significantly on raw scanning data. Furthermore, our multi-scale strategy with probabilistic scale selection can also be applied to detect other types of feature lines . In archeology, the creases detected using our method can be used as assistance for archeologists to draw "line drawing". In the future, more post-processing steps may be considered to make the results visually artistic.

Acknowledgments. This work was supported in part by NKBPRC (No. 2004C B318000), NHTRDP 863 Grant No. 2009AA01Z329 and NHTRDP 863 Grant No. 2007AA01Z336.

References

1. Ohtake, Y., Belyaev, A., Seidel, H.P.: Ridge-valley lines on meshes via implicit surface fitting. ACM Trans. on Graphics 23(3), 609–612 (2004)
2. Page, D.L., Sun, Y., Koschan, A., Paik, J., Abidi, M.: Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. Journal of Graphical Models 64, 199–229 (2002)
3. Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled models. Computer Graphics Forum (Eurographics 2003) 22(3), 281–289 (2003)
4. Watanabe, K., Belyaev, A.G.: Detection of salient curvature features on polygonal surfaces. Computer Graphics Forum (Eurographics 2001) 20(3), 385–392 (2001)
5. Yoshizawa, S., Belyaev, A., Seidel, H.P.: Fast and robust detection of crest lines on meshes. In: ACM Symposium on Solid and Physical Modeling, pp. 227–232 (2005)
6. Yoshizawa, S., Belyaev, A., Yokota, H., Seidel, H.P.: Fast and faithful geometric algorithm for detecting crest lines on meshes. In: Pacific Graphics, pp. 231–237 (2007)
7. Decarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A.: Suggestive contours for conveying shape. ACM Trans. on Graphics 22(3), 848–855 (2003)
8. Judd, T., Durand, F., Adelson, E.: Apparent ridges for line drawing. ACM Trans. on Graphics 22(3), 19:1–19:7 (2007)
9. Kolomenkin, M., Shimshoni, I., Tal, A.: Demarcating curves for shape illustration. ACM Trans. on Graphics (SIGGRAPH Asia) 27(4), 157:1–157:9 (2008)
10. Novatnack, J., Nishino, K., Shokoufandeh, A.: Extracting 3d shape features in discrete scale-space. In: Int. Symposium on 3D Data Processing, Visualization and Transmission (2006)
11. Novatnack, J., Nishino, K.: Scale-dependent 3d geometric features. In: Proc. Int. Conference on Computer Vision (2007)
12. Lindeberg, T.: Feature detection with automatic scale selection. Int. Journal of Computer Vision 30, 77–116 (1998)
13. Pauly, M., Kobbelt, L.P., Gross, M.: Point-based multi-scale surface representation. ACM Trans. on Graphics 25(2), 177–193 (2006)
14. Luo, T., Zha, H.: Multi-scale creases detection on noisy meshes. In: Proc. 2008 IEEE Int. Conference on Image Processing, pp. 1960–1963 (2008)
15. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Trans. on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
16. Zhang, Y., Hamza, A.B.: Vertex-based diffusion for 3-d mesh denoising. IEEE Trans. on Image Processing 16(4), 1036–1045 (2007)
17. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of Statistics 11, 416–431 (1983)
18. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions and the bayesian restoration of images. IEEE Trans. on Pattern Analysis and Machine Intelligence 6, 721–741 (1984)
19. Marroquin, J., Velasco, F., Rivera, M., Nakamura, M.: Gauss-markov measure field models for low-level vision. IEEE Trans. on Pattern Analysis and Machine Intelligence 23, 337–348 (2001)
20. Rusinkiewicz, S.: Estimating curvatures and their derivatives on triangle meshes. In: Int. Symposium on 3D Data Processing, Visualization and Transmission (2004)

Improved Uncalibrated View Synthesis by Extended Positioning of Virtual Cameras and Image Quality Optimization

Fabian Gigengack and Xiaoyi Jiang

Department of Mathematics and Computer Science, University of Münster,
Einsteinstraße 62, 48149 Münster, Germany
{fabian.gigengack,xjiang}@uni-muenster.de
http://cvpr.uni-muenster.de

Abstract. Although there exist numerous view synthesis procedures, they are all restricted to certain special cases. Some procedures for instance can only handle a calibrated camera set while others are limited to interpolation between the reference views. In this paper we will present a fully automated uncalibrated view synthesis procedure. It allows an arbitrary camera placement in 3-D space on the basis of only two input images with a natural camera orientation. Natural camera orientation means that the focus of the virtual camera is intrinsically given by the geodesic which again is determined by the reference views. The presented procedure extends an existing view synthesis algorithm that allows only a camera placement on the 1-D geodesic (in the case of two reference views). The extensions are an additional camera placement along and orthogonally to the line of sight. The image quality of the virtual views will also be enhanced by utilizing the image information of both reference views.

Keywords: Uncalibrated View Synthesis, Relative Affine Structure.

1 Introduction

The term *view synthesis* denotes the generation of virtual views of a scene based on a few reference pictures of this scene. This paper concentrates on the case of exact two reference pictures. It will be shown that such two reference views are sufficient for a flexible camera placement in 3-D space by means of an approach for *uncalibrated* view synthesis *without* the need of any *user interaction*.

Motivation. View synthesis techniques can be applied to various fields. In the film industry for instance, it is widely used to create the 'bullet time' effect which became famous with the 1999 movie 'The Matrix'. By using multiple cameras and view synthesis algorithms, a virtual camera movement around a scene which seems to be frozen in time is simulated.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 438–447, 2010.

Another field of application is sports. View synthesis techniques were already used during the *European Soccer Championship 2008*, where important situations were analyzed with a computer program called *LiberoVision*[1]. Decisions about offsides for example can be supported by placing a virtual camera on a level with the affected players.

Another example of use in the range of sports could be goalkeeper practice in handball. A penalty shot can possibly determine whether a game will be won or lost. To prepare the goalkeeper for opposing players, the video material of past penalty shots of these players could be analyzed. A view synthesis algorithm together with cameras next to the goal allow to simulate the point of view of the goalkeeper for a realistic training.

Related Work. The roots of *view synthesis* can be found for example in the work of S. Seitz and C. Dyer [1] from 1996. They present a method called *view morphing* which is a combination of *view interpolation* and *image morphing*. Restrictions of this technique are that some amount of user interaction is necessary to mark feature points and only interpolated views can be generated.

A technique reminding of the already mentioned 'bullet time' effect was introduced in 2004 by Zitnick et al. [2]. View interpolation is performed between sparse synchronized cameras arranged along a one dimensional arc. A drawback of this technique is the limitation to interpolation only and the need of a calibrated camera system.

The 2007 paper of Criminisi et al. [3] addresses the issue view synthesis for teleconferencing and provides satisfying results. It generates new views with the constraint of a pair of rectified video streams.

Techniques without the need of a calibrated camera system have been proposed for example by S. Avidan and A. Shashua [4] in 1998. They established the so-called *trilinear tensor* for three reference images which can be used to describe the spatial relationships without prior calibration. The technique also allows an extrapolation of the views but has the general drawback of requiring some user interaction.

An automated procedure was presented in 2007 by A. Fusiello et al. [5,6] with focus on a novel positioning method of virtual cameras. The possible new camera positions for two reference views lie on a curve (1-D manifold) through the two reference cameras, allowing interpolation as well as extrapolation.

Contribution of the Paper. Based on the algorithm in [6], which will be introduced shortly in Section 2, we have developed methods that extend the camera placement from a 1-D manifold to arbitrary positions in 3-D space. These extensions are described in Section 3. Apart from the flexible camera placement the key benefits are the abandonment of a manual camera calibration and any kind of user interaction. Further in Section 4 a method for image quality improvement will be discussed based on the utilization of the image information of both reference images. Some experimental results will be discussed in Section 5 followed by a general conclusion in Section 6.

[1] http://www.liberovision.com/

2 Uncalibrated View Synthesis

The view synthesis algorithm as presented by A. Fusiello in [6] has a pipeline structure [5]. This has the general advantage that every single step of the pipeline can be realized by a suitable method. The pipeline consists of the following steps:

1. Parameter estimation *(keypoint detection, outlier elimination, rectification of the input images)*
2. Stereo analysis *(stereo matching, de-rectification of the disparity maps, calculation of the relative affine structure)*
3. Warping of the input images

In this work we will only discuss the third part of the pipeline, the warping of the input images. A detailed disquisition on the other two parts can be found in [5].

An important part of the warping procedure is represented by the so-called *uncalibrated rigid transformation matrices* which control the virtual camera movement between two views as we will see later on.

Definition 1. *We define the* uncalibrated rigid transformation matrix *as a combination of the homography at infinity $H_{\infty 12}$ between image* 1 *and image* 2 *and the epipole e_2 of the second image*

$$D_{12} := \begin{bmatrix} H_{\infty 12} & e_2 \\ 0 & 1 \end{bmatrix}. \tag{1}$$

As shown in [6], we can identify the uncalibrated rigid transformation matrices with the well understood transformation of the *special euclidean group* $SE(3, \mathbb{R})$. Since $SE(3, \mathbb{R})$ is a Lie group it is possible to compute continuously varying interpolated and extrapolated virtual camera positions by scaling and combining transformation matrices.

The handling of the special euclidean group $SE(3, \mathbb{R})$ is realized through the following definitions from [6]:

Definition 2. *Given $G \in SE(3, \mathbb{R})$, the* scalar multiple *of G is defined as*

$$G^t := \exp(t \log(G)),\ t \in \mathbb{R} \tag{2}$$

As G in the definition is a 4×4 matrix the function identifier exp and log describe the matrix exponential and matrix logarithm. Matrices of $SE(3, \mathbb{R})$ can also be combined by a *linear combination*.

Definition 3. *Let G_1, $G_2 \in SE(3, \mathbb{R})$. The* linear combination *of G_1 and G_2 is defined as*

$$(G_1^u) \oplus (G_2^v) := \exp(u \log(G_1) + v \log(G_2)),\ u, v \in \mathbb{R} \tag{3}$$

Two linear independent uncalibrated rigid transformation matrices span a two dimensional manifold in $SE(3, \mathbb{R})$.

With these tools applied to uncalibrated rigid transformation matrices, a point m_1 from the first image can be transformed to its corresponding point m_2 in the second image with the following formula:

$$m_2 \cong [I|0]D_{12}\begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix}. \tag{4}$$

The symbol $\cong$ means 'equal up to a scale factor' and I denotes the identity matrix. The term γ_1 represents the *relative affine structure* of m_1. The theory of the relative affine structure was introduced by A. Shashua and N. Navab in [7] and [8]. It is used to describe the depth information of the scene relative to the corresponding reference view and is gained in our case from the disparity maps, which can be computed from the rectified reference views.

In order to perform the coordinate transformation in Equation (4) the relative affine structure γ_1 and the uncalibrated rigid transformation matrix D_{12} are needed. We will now briefly introduce the computation of the matrix D_{12} while further information on the derivation of the relative affine structure can be found in [6]. Referring to Definition 1 we need to compute the homography at infinity $H_{\infty 12}$ and the epipole e_2 of the second image. The homography at infinity can be easily obtained from the precomputed homographies of the first (H_1) and second (H_2) reference image out of the rectification step (see [9]):

$$H_{\infty 12} = H_2^{-1} H_1. \tag{5}$$

The next step is to compute the epipole e_2. As the fundamental matrix F is known from the parameter estimation step in the pipeline (see [5] for further information) and $e_2^T F = 0$ holds, the epipole e_2 is given as the left zero vector of F.

With this, the matrix D_{12} is fully defined. For $\nu \in \mathbb{R}$ and a point m_1 in the first reference image the new position m_ν of m_1 in the virtual image defined by $D_{1\nu} := D_{12}^{\nu}$ can be calculated according to Equation (4) via

$$m_\nu \cong [I|0]D_{1\nu}\begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix}. \tag{6}$$

3 Extended Positioning of Virtual Cameras

The method described in Section 2 is already a powerful tool for generating virtual views of a scene. In order to extend the potential of this algorithm we will present some refinements that augment the variety of camera positioning from a 1-D curve (geodesic) to an arbitrary point in 3-D space. In Section 3.1 we introduce a procedure that allows additional to the camera positions on the geodesic a camera movement along the line of sight and in Section 3.2 this will be extended to a camera movement orthogonally to the geodesic and the line of sight.

3.1 Camera Movement along the Line of Sight

For the purpose of a more flexible virtual camera placement we will discuss an extended camera movement along the line of sight[2] in this section.

The principle can be seen in Figure 1. L and R denote the two real cameras on the left and on the right. Between them is a virtual camera V which lies on the geodesic describing an interpolation. The observed scene is indicated by the big ball. The dashed line represents the extended camera positions along the line of sight for V. Remember that V is chosen randomly. The two small balls on the dashed line indicate possible new camera placements.

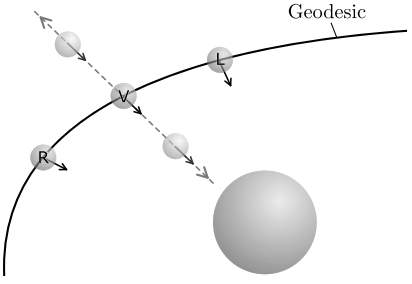


Fig. 1. Camera movement along the line of sight

Analyzing only the camera movement relative to V along the line of sight it can be observed that the new position and V are singular views. This is due to the fact that the camera center of the front camera (we will use the intuitive terms 'forward and backward' as a synonym for 'along the line of sight') lies in the field of view of the rear camera. Accordingly, the images cannot be rectified. The theory described in Section 2 can nevertheless be applied to this case.

The proceeding. Although any camera on the geodesic could be transformed along the line of sight we treat only the case of the first reference camera for simplicity and without loss of generality.

The task is to find a transformation matrix $D_{1\vartheta}$ that describes the camera movement to the front or back where ϑ symbolizes the new camera position. Then we can transform the first reference image according to Equation (6).

We need to determine the homography $H_{\infty 1\vartheta}$ and the epipole $e_{1\vartheta}$. The notation $e_{1\vartheta}$ means that we examine the epipole of the new position ϑ related to the first camera. Accordingly, $H_{\infty 1\vartheta}$ describes the homography between the first camera and the position ϑ. The homography $H_{\infty 1\vartheta}$ is the identity matrix because there is no rotation between the views. An intuitive guess reveals that the epipole $e_{1\vartheta}$ lies in the image center. Thus it is of the form $e_{1\vartheta} = (center_x, center_y, 1)^T$, where $center_x$ and $center_y$ describe the coordinates of the image center. The epipole is scaled to unit length. To control the extent of the displacement the epipole can be rescaled where applicable.

Let s be a scaling factor and $n := \left\|(center_x, center_y, 1)^T\right\|$ the euclidean norm. We get $\widehat{e_{1\vartheta}} = s/n \cdot (center_x, center_y, 1)^T$.

[2] The term zoom is wrong at this point, because we do not change the focal length but move the virtual camera forwards.

Now we can transform points to the virtual view:

$$m_\vartheta \cong H_{\infty 1\vartheta} m_1 + \widehat{e_{1\vartheta}}\gamma_1 = m_1 + \frac{s}{n}(center_x, center_y, 1)^T \gamma_1$$
$$= [I|0] D_{1\vartheta} \begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix} = [I|0] \begin{bmatrix} 1 & 0 & 0 & \frac{s}{n} \cdot center_x \\ 0 & 1 & 0 & \frac{s}{n} \cdot center_y \\ 0 & 0 & 1 & \frac{s}{n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix}. \quad (7)$$

In order to combine a camera movement $D_{1\nu} = D_{12}^{\nu} (\nu \in \mathbb{R})$ along the geodesic with a movement $D_{1\vartheta}$ to the front into a shared transformation $D_{1\tau}$ we only need to combine the transformation matrices in accordance with Equation (3):

$$D_{1\tau} = \exp(\nu \log(D_{12}) + v \log(D_{1\vartheta})) \text{ with } \nu, v \in \mathbb{R}. \quad (8)$$

3.2 Camera Movement Orthogonally to the Geodesic and the Line of Sight

We will introduce another virtual camera movement apart from the geodesic. In contradiction to the preceding section, this movement occurs not to the 'front and back' but 'up and down' relative to the original camera pose[3].

The main idea is illustrated in Figure 2. L and R again denote the left and right reference camera, respectively. The virtual camera on the geodesic is labeled with V. The movement of V orthogonally to the geodesic and the line of sight is indicated by a dashed line.

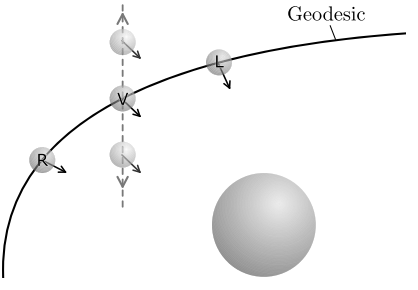


Fig. 2. Camera movement orthogonally to the viewing direction

The proceeding. Without loss of generality only the proceeding for the first reference image is examined. As a result of the fact that the new camera position μ is coplanar to the initial position we have rectified views. Hence, the homography at infinity is the identity matrix $H_{\infty 1\mu} = I$, and the epipoles lie at infinity according to the y-axis: $e_{1\mu} = (0, 1, 0)^T$.

Based on these observations can points in the virtual view μ be calculated with the following formula according to Equation (6):

$$m_\mu \cong H_{\infty 1\mu} m_1 + e_{1\mu}\gamma_1 = m_1 + (0, \gamma_1, 0)^T$$
$$= [I|0] D_{1\mu} \begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix} = [I|0] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} m_1 \\ \gamma_1 \end{pmatrix}. \quad (9)$$

[3] Pose: Position and orientation.

Again it is possible to combine these movements with any other displacement with Equation (3). Hence, together with the transformation from the preceding section, we are now able to place the virtual camera anywhere in 3-D space with a camera orientation given implicitly by the orientation of the two reference views.

4 Image Quality Optimization (Utilize Information of Both Reference Images)

In this section a theory will be discussed that allows the usage of the image information from *both* reference images. This is contrary to the processing in [6] where the view syntheses are generated from only one reference image.

Assuming that only the information of the *first* (w.l.o.g.) reference image is used by the procedure in [6], we will now concentrate on including the picture information of the *second* reference image. Thus the transformation for adjusting the second image along the geodesic according to D_{12} is to be found.

Therefore the homography at infinity $H_{\infty 21}$ can be calculated analog to Equation (5). We receive $H_{\infty 21} = H_1^{-1} H_2$ which leads to the coherence $H_{\infty 21} = H_1^{-1} H_2 = (H_2^{-1} H_1)^{-1} = H_{\infty 12}^{-1}$.

The epipoles of the two images are correlated with each other with respect to any homography, i.e. it can be used for example the homography at infinity: $e_1 = H_{\infty 21} e_2$. Using Definition 1 we compute D_{12}^{-1}:

$$\begin{aligned} D_{12}^{-1} &= \begin{bmatrix} H_{\infty 12} & e_2 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} H_{\infty 12}^{-1} & -H_{\infty 12}^{-1} e_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} H_{\infty 21} & -H_{\infty 21} e_2 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} H_{\infty 21} & -e_1 \\ 0 & 1 \end{bmatrix} =: \overline{D_{21}} \qquad \left(\text{Note: } D_{21} \overset{Def.1}{=} \begin{bmatrix} H_{\infty 21} & e_1 \\ 0 & 1 \end{bmatrix} \right). \end{aligned} \tag{10}$$

If the relative affine structure of the second reference image is computed with the inverse homography $H_{\infty 12}^{-1} = H_{\infty 21}$ (equally scaled) as the relative affine structure of the first reference image it is the negative equivalent of the relative affine structure of the first image. Referring to Equation (4) the points of the second image have to be transformed with the version of $\overline{D_{21}}$ adjusted to the negative relative affine structures. This adjusted matrix is D_{21} which is necessary to bring the second camera in the pose of the first camera on the same curve.

A property of the uncalibrated rigid transformation matrix is that the pose of the interstations $t \in \mathbb{R}$ computed for the first image via D_{12}^{t} and for the second image via $D_{21}^{(1-t)}$ are equal (see [10] together with Equation (10)). The two resulting images are then combined to the final view synthesis, e.g. by averaging them. This is illustrated in Figure 3. The left (L) and right (R) reference view are transformed and yield the *same* virtual view V.

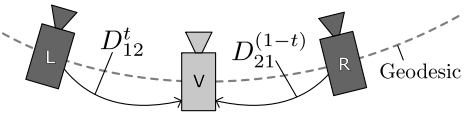


Fig. 3. Combination of two views

5 Experimental Results

In this section we show and discuss some results which were achieved with a MATLAB® implementation of the proposed methods.[4]

Given the two reference images[5] in Figure 4 we computed a smooth video sequence. Some frames of this sequence can be seen in Figure 5. The virtual camera performs a movement from the upper left to the lower right with a simultaneous motion backwards. This can be observed best while following the mound from frame to frame in relation to other objects in the scene. Compared to the results in [6] it can be observed that our 3-D camera placement allows a more flexible path generation. In addition the quality of the virtual images is notably improved as will be discussed in detail now.

Fig. 4. Pair of stereo images: Left and right reference view. The white dotted lines are added to accentuate the perspective differences.

Figure 6 illustrates the quality improvement achieved by using the image information from both reference images. Two different results of the same scene with a virtual camera describing the extrapolation $D_{21}^{1-(-0,5)} = D_{21}^{1,5}$ are shown. In Figure 6(a) only the image information of one reference image (right camera) is used while the image in Figure 6(b) is received by using the image information of both reference images. It can be clearly seen that the image computed from only one reference image contains much more holes. These holes are parts of the image for which the reference image does not provide any information. The holes still remaining in the virtual images can be closed for example by interpolation.

[4] For videos visit: http://cvpr.uni-muenster.de/research/viewsynthesis/
[5] Source: http://profs.sci.univr.it/~fusiello/demo/synth/

Fig. 5. Some frames of a video sequence computed from the images in Figure 4

(a) One reference image. (b) Both reference images. (c) Interpolation of (b).

Fig. 6. Quality improvement by exploiting the information of both reference images. The erroneous pixels were reduced about 28% by taking both reference images.

Using interpolation on the synthesis resulting from our method shown in Figure 6(b) yields the image in Figure 6(c), which represents a virtual view of excellent quality.

Performance. We evaluated our MATLAB® code on a 64-bit Linux machine with 2.50 GHz and 7.5 GB RAM. According to the three steps of the pipeline, the parameter estimation took about 11 seconds, the stereo analysis about 90 seconds, and the warping varied between 0.8 and 3 seconds per frame for an image size of $640 \times 480 \times 3$ (RGB). The variability in the warping step is due to a changing amount of pixels without image information (holes) which were filled by interpolation.

6 Conclusion

We extended the completely automated view synthesis procedure of A. Fusiello et al. [5,6] to a considerably more flexible procedure with enhanced quality.

After a brief introduction into the basic underlying methods, that allow only for a camera placement on a one-dimensional curve, we developed extensions that provide an arbitrary camera placement in 3-D space. Further we presented an approach to access the image information of all reference images, contrary to [6] where the virtual views are generated based on only one reference image. The gain of image quality could be demonstrated with an example.

Limitations and Future work. We experienced that the image quality of the resulting views almost exclusively depends on the results of the stereo matching procedure. Although the stereo matching step was not subject of this work we predict the most promising advances concerning image quality for view synthesis in general and the presented method in special in the field of stereo matching.

Furthermore, as the view synthesis procedure in [6] can handle more than two reference views, additional reference pictures could be included to provide more picture information for the virtual views. These additional pictures could also overcome the problem of occlusions in non-geodesic synthetic images.

References

1. Seitz, S., Dyer, C.: View Morphing. In: SIGGRAPH 1996, pp. 21–30 (1996)
2. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-Quality Video View Interpolation Using a Layered Representation. ACM Trans. Graph. 23(3), 600–608 (2004)
3. Criminisi, A., Blake, A., Rother, C., Shotton, J., Torr, P.H.: Efficient Dense Stereo with Occlusions for New View-Synthesis by Four-State Dynamic Programming. Int. J. Comput. Vision 71(1), 89–110 (2007)
4. Avidan, S., Shashua, A.: Novel View Synthesis by Cascading Trilinear Tensors. IEEE Transactions on Visualization and Computer Graphics 4(4), 293–306 (1998)
5. Fusiello, A., Irsara, L.: An Uncalibrated View-Synthesis Pipeline. In: ICIAP 2007: Proceedings of the 14th International Conference on Image Analysis and Processing, pp. 609–614. IEEE Computer Society, Los Alamitos (2007)
6. Fusiello, A.: Specifying Virtual Cameras in Uncalibrated View Synthesis. IEEE Trans. Circuits Syst. Video Techn. 17(5), 604–611 (2007)
7. Shashua, A., Navab, N.: Relative Affine Structure: Theory and Application to 3D Reconstruction from Perspective Views. In: CVPR 1994, pp. 483–489 (1994)
8. Shashua, A., Navab, N.: Relative Affine Structure: Canonical Model for 3D From 2D Geometry and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(9), 873–883 (1996)
9. Irsara, L., Fusiello, A.: Quasi-Euclidean Uncalibrated Epipolar Rectification. Rapporto di Ricerca RR 43/2006, Dipartimento di Informatica - University of Verona (2006)
10. Alexa, M.: Linear Combination of Transformations. ACM Trans. Graph. 21(3), 380–387 (2002)

Region Based Color Image Retrieval Using Curvelet Transform

Md. Monirul Islam, Dengsheng Zhang, and Guojun Lu

Gippsland School of Information Technology, Monash University, VIC 3842, Australia
{md.monirul.islam,dengsheng.zhang,
guojun.lu}@infotech.monash.edu.au

Abstract. Region based image retrieval has received significant attention from recent researches because it can provide local description of images, object based query, and semantic learning. In this paper, we apply curvelet transform to region based retrieval of color images. The curvelet transform has shown promising result in image de-noising, character recognition, and texture image retrieval. However, curvelet feature extraction for segmented regions is challenging because it requires regular (e.g., rectangular) shape images or regions, while segmented regions are usually irregular. An efficient method is proposed to convert irregular regions to regular regions. Discrete curvelet transform can then be applied on these regular shape regions. Experimental results and analyses show the effectiveness of the proposed shape transform method. We also show the curvelet feature extracted from the transformed regions outperforms the widely used Gabor features in retrieving natural color images.

1 Introduction

Regions are fundamental blocks for recent region based image retrieval (RBIR) techniques involving regions and semantic learning [1]. Texture feature is an essential component in most region based image retrieval (RBIR) techniques because of its strong discriminative power. Many texture feature extraction techniques have been proposed including spatial and spectral. Spatial techniques are subject to noise and difficult to obtain. So far, spectral methods, like Gabor [2, 3] and wavelet [4], have shown the best retrieval performance. Recent researches show that curvelet transform has significant advantages over Gabor due to curvelet is more effective in capturing curvilinear properties, like lines and edges [5]. It shows promising results in image de-noising [6], character recognition [7] and texture image retrieval [8]. To date, no application has been reported on real world image retrieval using curvelet transform. This paper applies curvelet transform in a region based image retrieval technique to retrieve color images. The application of curvelet transform in a region based technique is challenging due to the fact that curvelet transform requires rectangular images or regions, while segmented regions are usually irregular, as shown in Fig. 1.

Most of the existing RBIR techniques define a region as a set of small blocks of size 4 by 4 pixels and apply spectral transform on those blocks [9]. Then the feature of the region is calculated as the average feature of those blocks. This technique has

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 448–457, 2010.

drawbacks because it assumes that texture of a region can be represented by the blocks [10]. This assumption is not true, because it significantly loses the edge and line information of the region. To solve this problem, this paper proposes a novel method to extract regular shape regions from irregular shape regions. The curvelet transform is then applied on the regular shape regions to extract texture features. The effectiveness of the proposed shape transform is compared with that of the widely used zero padding method. Finally, we compare the retrieval performance between the curvelet feature and popular Gabor feature.

Fig. 1. Regions image segmentations

The rest of this paper is organized as follows. In section 2, we briefly introduce the curvelet transform, while section 3 describes curvelet feature extraction for irregular regions. The experimental results and comparison are presented in section 4. Section 5 concludes the paper.

2 The Curvelet Transform

The curvelet transform is a natural extension of the ridgelet transform. The continuous ridgelet transform for a given image $f(x, y)$ at scale a, translation b, and orientation θ, is defined as,

$$CRT_f(a,b,\theta) = \iint \psi_{a,b,\theta}(x,y) f(x,y) dxdy \tag{1}$$

where, the ridgelet $\psi_{a,b,\theta}(x,y)$ is a wavelet type function $\psi(x)$ with scale a, translation b, and rotation θ,

$$\psi_{a,b,\theta}(x,y) = a^{-1/2}\psi((x\cos\theta + y\sin\theta - b)/a) \tag{2}$$

Similar to Gabor wavelet, the ridgelet can be tuned at different scales and orientations to generate a set of curvelets. However, unlike the Gabor functions which cover only a part of the frequency spectrum, the set of curvelets covers the complete spectrum as shown in Fig. 2. Fig. 2(a) shows the frequency tiling by curvelet transform with 4 scale decomposition [11]. The shaded region is the response at scale 4 and orientation 9 (counting from top left corner for each scale). The figure shows that the entire spectrum is covered. Fig. 2(b) shows that there are many holes between the ovals in the frequency plan of the Gabor filters [3].

Given a digital image $f[m, n]$ of dimension M by N, the digital curvelet transform, $CT^D(a, b, \theta)$ is obtained using Equation (3).

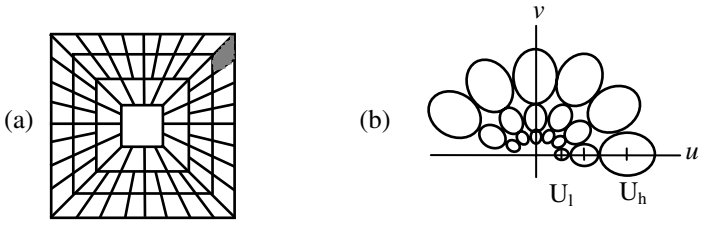


Fig. 2. Frequency spectrum coverage by (a) curvelet and (b) Gabor

$$CT^{D}(a,b,\theta) = \sum_{0\le m<M} \sum_{0\le n<N} f[m,n]\psi^{D}_{a,b,\theta}[m,n] \tag{3}$$

Equation (3) is implemented in frequency domain and can be expressed as,

$$CT^{D}(a,b,\theta) = IFFT(FFT(f[m,n]) * FFT(\psi^{D}_{a,b,\theta}[m,n])) \tag{4}$$

A detail description for the implementation of Equation (4) can be found in [8]. After obtaining the coefficients in CT^{D} (a, b, θ), the mean and standard deviation are calculated from each set of curvelet coefficients. Therefore, if n curvelets are used, a feature vector of dimension $2n$ is used to represent an image or region.

3 Curvelet Feature Extraction for Irregular Region

3.1 Irregular to Regular Shape Transform

As shown in Fig. 1, image-regions from segmentation are irregular. Spectral transforms, like Curvelet, require that the image or region should be rectangular. Therefore, irregular shape regions must be transformed to regular shape regions before applying the curvelet feature extraction process.

An irregular shape region can be transformed into a regular shape region by finding either the smallest outer rectangle or the largest inner rectangle. Fig. 3 shows outer and inner rectangles of few regions in red and blue lines, respectively. Though an outer rectangle is easy to find, it always includes non-region pixels. For example, the white spaces in the outer rectangles of Fig. 3 are non-region pixels. These non-region pixels need to be filled in. The features extracted from an outer rectangle heavily depend on the values which fill those non-region pixels. The most commonly used technique is 'zero-padding' which fills these positions with zeros. Zero padded regions are so different from the original region that the overall texture information of the region is significantly changed. In contrast, an inner rectangle consists of only valid region-pixels. Thus the features extracted from an inner rectangle are more accurate than the features extracted from an outer rectangle. Finding the largest inner rectangle is computationally expensive, while it is cheap to find the largest inner square because an efficient dynamic algorithm can be used [12]. However, an inner

Fig. 3. Inner and outer rectangles are in blue and red lines

square itself is useful only when it is large enough to cover a significant portion of a region. Unless this condition is met, the extracted texture feature will not truly represent the complete texture information of the region. Therefore, we propose a method to use both the outer rectangle and the largest inner square to extract a square shape with reasonable size. Fig. 4 outlines the algorithm for the shape transform.

1. Remove boundary pixels using pre-processing.
2. Find the bounding box of the pre-processed region.
3. Find the largest internal square of the desired region.
4. Extend the internal square to the size of the bounding box using mirror padding.
5. Super impose the bounding box over the mirror padded rectangle.
6. Extend the superimposed rectangle to a square of given size using mirror padding.

Fig. 4. Algorithm of transforming an irregular region to a square region

We describe the main idea of the algorithm of Fig. 4 in Fig. 5 using an example. The region in the upper left corner of Fig. 5(a) is to be transformed into a square region. Fig. 5(b) shows the region in grey scale. As segmented regions often include boundary pixels, these pixels need to be removed for the regions. Therefore, a pre-processing is applied to remove these non-image pixels. Fig. 5(c) shows the region after applying pre-processing on the region of Fig. 5(b). The red boundary is the outer rectangle. The largest inner square is then found and is shown in Fig. 5(d) with the white boundary. The inner square is used to fill in the blank space of the outer rectangle of the region of Fig. 5(c). At first, the inner square is enlarged to the size of the outer rectangle (Fig. 5(e)). To preserve the natural transition, mirror padding is used for this purpose. The region of Fig. 5(c) is then superimposed on the mirror padded region of Fig. 5(e). Fig. 5(f) shows the superimposed region. The original region is retained in Fig. 5(f). Furthermore, the blank space of the outer rectangle is filled with valid pixels from the region. Thus the filled space will carry the same texture information of the original region. As segmentation algorithms often segment images into quite small regions, superimposed rectangular regions can also be small. Therefore the rectangular region of Fig. 5(f) is further enlarged to a reasonable size of 128 by 128 by using mirror padding. Fig. 5(g) shows the final enlarged region which looks very similar to the original region. The texture property is also retained in the transformed region.

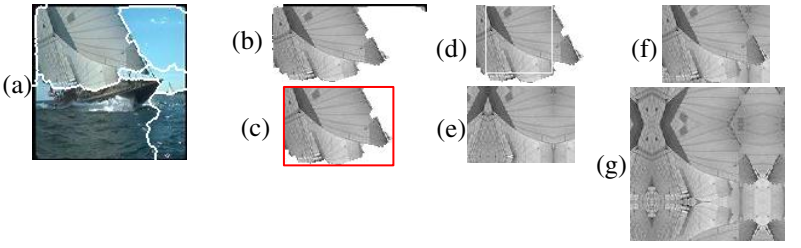


Fig. 5. Conversion from irregular to regular regions. (a) A segmented image. (b) The example region. (c) The region pre-processing. (d) The largest inner square. (e) Enlargement of the inner square (f) Superimposed region. (g) Final enlarged region.

3.2 Feature Extraction and Distance Measure

Curvelet feature extraction process is applied to each region. At first, each segmented image is transformed into a grey scale image. Each irregular image-region is then transformed to a regular shape region using the method described in section 3.1. Then curvelet feature is extracted from the regular shape region using the technique described in section 2.

For curvelet feature extraction, each regular shape region is decomposed into 4 scales. For decompositions at scale number 1, 2, 3, and 4, there are 1, 16, 32, and 1 subbands, respectively. However, due to the symmetric nature of curvelet, only half of the sub-bands at decomposition scale number 2 and 3 are necessary. Because a curvelet oriented at an angle, θ, produces the same coefficients as a curvelet oriented at an angle, $\pi+\theta$. Therefore, 26(=1+8+16+1) subbands are actually used. Mean and standard deviation are extracted from the coefficients of each subband. Therefore each region is represented and indexed by a curvelet feature vector of 52 dimensions.

During retrieval, a region of interest is given as a query. The feature vector of the query region is compared with the feature vectors of all regions in the database using L_2 distance measure. The distance, D, between a query region feature vector, Q, and a target region feature vector, T, is given by,

$$D^2 = \sum_{i=1}^{2n} (Q_i - T_i)^2 \tag{5}$$

Database regions are ranked based on the distance measures and displayed to the users.

4 Retrieval Performance

In this section, we first compare the effectiveness of the proposed shape transformed method with that of basic zero padding method using curvelet feature based retrieval. Then, the performance of curvelet feature is compared with that of Gabor feature. Region based image retrieval (RBIR) is used to test the performance.

To test the retrieval performance, 5,100 Corel images are segmented into 36,692 regions using one of the state-of-the-art segmentation algorithms, JSEG [13]. 3,259

regions are collected out of 36,692 regions. These regions are chosen because they significantly represent some concepts. The concepts of these regions are ape, balloon, bear, bird, butterfly, car, copter, deer, elephant, fighter plane, fireworks, flower, fox, horse, plane, tiger, and tree. For each region, two curvelet feature vectors are extracted. The first feature vector is extracted using the method described in Section 3.2 after applying the proposed shape transformed method of Section 3.1. In the second case, non region pixels of the outer rectangle of a region are filled with zeros. Then, the curvelet feature vector is extracted from the zero padded region.

The performance of the proposed mirror padded curvelet features is compared with that of zero padded curvelet features. As the ground truth of the database regions are known, each of the query regions is used as a query. The Euclidean distance measure is used to find the distance between the feature vectors of the query and a database region. The conventional 'precision vs. top K retrieval' curve is used as the performance measure. From each query, precisions are measured at 10 levels of K (that is, for K = 10, 20, 30, …, and 100). The average precisions are calculated from all of 3,259 queries at each of the 10 K values.The average retrieval performance for both the methods is shown in Fig. 6. It should be noted that coloured regions are difficult to retrieve using texture feature alone. In actual CBIR systems, texture feature is combined with colour to achieve good retrieval performance.

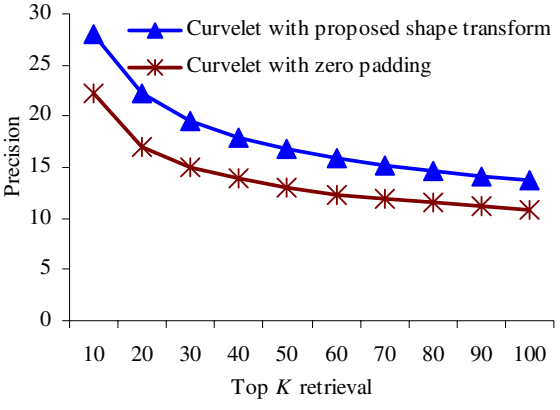


Fig. 6. Performance comparison of the proposed shape transform and the zero padding

As shown in Fig. 6, the curvelet features calculated from the proposed shape transform method significantly outperforms the curvelet features calculated from zero padding method. The reason is that zero padding method fills the outer space of a region with a smooth and constant texture pattern which is different from the original region. Therefore, the texture characteristics of the entire padded region drastically vary with that of the original region. As a result, the effectiveness of extracted curvelet feature decreases. On the other hand, the proposed method preserves the original pattern in the transformed region and no information is lost.

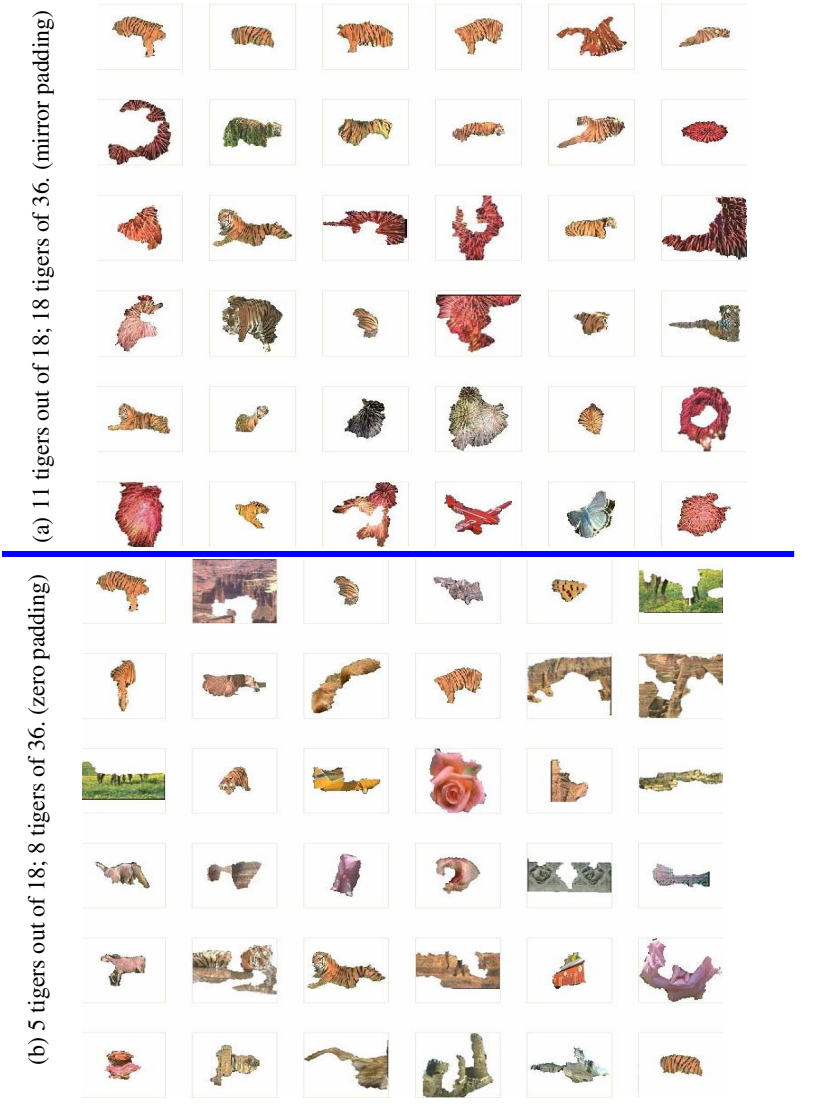


Fig. 7. Retrieval snapshots by different padding methods

Fig. 7 shows a few examples of region retrieval using both types of curvelet features. Top left region is the query. Regions are organized from left to right and top to bottom in increased distances from the query. In all the cases, the difference between the two is significant. For example, in the first case, when the top 18 regions are considered, transformed shape based curvelet feature retrieves 11 tiger regions, while zero padded curvelet only retrieves 5 tigers. When the top 36 regions are considered, the precision of the proposed feature is 18 out of 36, while that of zero padding is 8 out of 36.

Next, we compare the performance between curvelet and Gabor features. Gabor feature is extracted by using 4 scales and 6 orientations which are found to be the best parameters in [14]. This configuration generates 24 filters. Mean and standard deviation are calculated from each filtered output. Thus, each region is represented by a Gabor feature vector of 48 dimensions. Fig. 8 shows the comparison between the retrieval performance of curvelet and Gabor features

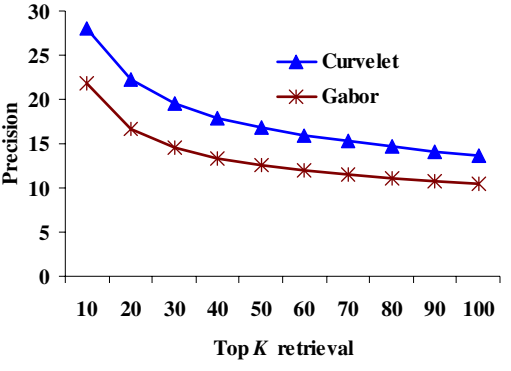


Fig. 8. Performance comparison between curvelet and Gabor features

As shown in Fig. 8, the performance of curvelet feature is significantly higher than that of Gabor feature in retrieving natural image-regions. There are several reasons behind this high performance of curvelet feature. Some reasons are due to the characteristics of natural images, while some others are due to the characteristics of curvelet itself. The textures of most natural images are irregular, even for the images from the same category. They cover the entire spectrum - from smooth to rough, fine to coarse, non-directional to very directional. Thus the filters together should cover the entire frequency spectra, and must be robust enough to capture the different variations of textures of natural images-regions. Curvelet has several advantages over Gabor in this regard [8]. Firstly, curvelets cover the entire spectra while Gabor filters loses some frequency information because of the uncovered holes created in spectra. Secondly, curvelet transform accurately captures texture information of natural image-regions at different scales because it scales the regions at different levels in addition to using scale modifiers for different ridgelets. In contrast, Gabor transform only uses scale modifiers for its filters and does not scale down/up the regions. Thirdly, curvelet transform has higher number of sub-bands at finer scales than coarser scales while Gabor transform has the equal number of subbands at all scales. It is quite natural that at finer scales information is more densely distributed than coarser scales. Thus finer scales need more sub-band divisions than coarser scales. Curvelet transform realizes this and thus each subband in curvelet transform captures frequency information more accurately than the subbands in Gabor.

Fig. 9 shows an example of firework retrieval using both curvelet and Gabor features. The firework region in the upper left corner is used as the query which is very irregular and difficult to retrieve. However, curvelet feature gives significantly better

retrieval results than Gabor feature. For example, when we consider the top 18 regions, curvelet and Gabor features retrieve 13 and 7 firework regions, respectively. When the top 36 regions considered, these features retrieved 22 and 9 firework regions. Fig. 9 also shows that curvelet transform captures the curvilinear properties of natural image-regions more accurately than Gabor filter.

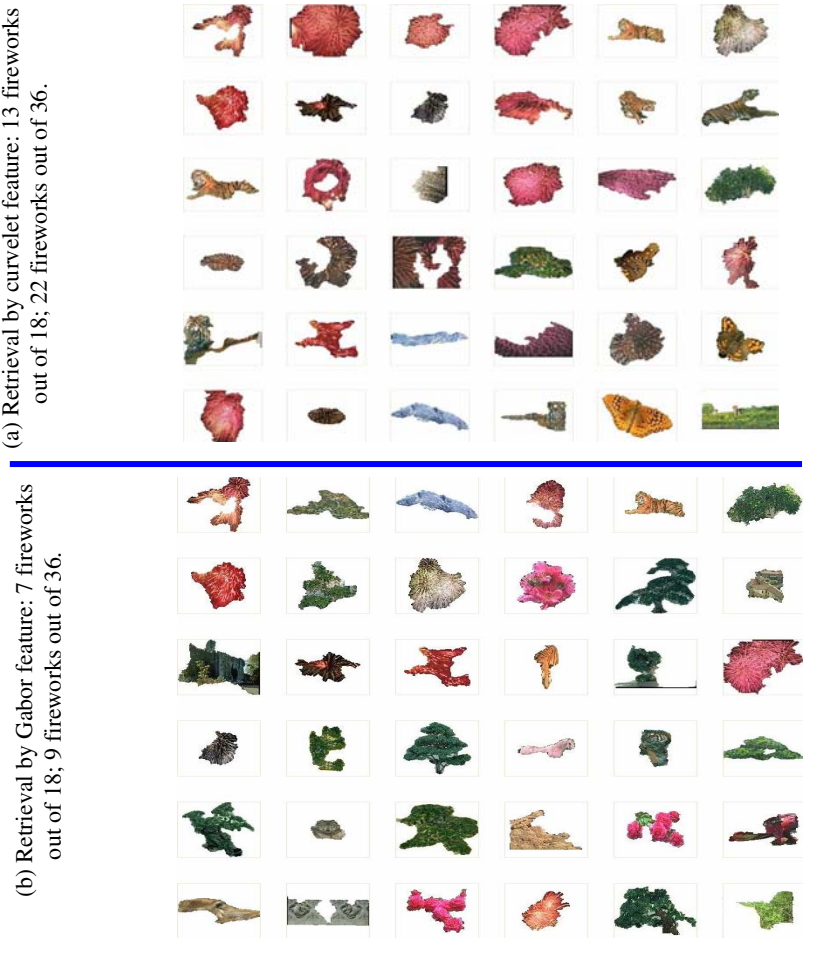


Fig. 9. First 36 retrieved regions by (a) curvelet and (b) Gabor features

5 Conclusion

This paper has presented a new region based color image retrieval technique using curvelet transform. The paper has two contributions. First, the curvelet transform has been applied for retrieving real world color images. A region based methodology has been used which facilitates query by object of interest. Second, a method of transforming irregular regions to regular regions is proposed. The shape transformation

method not only makes the curvelet feature extraction possible for segmented image-regions, but also preserves the original texture property of a region in the transformed region. Experimental result has shown that curvelet features calculated from the proposed shape transform method significantly outperforms the curvelet features calculated from zero padding method. Our experiment has also shown that RBIR performance based on curvelet feature extracted from transformed regions significantly outperforms the widely used Gabor features in retrieving natural color images. Currently, we are investigating the application of curvelet feature in semantic learning of image-regions. Rotation and scale invariance issues are still unsolved problems which will be addressed in our future work to further improve curvelet's retrieval performance.

References

1. Liu, Y., et al.: Region-based image retrieval with high-level semantics using decision tree learning. Patt. Recog. 41(8), 2554–2570 (2008)
2. Manjunath, B.S., Ma, W.Y.: Texture Features for Browsing and Retrieval of Large Image Data. IEEE Trans. on PAMI 18(8), 837–842 (1996)
3. Manjunath, B.S., et al.: Introduction to MPEG-7. John Wiley & Son Ltd., Chichester (2002)
4. Bhagavathy, S., Chhabra, K.: A Wavelet-based Image Retrieval System. Technical Report—ECE278A, Vision Research Laboratory, University of California, Santa Barbara (2007)
5. Do, M.N.: Directional Multiresolution Image Representations. PhD Thesis, EPFL (2001)
6. Starck, J., et al.: The Curvelet Transform for Image Denoising. IEEE Trans. on Image Processing 11(6), 670–684 (2002)
7. Majumdar, A.: Bangla Basic Character Recognition Using Digital Curvelet Transform. Journal of Pattern Recognition Research 1, 17–26 (2007)
8. Sumana, I.J., et al.: Content based image retrieval using curvelet transform. In: Proc. of Int. workshop on MMSP, pp. 11–16 (2008)
9. Wang, J.Z., et al.: Simplicity: semantics-sensitive integrated matching for picture libraries. IEEE Trans. on PAMI 23(9), 947–963 (2001)
10. Liu, Y.: Region-based image retrieval with high-level semantics. Ph. D. Thesis, Monash University (2006)
11. Candes, E., et al.: Fast Discrete Curvelet Transforms. Multiscale Modeling and Simulation 5(3), 861–899 (2006)
12. Coreman, T.H., et al.: Introduction to Algorithms. The MIT Press, Cambridge (2001)
13. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. IEEE Trans. on PAMI 23(8), 800–810 (2001)
14. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Trans. on PAMI 18(8), 837–842 (1996)

Extracting Spatio-temporal Local Features Considering Consecutiveness of Motions

Akitsugu Noguchi and Keiji Yanai

Department of Computer Science,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
noguchi-a@mm.cs.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. Recently spatio-temporal local features have been proposed as image features to recognize events or human actions in videos. In this paper, we propose yet another local spatio-temporal feature based on the SURF detector, which is a lightweight local feature. Our method consists of two parts: extracting visual features and extracting motion features. First, we select candidate points based on the SURF detector. Next, we calculate motion features at each point with local temporal units divided in order to consider consecutiveness of motions. Since our proposed feature is intended to be robust to rotation, we rotate optical flow vectors to the main direction of extracted SURF features. In the experiments, we evaluate the proposed spatio-temporal local feature with the common dataset containing six kinds of simple human actions. As the result, the accuracy achieves 86%, which is almost equivalent to state-of-the-art. In addition, we make experiments to classify large amounts of Web video clips downloaded from Youtube.

1 Introduction

Recently the number of videos people have and on the Web is increasing rapidly, and content-based video analysis becomes more important. For example, video summarization and content-based video retrieval help users to find videos which they want to watch efficiently.

As one of the methods for that, recently spatio-temporal local features have been proposed as image features to recognize events or human actions in videos. Local features are commonly used for object recognition because of its robustness about noise, rotation and occlusion. Recently this idea has been imported to event and action recognition for video. Video analysis with spatio-temporal features is new, and has not been explored much yet. Then, in this paper, we propose yet another spatio-temporal feature based on the SURF local feature. The existing methods of extraction features from videos are classified into two types. The first one is extracting global features from a whole video. The second one is extracting many local spatio-temporal features from a video. In this paper, we focus on the second type of methods based on spatio-temporal features.

To extract spatio-temporal feature, local cuboid is one of the common methods. However, it is difficult to decide cuboid size and features extracted from

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 458–467, 2010.

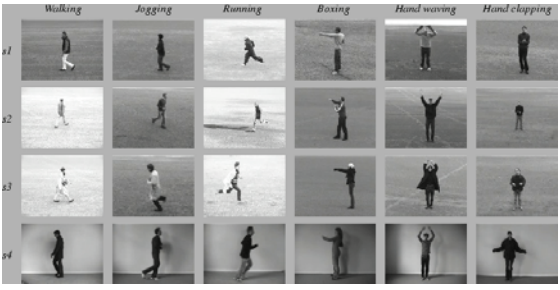


Fig. 1. KTH dataset

cuboid. Dollar et al.[1] and Laptev et al.[2] proposed extracting Histogram of Gradient (HoG) and Histogram of Flow (HoF) from a cuboid, respectively. Extracting such features from a whole cuboid is costly in terms of computation and is not robust to noise generally.

In this paper, we detect spatio-temporally interest points and extract local pattern around them as features by extending the SURF method. This proposed method is more simple, fast and efficient method to extract spatio-temporal features than the existing ones.

In the experiment, we classify simple human motion. We use KTH dataset (Figure 1), which is a standard dataset for evaluation of human action recognition methods. This dataset contains six kinds of simple human primitive actions: "walking", "running", "jogging", "boxing", "hand waving" and "hand clapping". This dataset assumes that "each video contains only single human and action", and "no camera motion". As the result of classification experiments, we obtain the 86% classification rate. As an additional experiment, we classified shots of Web videos which are 100 soccer videos downloaded from Youtube.

In the rest of this paper, we describe related work in Section 2. Then we explain the proposed method in Section 3. Section 4 describes the experimental results. Finally we conclude this paper in Section 5.

2 Related Work

The existing methods of extraction features from videos can be classified into two types. The first one is tracking major parts of human bodies and extracting features from their regions. However, this method assumes that tracking and detection of body parts are almost successful. This assumption is sometimes difficult.

The other one is sampling many local cubic spatio-temporal regions, which is called "cuboid", from a video, and extracting features from cuboids. In this paper, we focus on this second type of methods based on spatio-temporal features.

Dollar et al. proposed the method to detect local cuboids to apply 2-D Gaussian kernels in the spatial space and 1-D Gabor filters for the temporal direction [1], and they generated video visual words by vector-quantizing local cuboids in the same way as bag-of-visual-words for object recognition [3].

Fig. 2. Detected interest points by the SURF

Laptev et al proposed STIP (Spatio-Time Interest Points) [2] as a method to detect cuboids. This method can be regarded as an extension of Harris detector. They extracted Histogram of Gradient (HoG) and Histogram of Flow (HoF) from detected cuboids as features.

Alireza et al. proposed to extract low-level optical flows from cuboids and select good features from them with boosting to improve accuracy of classification [4].

However, computational cost of extracting features from cuboids by the methods described above is relatively high. In addition, it is difficult to decide the proper size of cuboid. To overcome these problem, in this paper, we propose to detect interest points using SURF [5] and Lucas-Kanade optical flow detection methods [6] both of which are very fast detectors and extract features by tracking interest points instead of cuboids. Since we do not use cuboids, the proposed method is more simple, fast and efficient method to extract spatio-temporal features than the existing ones.

3 Proposed Method

Our proposed method consists of four steps. In the first step, we detect interest points and extract SURF features for the detected points employing the SURF (Speeded-Up Robust Feature) [5] from the frame images which are extracted from a given video every N frames. Extracted SURF descriptors represent local appearances around interest points. Figure 2 shows that extracted interest points by the SURF, which are candidate points for tracking. In the second step, we estimate the degree of motion for each candidate points based on optical flows computed by the Lucas-Kanade [6], and select points having motion from the candidate points. This is because interest points without motion are not suitable for the points from which spatio-temporal features are extracted. In the third step, we track each tracking point locally in the temporal direction and extract motion features. In the forth step, we generate spatio-temporal features by combining SURF features and motion features for the points in the third step.

3.1 Extraction of Appearance Features

In the proposed method, we extract both local appearance features and local motion feature, and combine them into local spatio-temporal features. As local appearance features, we use the SURF descriptor [5].

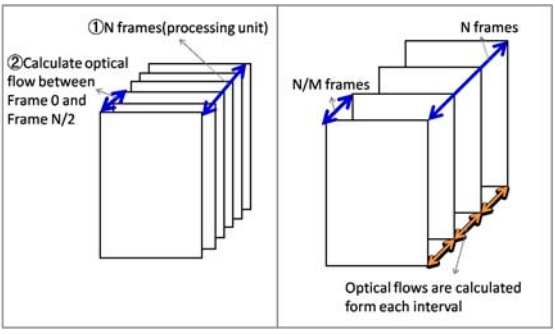


Fig. 3. Selecting frames from which optical flows are extracted (left). Extracting optical flows from the selected frames (right).

The SURF is a method to extract and describe local features from one still image. Although its function is the same as SIFT [7], its processing is much lighter and faster than SIFT. The SURF method consists of two steps: detector and descriptor. In the part of the SURF detector, it selects interest points based on the Hessian matrix. In the part of the SURF descriptor, it describes local patterns around detected points with 64-d vectors per point based on the Haar-like wavelet. Refer to [5] for the detail. We obtain SURF vectors the number of which is the same as the number of the interest points. However, the SURF vectors used as actual descriptor of a video are selected in the next step.

3.2 Selection of Motion Points

In this step, we select in-motion points from all the points detected by the SURF detector by optical flow analysis.

As mentioned before, we apply the SURF detector every N frames. Then, we calculate optical flows between the first frame and the $N/2$-th frame by Lukas-Kanade optical flow detector [6] as shown in the left side of Figure 3, and select the points where optical flows are detected among the points extracted by the SURF detector. We call such points as "motion points". In the proposed method, we extract both spatially local appearance features and temporally local motion features for each motion point.

3.3 Extraction of Motion Features

In the third step, we extract optical flows to generate motion features from $M-1$ intervals among the N frames which is a unit of motion processing, after picking up M frames out of N frames (M should be a factor of N). As shown in the right side of Figure 3, we calculate optical flows from $M-1$ consecutive intervals at each motion point in order to consider consecutiveness of motions. In case that M is 1, we can extract detailed motions. On the other hand, In case that M

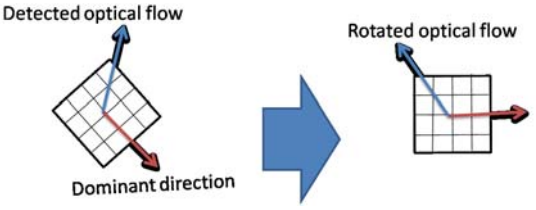


Fig. 4. Normalizing the direction of an optical flow by rotating it based on the dominant direction detected by the SURF detector

equals to N, motion information becomes condensed. In the experiment, we set both N and M as 5.

As representation of motion features, we generate a 5-d vector for each interval of each motion point from the motion matrix estimated by the Lucas-Kanade method [6]. The 5-d vector consists of x^+, x^-, y^+, y^- and no optical flow x^0, where x^+ means the degree of the positive elements along x-axis and x^- means the degree of the negative elements along x-axis. The motion feature for each interval is normalized so that the summation of all the elements equals to 1. We combine M 5-d vectors extracted from $M-1$ intervals into one motion vectors for each motion points, and totally the dimension of motion feature becomes $(M-1) \times 5$.

We hope that this feature is robust about rotation. The same feature should be extracted from "walk to right" and "walk to left", since our objective is proposing spatio-temporal features to categorize actions ignoring the directions of actions. To this end, in this paper, we propose to rotate optical flows along the dominant direction of visual features to normalize their direction. Figure 4 shows the rotation of an optical flow.

The rotated optical flow vector (x, y) are represented as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{1}$$

where (x_0, y_0) is the original optical flow vector for the motion point, and θ is the dominant direction of the SURF descriptor at the motion point.

3.4 Generation of Local Spatio-temporal Features

In the final step, in the proposed method, we extract both local appearance features and local motion features, and we combine local appearance features extracted in the first step and local motion features extracted in the third step into local spatio-temporal features.

The SURF-based appearance feature is represented by a 64-d vector, and the motion feature is represented by a $(M-1) \times 5$-d vector. After weighting the motion vector with w, we concatenate both vectors into in one $(64+(M-1)\times 5)$-d vector.

In the experiment, we set 5 to both M and N, and totally the dimension of the final feature vector becomes 84. In the experiment, we explored the optimal weight. As the result, we found that 2.5 is optimal for w.

4 Experimental Results

We made experiments to evaluate the proposed feature by classifying Web videos as well as simple human actions. In this section, we describe classification methods, datasets and results.

4.1 Action Recognition

Dollar et al.[1] classified human action employing bag-of-video-words. Bag-of-video-words (BoVW) is an extension of bag-of-feature (BoF) for action recognition. Following this, we generate bag-of-video-words from the proposed local spatio-temporal features, and classify human action by a support vector machine (SVM) with a RBF kernel.

First, we extract local spatio-temporal features proposed in this paper from training video data and generate a codebook by k-means clustering from all of the extracted features. Then, a BoVW vector is generated based on the codebook for each training video, and we train a SVM with the generated BoVW vectors. Next, each test video is also converted into a BoVW vector based on the precomputed codebook, and we classify test videos with the trained SVM.

As data set, we use the KTH dataset which is commonly used for benchmark test of spatio-temporal features. This dataset contains six kinds of primitive motions such as "walking", "running", "jogging", "boxing", "hand waving" and "hand clapping". This dataset assumes that "there is no camera motion" and "each video contain only one human and motion". At each motion, 25 individuals engaged 4 times, wearing different clothing. So each motion contains 100 videos. In the experiment, we did a multi-class classification with 5-fold cross validation employing the 1-vs-rest strategy. Note that the average length of videos in the KTH dataset is about 20 second long, and we extracted about 4000 features from each video.

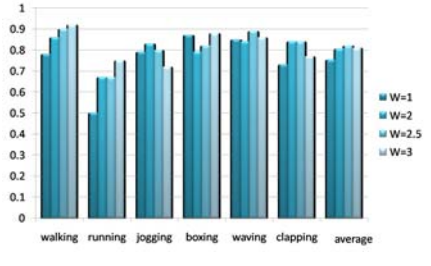


Fig. 5. Results in case of changing the motion weight w

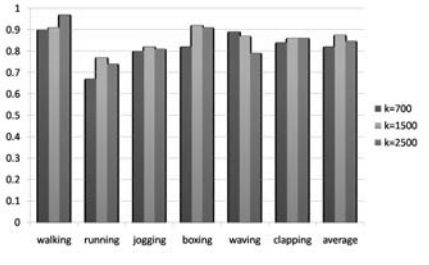


Fig. 6. Results in case of changing the codebook size k

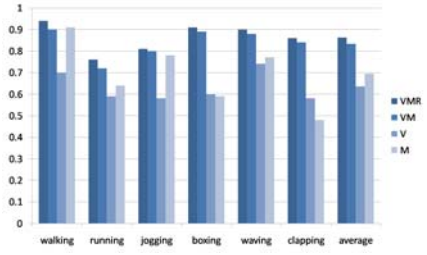


Fig. 7. Results by four types of combinations of features

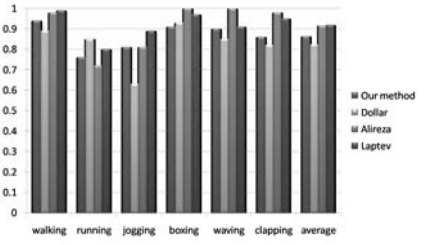


Fig. 8. Comparison with other results by the state-of-the-art methods

First, we explored optimal parameters of the motion weight w and the codebook size k. Figure 5 shows that classification rates of the six actions and their average in case of changing the motion weight w with 1, 2, 2.5 and 3. Figure 6 shows results in case of changing the codebook size k with 700, 1500 and 2500. These results indicate that the case of $w = 2.5$ and $k = 1500$ performed well. We used this setting for all the rest of the experiments,

In the next experiments, we evaluate the following four combinations of the extracted features.

1. visual appearance + motion + rotation (VMR)
2. visual appearance + motion (VM)
3. visual appearance (V)
4. motion (M)

Figure 7 shows the results of the classification rates for the six motions and their average. The average accuracy of VMR and VM both of which combine visual appearance and motion features are better than V and R both of which utilize only a single feature. VMR is better than VM, which indicates that considering rotation improved the results.

The single motion feature (M) performed well for "walking", "running" and "hand waving", while for "boxing" and "hand clapping" the results are very bad. This is because both actions of "boxing" and "hand clapping" include only horizontal motion as shown in Figure 1. Since "hand waving" contains not only horizontal motion but also small vertical motion, we can classify this action with only motion features relatively well.

On the other hand, the single visual feature (V) did not performed well for all the actions, and especially did not for "walking", "running" and "jogging" since appearances of these actions are very similar to each other.

Table 1-4 shows the confusion matrix of the classification results by four types of combinations of the features. Regarding all the combinations, the results for "walking " was good. On the other hand, it is difficult to classify "running" and "jogging" for all the combinations, because these two actions are so similar to each other that sometimes it is difficult for even human to classify.

Table 2 and Table 3 show the confusion matrices in case of only the visual appearance feature (V) and only the motion feature (M), respectively. From these

Table 1. Confusion matrix for VMR

	walking	running	jogging	boxing	waving	clapping
walking	0.94	0.02	0.03	0.01	0	0
running	0.02	0.76	0.22	0	0	0
jogging	0.04	0.15	0.81	0	0	0
boxing	0.01	0	0	0.91	0.02	0.07
waving	0	0	0	0.04	0.9	0.06
clapping	0	0	0	0.1	0.03	0.86

Table 2. Confusion matrix for V

	walking	running	jogging	boxing	waving	clapping
walking	0.7	0.13	0.16	0.01	0	0
running	0.1	0.59	0.21	0	0	0
jogging	0.12	0.29	0.58	0	0	0.01
boxing	0.13	0.13	0.1	0.6	0.03	0.01
waving	0.03	0.09	0.01	0.05	0.74	0.08
clapping	0.04	0.05	0.02	0.06	0.25	0.58

Table 3. Confusion matrix for M

	walking	running	jogging	boxing	waving	clapping
walking	0.91	0	0.06	0.03	0	0
running	0	0.64	0.3	0	0.02	0.04
jogging	0.04	0.13	0.78	0.02	0.03	0
boxing	0.01	0	0	0.59	0.32	0.08
waving	0	0	0.01	0.17	0.77	0.05
clapping	0	0	0	0.18	0.33	0.48

Table 4. Confusion matrix for VM

	walking	running	jogging	boxing	waving	clapping
walking	0.9	0.01	0.07	0.01	0	0
running	0.01	0.72	0.27	0	0	0
jogging	0.01	0.18	0.8	0.01	0	0
boxing	0	0	0	0.89	0	0.11
waving	0	0	0	0.06	0.88	0.06
clapping	0	0	0	0.13	0.02	0.84

tables, we found that it is difficult to classify "walking", "running" and "jogging" with only the visual appearance feature, while "'boxing', "hand waving" and "hand clapping" tend to be confused with only the motion feature.

Table 4 shows the confusion matrix in case of the visual appearance and motion feature without rotation. Compared to Table 1 (VMR), the accuracy of classification for all the action are worse. This means considering rotation contributes to improve the classification results.

Finally we compared our results to the other results by the state-of-the-art methods such as Dollar et al. [1], Alireza et al. [4] and Laptev et al. [2] as shown in Figure 8. The average classification rate by our method was 86%, one by the Dollar's method is 82.3%, one by the Alireza's method is 91.5% and one by the Laptev's method is 91.8%. Therefore, the proposed method is almost equivalent to the state-of-the-art methods.

4.2 Web Video Shot Classification

We classify Web video shots by k-means clustering to confirm efficiency of our features. Classifying Web video shots helps search video.

This experiment consists of four steps: (1) collect Web video, and divide them into shots by comparing HSV color histograms of consecutive frames, (2) extract the proposed feature from each shot, (3) build BoVW vectors and (4) cluster shots extracted from a single video with $k = 8$ or all the video with $k = 50$. In the experiment, we used 100 soccer videos collected from the Youtube.

Figure 9 shows the result of Web video shot clustering for a single video. This figure shows only 3 clusters out of 8 clusters. The cluster in the top row includes only shots taken from far places, the shots in the cluster in the middle row are taken near the field relatively, and the shots in the bottom are close-up of players.

Figure 10 shows 3 clusters out of 50 clusters as clustering results for all the video shots. Most of the shots in the cluster in the top row are taken from far places, and the shots in the middle are taken mainly for players. On the other hand, the bottom cluster contains many noisy shots. Overall, shot clustering performed well, and it shows that the proposed feature is also effective to classify Web video.

Fig. 9. Result of web video shot clustering per single video: cluster of far angle(top), near angle(middle) and closed-up person(bottom)

Fig. 10. Result of all web video shot clustering: cluster of far angle(top), near angle(middle) and noisy(bottom)

However, in this experiment, we extracted ten thousands of features on average and 200 thousand features at most from one shot. This is because of camera motion. For shots with camera motion, all extracted interest points are detected as motion points, so that processing time becomes larger. To solve this, we need to detect the direction and speed of camera motion and compensate it for motion features. This is one of our future work.

5 Conclusion

In this paper, we proposed a yet-another spatio-temporal feature. Proposed method consists of two parts: extracting visual appearance features and extracting motion features. First, we select candidate points based on the SURF detector. Next, we calculate several motion features at each point with local temporal units divided in order to consider consecutiveness of motions. Since

our proposed feature is intended to be robust to rotation, we rotate optical flow vectors to the dominant direction of extracted SURF features.

In the experiments, we evaluate the proposed spatio-temporal local feature with KTH. As the result, the accuracy achieves 86%, which is almost equivalent to state-of-the-art. In addition, we make experiments to classify large amounts of Web video clips downloaded from Youtube, and indicate the efficiency of our feature.

In future work, we can consider two ways. The first one is to improve the proposed feature to add more features, to improve feature descriptors, and to consider camera motions. The second one is to apply the proposed feature and build applications, such as content-based video retrieval, video summarization, and video surveillance system.

References

1. Dollar, P., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. of Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65–72 (2005)
2. Laptev, I., Lindeberg, T.: Local descriptors for spatio-temporal recognition. In: Proc. of IEEE International Conference on Computer Vision (2003)
3. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc. of ECCV Workshop on Statistical Learning in Computer Vision, pp. 59–74 (2004)
4. Alireza, F., Greg, M.: Action recognition by learning mid-level feature. In: Proc.of IEEE Computer Vision and Pattern Recognition (2008)
5. Herbert, B., Andreas, E., Tinne, T., Luc, G.: Surf: Speeded up robust features. Computer Vision and Image Understanding, 346–359 (2008)
6. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 91–110 (2004)
8. Fanti, C., Perona, P.: Hybrid models for human motion recognition. In: Proc. of IEEE Computer Vision and Pattern Recognition (2005)
9. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. International Journal of Computer Vision 50(2), 203–226 (2002)
10. Yacoob, Y., Black, M.J.: Parameterized modeling and recognition of activities. Computer Vision and Image Understanding 72(2), 203–226 (2002)
11. Konrad, S., Luc, G.: Action snippets: How many frames does human action recognition require? In: Proc. of IEEE Computer Vision and Pattern Recognition (2008)

Multi-view Texturing of Imprecise Mesh

Ehsan Aganj[1], Pascal Monasse[1], and Renaud Keriven[1]

IMAGINE, École des Ponts ParisTech 6 Av Blaise Pascal - Cité Descartes, Marne-la-Vallée, France

Abstract. Reprojection of texture issued from cameras on a mesh estimated from multi-view reconstruction is often the last stage of the pipeline, used for rendering, visualization, or simulation of new views. Errors or imprecisions in the recovered 3D geometry are particularly noticeable at this stage. Nevertheless, it is sometimes desirable to get a visually correct rendering in spite of the inaccuracy in the mesh, when correction of this mesh is not an option, for example if the origin of error in the stereo pipeline is unknown, or if the mesh is a visual hull. We propose to apply slight deformations to the data images to fit at best the fixed mesh. This is done by intersecting rays issued from corresponding interest points in different views, projecting the resulting 3D points on the mesh and reprojecting these points on the images. This provides a displacement vector at matched interest points in the images, from which an approximating full distortion vector field can be estimated by thin-plate splines. Using the distorted images as input in texturing algorithms can result in noticeably better rendering, as demonstrated here in several experiments.

1 Introduction

Recovering 3D geometry from multi-view still images or videos is the focus of the stereo research community in computer vision, robotics, and photogrammetry [1]. Usage dictates the requirements and priorities about accuracy of the estimated depth information: from rough precision for obstacle avoidance in robot navigation to highly precise and controlled measurements in telemetry and surveying. Despite years of research and development of computing capacities, and whereas the mathematical foundations are well understood [2,3,4], the required precision is not always practically reachable, which may be due to faulty calibration (uncorrected geometric distortion, imprecise focal position), approximations (interpolation of disparity in non-textured regions), or plain errors of algorithms in presence of unexpected conditions (specular surfaces, transparency, etc). Also several stereo pipelines include a step of global, non-convex energy minimization, as for example [5,6,7]. As they typically involve a gradient descent scheme, they are susceptible of stopping at a local minimum and have no way of recovering a better 3D geometry. Other methods involve a careful succession of heuristics to refine a visual hull obtained from silhouettes, as for example [8]. The base hypotheses of such heuristics may also be somewhat in default. In other cases,

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 468–476, 2010.

the visual hull is used directly for efficiency reasons. Whereas the resulting information may be unusable for precise scientific measures, it may still be useful and sufficient in motion capture for example. In that case, the rendering should mask as best as possible the incorrect geometry.

While algorithms exist that select the image to use as texture on each part of the mesh to minimize illumination change artifacts, they assume that the images are compatible with the mesh. In our case, that assumption does not stand and we must do correct rendering in spite of these inconsistencies. As the mesh is already the result of an optimization, it cannot be refined. The only possibility is to modify the images themselves. This is the approach of Eisemann *et al.* in [9]. The authors warp the input images by aligning reprojected images through optical flow estimation, for which they use a near-real-time GPU implementation. By contrast, we propose to use feature points as tie points for the registration of images, and to warp the images following a thin-plate spline approximation of the displacement field. Computational cost is normally low, as correspondence of tie points is often already computed and used earlier in the stereo pipeline to estimate epipolar geometry.

Recent work of Tzur and Tal [10] is an interesting approach to the problem. The model is assumed to fit imperfectly with the image, and given a set of projected vertices, a local projection matrix is estimated. The final warp is a weighted average of these local maps. Notice however that the method requires manual input of some projected vertices of mesh in the image. An interactive software specialized to plant modeling is also described in [11].

The rest of this paper is organized as follows. Section 2 describes the details of our algorithm and the required mathematical foundation. Section 3 shows experimental results of this method on diverse data. Finally we draw some conclusions in Sect. 4.

2 Morphing Images to Adapt to the Mesh

2.1 Overview of the Algorithm

Instead of correcting the mesh to fit the input images, which we assume we cannot do as the mesh is already obtained as some optimum, we correct the input images to fit the output mesh. We suppose that the camera positions, orientations and internal parameters, so as the mesh, are all correct, and we look for deformations in each input image to fit them. This is done in 4 steps:

1. Find matching points in different views.
2. Project on the mesh the obtained 3D points and reproject them onto the views.
3. Approximate the resulting sampled vector field in each image and deform them accordingly.
4. Use a multi-view texturing algorithm for rendering.

Notice that the match points detection is often already done as a first step in the stereo pipeline for calibration, therefore this entails no additional computation. Next sections give details on these different steps.

2.2 Interest Point Matching

Detection of points that have a non-ambiguous local neighborhood has seen remarkable progress in the last few years. They are some kind of generalized extrema or corners. Most of these encode their neighborhood with a similarity invariant signature, although affine invariance can be partly accommodated. Most popular of those are SIFT [12], which correspond to local extrema in the Gaussian pyramid, or generalized corners, and MSER [13], which are centroids of contrasted upper or lower level sets of the image radiometry. Any type of feature points can be used to match between different views [14]. We use SIFT points in our experiments, although MSER would also fit.

As noticed above, the interest points are already computed for calibration of the stereo system, and provide 3D point clouds for the initial mesh. However the mesh is often subsequently modified by some smoothing procedure, and then the 3D points are not anymore on the mesh. The next step measures this difference to adapt the images.

2.3 Reprojection of 3D Points through the Mesh

Reprojection is illustrated in Fig. 1. Intersecting rays passing through matching feature points via the respective focal points yields the 3D point position. Ideally, these rays would intersect in 3D, but because of imprecise calibration or imprecise detection they may not[1]. The least squares error solution is the 3D point that minimizes the sum of square distances to the rays and can be computed by a closed formula. Such a point is expected to be on the mesh, but because of the imprecision of the mesh, it may reside nearby. A natural adjustment is to project the 3D point P on the mesh, yielding a point P_M. We can then assume that P_M is the real 3D position and that the images are faulty. We reproject P_M on the images where it has been observed, yielding corrected positions of the feature points.

To ignore outliers, we simply reject the 3D points that are too far from the mesh. Otherwise, a single large mishap can distort the applied warping and ruin the correction effect.

2.4 Dense Deformation

Previous step indicates the desired position of matched feature points so that they correspond to 3D points on the mesh. However we need a dense deformation of each image to accommodate these displacements. In other words, in each image we are looking for an interpolation or approximation of a vector field irregularly sampled. A standard technique for that is using thin-plate splines [15,16]. Given the n feature points p_i and their reprojected positions p'_i through the mesh, thin-plate splines minimize the energy:

$$E(f) = \sum_i \|p'_i - f(p_i)\|^2 + \lambda \int \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dx\,dy \quad (1)$$

[1] Bundle adjustment would try to enforce these intersections as best as possible.

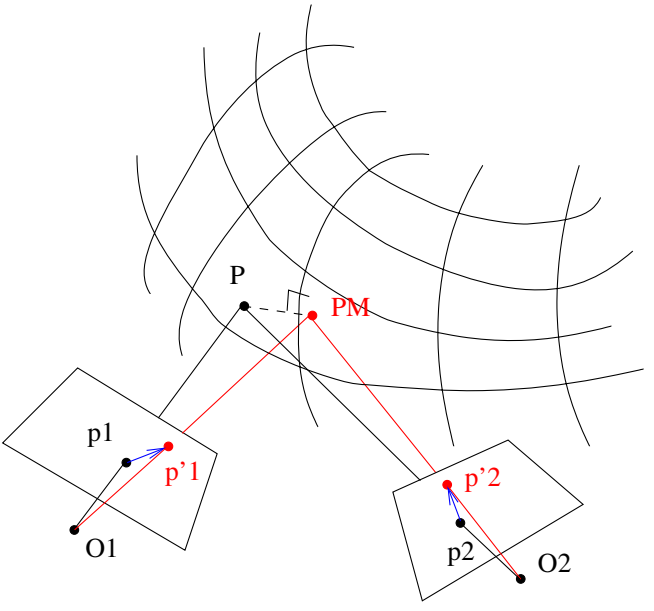


Fig. 1. Projection of 3D points through the mesh. Corresponding feature points p_1 and p_2 allow to recover a 3D point P. This point in projected on the mesh M to P_M, which would be observed at p'_1 and p'_2.

with a 2-variable function f of the form

$$f(z) = Az + \sum_i \Phi(\|z - p_i\|)w_i \ , \tag{2}$$

with A a plane affine transform, Φ a kernel function, usually $\Phi(r) = r^2 \log r$, and w_i a list of 2-vectors representing the non-affine part of the transform.

Defining K as the $n \times n$ symmetric matrix with entries $K_{ij} = \Phi(\|p_i - p_j\|)$, P as the $3 \times n$ matrix whose column j is composed of homogeneous coordinates of p_j, $P(:, j) = (x_j, y_j, 1)^T$, and P' the $2 \times n$ matrix whose column j is composed of Cartesian coordinates of p'_j, we minimize:

$$E(A, W) = \|P' - AP - WK\|^2 + \lambda \operatorname{trace}(WKW^T) \ ,$$

with A the 2×3 affine transform matrix and W the $2 \times n$ concatenation of the w_j written in columns. The involved norm is the Frobenius norm $\|X\|^2 = \operatorname{trace}(X^T X) = \sum_{i,j} X_{ij}^2$, that is the sum of square coefficients of X, associated to the scalar product $< X, Y >= \operatorname{trace}(X^T Y)$.

Equating to 0 the gradients of E (relative to this scalar product), with respect to A and W, yields:

$$(P' - AP - WK)P^T = 0 \tag{3}$$

$$(P' - AP - WK)K + \lambda WK = 0 \tag{4}$$

Using the QR decomposition of $P^T = Q_1 R$ (see [17]), let Q_2 be any $n \times (n-3)$ matrix such that $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ is orthogonal. Right-multiplying (4) by $K^{-1}Q_2$, we get

$$P'Q_2 - WKQ_2 + \lambda WQ_2 = 0 \ ,$$

so that

$$WQ_2 = P'Q_2(Q_2^T K Q_2 - \lambda I)^{-1}$$

and since $WQ_1 = 0$, obtained by substituting $-\lambda W$ to the left factor of (3),

$$W \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} = \begin{pmatrix} 0 & P'Q_2(Q_2^T K Q_2 - \lambda I)^{-1} \end{pmatrix} \ ,$$

which, right-multiplied by $\begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^T$, yields:

$$W = P'Q_2(Q_2^T K Q_2 - \lambda I)^{-1} Q_2^T \ .$$

Finally, right-multiplying (4) by $K^{-1}Q_1$ gives

$$P'Q_1 - AR^T - WKQ_1 = 0 \ ,$$

whence the solution for A:

$$A = (P' - WK)Q_1 R^{-T} \ .$$

In our experiments, we used an open-source `C++` implementation of thin-plate spline, available at `http://elonen.iki.fi/code/tpsdemo/`.

2.5 Texture Mapping

Mapping textures from several views on the mesh can be achieved by several methods. Projecting all images on the mesh and doing some weighted averaging, as for example in [18], leaves some artifacts, such as ghosting. Other methods extract an atlas of the mesh, where each region of the mesh gets its texture from one unique view. The challenge is then to reduce seams visibility. The atlas can be computed by formulating the problem as a Markov Random Field energy minimization [19] and then masking the contrast changes between the view by multiband blending at the seams, generalizing work of Burt and Adelson [20]. This is the strategy presented in [21], which we use in our experiments.

3 Experiments

We first demonstrate the proposed algorithm using simulated wrong 3D geometry. The image data are courtesy of R. White *et al.*, who used them in [22]. The 3D geometry was estimated by Furukawa and Ponce [23]. To this 3D model, we apply artificially a translation in space before texturing by the algorithm of [21]. For each point on the mesh, the texture comes from one single view (the most

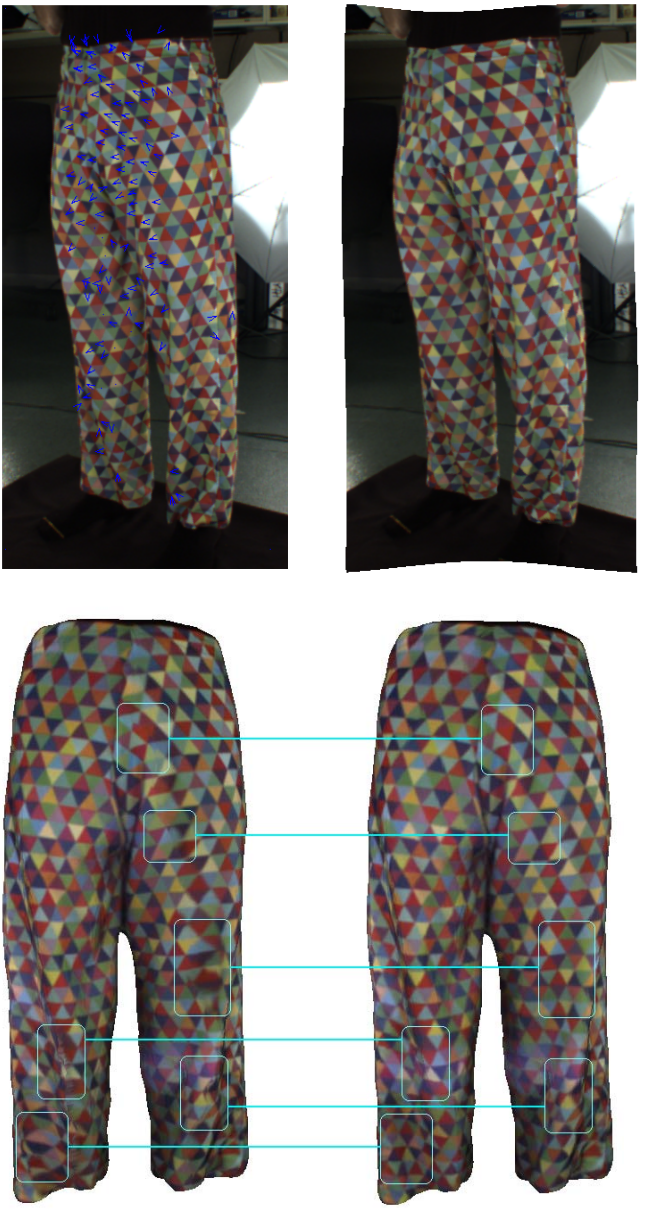


Fig. 2. Texturing on simulated imprecise mesh ([23] plus erroneous deformation). Top: Warping of image to adapt to the mesh. Top left: one original image with displacement vector of key points superimposed. Top right: the warped image using thin-plate spline approximation of this sampled vector field. Bottom: Multi-view texturing on imprecise mesh using [21]. Bottom left: texturing with original images. Bottom right: texturing with warped images.

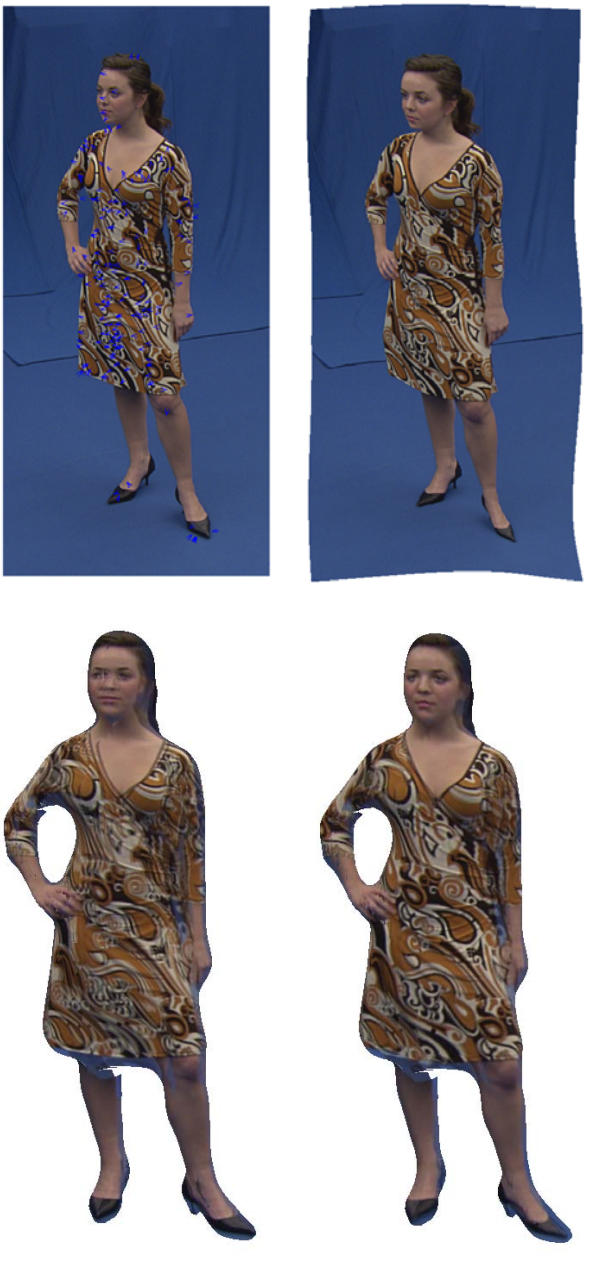


Fig. 3. Texturing on real imprecise mesh (visual hull from [24]). Top: Warping of image to adapt to the mesh. Top left: one original image with displacement vector of key points superimposed. Top right: the warped image using thin-plate spline approximation of this sampled vector field. Bottom: Multi-view texturing on imprecise mesh using [21]. Bottom left: texturing with original images. Bottom right: texturing with warped images.

frontal one), so that errors can only be seen at transitions from one view to another. Still the benefits of our mesh reprojection algorithm are visible in Fig. 2. The warping effect is most visible in the sides of the image.

In the next experiment, we use image data courtesy of J. Starck[2]. The 3D geometry was estimated by visual hull from silhouettes (using an implementation of the algorithm of Franco and Boyer [24]) and refined using Poisson surface reconstruction [25]. Texturing is done with the algorithm of [21] slightly modified to enhance errors: instead of selecting one pixel value, issued from the "best" view, to any point on the mesh, the average of the *two* best views is used. Only the 3 front views were used in the texturing process. This produces blur at misregistered points, otherwise the errors can only be observed at transitions between different cameras in the atlas, which is still noticeable but less striking. Notice that the original images produce artifacts on the arms and on the dancer's left hand, while the dress exhibits some wrong texture. Most of these problems are fixed by the warping, except on some part of the left arm.

4 Conclusion

When the multi-view reconstruction pipeline yields an imprecise mesh, we have shown how the input images themselves can be modified to mask the imprecisions in the rendering. Mapping these images as texture on the mesh limits the visible artifacts. Reversing the problem by changing the input (the images) to match the erroneous output (the mesh) does not allow better measurements, but if only a visually pleasing rendering is sufficient, as for example in motion capture for computer generated imagery, this technique provides a simple solution. The algorithm was demonstrated on simulated and real imprecise meshes. Extension of this work to dynamic multi-view stereo (3D+time) using similar algorithms will be investigated.

References

1. Atkinson, K. (ed.): Close Range Photogrammetry and Machine Vision. Whittles Publishing (2001)
2. Faugeras, O., Luong, Q.: The Geometry of Multiple Images. MIT Press, Cambridge (2001)
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
4. Ma, Y., Soatto, S., Koseck, J., Sastry, S.: An Invitation to 3-D Vision. Interdisciplinary Applied Mathematics, vol. 26. Springer, Heidelberg (2004)
5. Keriven, R., Faugeras, O.: Complete dense stereovision using level set methods. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 379–394. Springer, Heidelberg (1998)
6. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. The International Journal of Computer Vision 72(2), 179–193 (2007)

[2] `http://personal.ee.surrey.ac.uk/Personal/J.Starck/`

7. Vu, H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
8. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
9. Eisemann, M., De Decker, B., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., Sellent, A.: Floating Textures. Computer Graphics Forum (Proc. Eurographics EG 2008) 27(2), 409–418 (2008)
10. Tzur, Y., Tal, A.: Photogrammetric texture mapping using casual images. In: Proceedings of ACM SIGGRAPH (2009)
11. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.: Image-based plant modeling. In: Proceedings of ACM SIGGRAPH (2009)
12. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
13. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference, vol. I, pp. 384–393 (2002)
14. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 65 (2005)
15. Bookstein, F.: Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans. on PAMI 11(6), 567–585 (1989)
16. Wahba, G.: Spline Models for Observational Data. SIAM, Philadelphia (1990)
17. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins University Press, Baltimore (1996)
18. Bernardini, F., Martin, I., Rushmeier, H.: High-quality texture reconstruction from multiple scans. IEEE Trans. on Visualization and Computer Graphics 7(4), 318–332 (2001)
19. Lempistky, V., Ivanov, D.: Seamless mosaicing of image-based texture maps. In: Proc. of ICCV (2007)
20. Burt, P., Adelson, E.: A multiresolution spline with application to image mosaics. ACM Trans. on Graphics 2(4), 217–236 (1983)
21. Allène, C., Pons, J.P., Keriven, R.: Seamless image-based texture atlases using multi-band blending. In: Proc. of ICPR, pp. 1–4 (2008)
22. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. In: SIGGRAPH (2007)
23. Furukawa, Y., Ponce, J.: Dense patch models for motion capture from synchronized video streams. Technical report, Willow Technical report 02-07 (2007)
24. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: British Machine Vision Conference, vol. 1, pp. 329–338 (2003)
25. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Eurographics Symposium on Geometry Processing, pp. 61–70 (2006)

Semantic Classification in Aerial Imagery by Integrating Appearance and Height Information*

Stefan Kluckner, Thomas Mauthner, Peter M. Roth, and Horst Bischof

Institute for Computer Graphics and Vision,
Graz University of Technology, Austria
{kluckner,mauthner,pmroth,bischof}@icg.tugraz.at
http://www.icg.tugraz.at

Abstract. In this paper we present an efficient technique to obtain accurate semantic classification on the pixel level capable of integrating various modalities, such as color, edge responses, and height information. We propose a novel feature representation based on Sigma Points computations that enables a simple application of powerful covariance descriptors to a multi-class randomized forest framework. Additionally, we include semantic contextual knowledge using a conditional random field formulation. In order to achieve a fair comparison to state-of-the-art methods our approach is first evaluated on the MSRC image collection and is then demonstrated on three challenging aerial image datasets Dallas, Graz, and San Francisco. We obtain a full semantic classification on single aerial images within two minutes. Moreover, the computation time on large scale imagery including hundreds of images is investigated.

1 Introduction

Internet driven initiatives, like *Google Earth* and *Virtual Earth*, collect an enormous amount of aerial and satellite images in order to automatically construct 3D worlds of urban environments because of the demand for fast realistic 3D modeling, cartography, navigation support, etc. These location-aware applications on the internet push the development of efficient, accurate, and automatic technologies. The first step is to acquire high resolution images. In particular, the *Microsoft Ultracam* takes multi-spectral images in overlapping strips, resulting in high redundancy, which adheres every visible spot of urban environments from many different camera viewpoints. The high redundancy within the data enables methods for automatic height data generation [1] or full photo-realistic 3D modeling [2]. In contrast to photo-realistic 3D modeling, where the model consists of millions of triangles with fitted texture extracted from aerial images, we aim

* This work was supported by the Austrian Science Fund Projects W1209 and P18600 under the doctoral program Confluence of Vision and Graphics, by the FFG projects APAFA (813397) and AUTOVISTA (813395), financed by the Austrian Research Promotion Agency, and by the Austrian Joint Research Project Cognitive Vision under the projects S9103-N04 and S9104-N04.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 477–488, 2010.

for synthetic modeling, i.e., based on the information directly derived from the images to build a virtual model of a city. In addition, a synthetic model reduces the problem of privacy violations due to modeling the semantic interpretation instead of the realistic appearance.

Due to high variability in aerial imagery, automatic classification and semantic description still pose an unsolved task in computer vision. We aim to use appearance cues, such as color, edge responses, and height information for accurate semantic classification into five classes. For instance, using a combination of color and height data successfully separates the street regions from gray-valued roof tops or distinguishes between green areas and trees. Figure 1(a) shows corresponding color and height images, extracted from the dataset *San Francisco*. The classification of aerial images into several classes provides a semantic knowledge of the objects on ground and approves a specified post-processing to build up a semantic 3D world, where each object is modeled according to its obtained interpretation. A semantic description of a small sub-image is illustrated in Fig. 1(b).

Fig. 1. A pair of images extracted from the dataset *San Francisco* consisting of color and height information, and the corresponding semantic description of the sub-image (highlighted rectangles).

In [3], the authors proposed an appearance driven approach to exploit color and infrared data for initial classification. Several methods concentrate on extracting single object classes, e.g., buildings by integrating only LIDAR data [4] or height models [5]. The tight integration of 3D data into image classification, as additional information source, is still a new and upcoming field of research. Hoiem [6] extracted 3D information, such as surface orientation or vanishing lines, from single images to improve 2D object recognition. Recent approaches [7,8] include SfM to improve the interpretation in street side images. In this work, we exploit dense matching results [1] together with appearance features to obtain an accurate semantic interpretation.

Shotton et al. [9] proposed simple color value differences in a small neighborhood for initial semantic classification on the pixel level using a randomized forest (RF) classifier [10]. Schroff et al. [11] extended this approach by including multiple feature types for an improved RF classification. Strong low level feature representations, such as SIFT [12], histograms of oriented gradients [13], or

various types of filter responses [14,15,16] are widely used in appearance driven supervised classification. However, a compact combination of different feature cues is computationally very expensive. In addition, an integration into a common classification framework requires sophisticated techniques.

Thus, our work has three main contributions: To allow an efficient semantic classification, we first introduce a novel technique to obtain a powerful feature representation, derived from compact covariance descriptors [17] which is directly applicable to RF classifiers. Covariance matrices [17] can be efficiently computed and provide an intuitive integration of various feature channels. Since the space of covariance matrices does not form a Euclidean vector space [17], this representation can not be directly used for most machine learning techniques. To overcome this drawback, manifolds [18,17,19] are typically utilized, which, however, is computationally expensive. In contrast to calculating similarity between covariance matrices on Riemannian manifolds [18], we present a simple concept for mapping individual covariance descriptors to Euclidean vector space. The derived representation enables a compact integration of appearance, filter responses, height information etc. while the RF efficiently performs a multi-class classification task on the pixel level. Second, we introduce semantic knowledge by applying an efficient conditional random field (CRF) stage incorporating again several feature cues and co-occurrence information. To demonstrate the state-of-the-art performance we present quantitative results on the *Microsoft Research Cambridge* dataset MSRC-9 [15] by integrating visual appearance cues, such as color and edge information. Third, we apply our proposed method to real world aerial imagery, performing large scale semantic classification. We extend the novel feature representation with available height data as an additional cue and investigate the classification accuracy in terms of correctly classified pixels. Labeled training data, representing five annotated classes (building, tree, waterbody, green area and streetlayer), provides the input for the training process.

The remainder of this paper is structured as follows. Section 2 describes the derived covariance region descriptor in detail, illustrates the application to the RF framework, and also addresses the integration of the contextual constraints. Section 3 highlights the included feature cues and presents results on the MSRC-9 dataset and various real world aerial images. Finally, Sec. 4 concludes our work and gives an outlook on future work.

2 Semantic Classification

In this section we highlight the semantic classification pipeline including the feature representation based on covariance descriptors and *Sigma Points*, respectively, the straight forward application to a multi-class RF framework, and the CRF stage to handle the contextual constraints.

2.1 Approximated Covariance Representation

Tucel et al. [17,19] presented a compact feature representation based on covariance matrices for rapid object detection and classification. In fact, covariance

descriptors [17] provide a low-dimensional feature representation that simply integrates multiple feature channels, such as color, filter responses, height information, etc. and also exploits the correlation between them. The diagonal elements provide the variances of the feature attributes in one channel, whereas the off diagonal elements capture the correlation values between the different feature modalities. The statistics up to second order of N independent and identically distributed feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ can be represented by the sample mean $\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ and the sample covariance $\Sigma \in \mathbb{R}^{d\times d}$:

$$\Sigma = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T . \tag{1}$$

As shown by Tuzel et al. [17] the concept of integral images [16] can be applied to compute covariance descriptors on a rectangular image grid in constant time: Given a multi-channel feature image I of the dimension $w \times h \times d$, any $n \times m$ rectangular region $R \subseteq I$ can be represented by a $d \times d$ covariance matrix Σ. An extension of common integral images to higher dimensions incorporating additional tensor integral images, enables the computation of symmetric covariance matrices using the law of total variance. Implementation details can be found in [17].

Because of the missing symmetry requirement the space of covariance matrices is non-Euclidean [17]. Hence, standard machine learning methods, which require similarity computations can not be used directly. Instead of exploiting computationally costly manifolds [17,19] to obtain a valid covariance similarity measurement, we propose a technique to represent individual covariance matrices directly on Euclidean vector space. Julier et al. [20] proposed the unscented transform (UT), which approximates a single distribution by sampling instead of approximating an arbitrary non-linear function by mapping to manifolds [18]. The UT provides an efficient estimator for the probability distribution and was successfully applied to unscented Kalman filtering [21], where it overcomes the drawbacks of truncated (second order) Taylor expansions. In the d-dimensional case the UT relies on constructing a small set of $2d+1$ specific vectors $\mathbf{s}_i \in \mathbb{R}^d$, also referred to as *Sigma Points* [20]. We construct the set of *Sigma Points* as follows:

$$\mathbf{s}_0 = \mu \qquad \mathbf{s}_i = \mu + \alpha(\sqrt{\Sigma})_i \qquad \mathbf{s}_{i+d} = \mu - \alpha(\sqrt{\Sigma})_i, \tag{2}$$

where $i = 1 \ldots d$ and $(\sqrt{\Sigma})_i$ defines the i-th column of the required matrix square root $\sqrt{\Sigma}$. The scalar α defines a constant weighting for the elements in the covariance matrix and is set to $\alpha = \sqrt{2}$ for Gaussian input signals [20].

In contrast to Monte Carlo methods, where test vectors are selected at random, the construction of the *Sigma Points* can be seen as an efficient mapping of a specified set of test vectors $t_i \in \mathbb{R}^d$ that deterministically sample the

intersections of an unit hypersphere with a d-dimensional Cartesian coordinate system. Here, the mean vector $t_0 = \mu$ represents the origin. The computed statistics of these points $\mathbf{s}_i$ accurately capture the original information about Σ up to third order for Gaussian and up to second order for non-Gaussian inputs [21]. Figure 2 illustrates the specified sampling of the test vectors and the mapping for a simplified 2D case.

Since covariance matrices Σ are positive semi-definite by definition, we first perform a simple regularization $\Sigma = \Sigma + \epsilon\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and $\epsilon =$ 1e-6, to obtain symmetric positive definite matrices. Due to symmetry and positive definiteness of the regularized covariance matrices, the efficient Cholesky factorization can be applied to compute the matrix square root by decomposing $\Sigma = LL^T$. Then, L corresponds to $\sqrt{\Sigma}$ and is a lower triangular matrix. In principle any method for square root factorization can be used, however, the Cholesky decomposition requires the lowest mathematical operations yielding a complexity of $O(n^3/3)$.

The resulting feature representation $\mathcal{S}^k = \left(\mathbf{s}_0^k, \mathbf{s}_1^k \dots \mathbf{s}_{2d}^k\right)$ is obtained by concatenation of the *Sigma Points* and captures both, first and second order statistics, which are given by the mean and covariance information. Each of these generated vectors $\mathbf{s}_i^k \in \mathbb{R}^d$ describe Euclidean space, therefore, element-wise distance computations between corresponding samples of a given distribution are feasible. The construction pipeline for the set of *Sigma Points* is summarized in Algorithm 1.

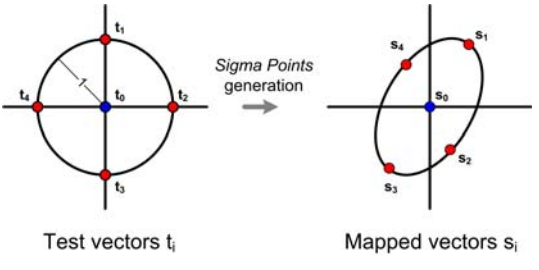


Fig. 2. The mapping of a fixed set of test vectors $\mathbf{t}_i$ to the *Sigma Points* $\mathbf{s}_i$ given in a second coordinate system, representing the original characteristics of the covariance matrix $\Sigma = LL^T$

The structure of this feature representation $\mathcal{S}^k$ perfectly fits the concept of randomized forest classifiers, where the learning and evaluation strategy is based on comparing randomly selected attributes of an available representation. Note that, since a reference representation is missing, similarity measurements, such as the Foerstner metric [18] are intractable to directly use in decision trees. In the following section we show how our representation can be applied straight forward to a RF framework.

Algorithm 1. Construction of our proposed feature representation based on *Sigma Points*.

Require: Mean vector μ^k and covariance matrix Σ^k
1: Perform a simple regularization $\Sigma^k = \Sigma^k + \epsilon \mathbf{I}$
2: Compute matrix square root $\Sigma^k = LL^T$
3: Compute s_i^k according to (2)
4: Construct the set of *Sigma Points* $\mathcal{S}^k = \left(\mathbf{s}_0^k, \mathbf{s}_1^k \dots \mathbf{s}_{2d}^k\right)$

2.2 Randomized Forest Framework

Randomized forests [10] have proven to give robust and accurate classification results for multi-class tasks [9,11,22]. An RF consists of an ensemble of binary decision trees, where the nodes of each tree include split criteria that give the direction of branching left and right down the tree until a leaf node is reached. Each leaf node l_i in a given maximal depth D contains a learned class distribution $P(\mathbf{c}|l_i)$. By averaging the decisions over all T trees in a forest the resulting accumulated probabilities yield an accurate class distribution $P(\mathbf{c}|L) = \frac{1}{T}\sum_{i=1}^{T} P(\mathbf{c}|l_i)$. To rapidly grow each tree of the forest, the split node criteria are learned using only a subset $\mathcal{S}'$ of the whole training data $\mathcal{S}$. For training a class label c_k is assigned to each feature representation $\mathcal{S}^k \in \mathcal{S}$. The learning proceeds from the root node top-down by tiling the available subset at each split node into left and right sets. Proposed splitting decisions in [9,22] are achieved by comparing two or multiple randomly chosen elements $\mathbf{s}_i^k$ and $\mathbf{s}_j^k$ of the given feature sample $\mathcal{S}^k$. In our implementation we follow a strategy similar to [22], randomly taking into account the correct corresponding dimension $a \in \{1 \dots d\}$ selecting two weighted elements i and j according to

$$\alpha \mathbf{s}_i^k(a) + \beta \mathbf{s}_j^k(a) = \begin{cases} > \gamma, & split\ left \\ \leq \gamma, & split\ right \end{cases} . \tag{3}$$

Here, α, β, and γ denote the greedy-optimized parameters that minimize the information gain with respect to the training labels [9,22]. We take the numbers of split node tests as suggested in [22]. Once the forest is trained, we evaluate the classifier at each pixel location by parsing down the extracted feature representation in the forest and accumulating the class distribution to obtain an overall probability map $P(\mathbf{c}|L)$.

2.3 Incorporating Contextual Information

Although our feature representation includes a spatial neighborhood of $n \times m$ implicitly, each pixel is classified independently. In this work we apply an efficient conditional random field (CRF) stage based on linear programming [23] to incorporate semantic contextual constraints yielding a smooth labeling of the final image classification. In addition, we include edge information into the four-connected graph to exactly capture the real object boundaries.

In order to obtain the contextual semantic information, we construct a dataset dependent co-occurrence matrix by counting the frequency of class labels in the training images within randomly chosen rectangular sub-windows [9]. The frequency counts can be performed quickly on single images using integral structures [16]. Furthermore, we follow the concept of [24] to compute a normalized co-occurrence matrix $\theta(c_i, c_j)$ representing the pairwise semantic contextual information of the grid nodes i and j. The application of the CRF allows us to include the posterior class distribution $P(\mathbf{c}|L)$, the likelihood co-occurrence matrices $\theta(\cdot)$ and an edge penalty function to preserve the object boundaries. Given a four-neighborhood connected image I we define an energy with respect to the class labels according to

$$E(\mathbf{c}) = \sum_i D(c_i) + \sum_{i,j} w_{ij} V(c_i, c_j), \tag{4}$$

where $D(c_i)$ denotes the data term, including the unary potentials according to $D(c_i) = -\log(P(c_i|L))$ at grid node i. The pairwise class potentials are computed according to $V(c_i, c_j) = -\log(\theta(c_i, c_j))\delta(c_i \neq c_j)$ and include the semantic knowledge. The weight w_{ij} describes an edge penalty term between the nodes i and j. Following the concept suggested in [11], where the authors used color distance computations to capture the object boundaries, we exploit the height information in case of the aerial images. Thus, the weight is constructed with $w_{ij} = \exp\left(-\lambda \|h_i - h_j\|^2\right)$, where h_i, h_j are the height values at the neighboring graph nodes. λ defines a factor and is learned while training. In this work we apply the strategy of Komodakis et al. [23] to minimize the energy defined in (4). In the experimental evaluation we present overall results incorporating semantic contextual information into the classification pipeline.

3 Experimental Evaluation

Due to efficient computation of our region based covariance representation, we exploit several feature cues incorporating a small spatial neighborhood. First, we construct the required integral images to compute the covariance descriptors including the feature cues, such as color channels, first derivatives in x and y direction, and the height values. Then, the feature instances are constructed according to our proposed concept (see Sec. 2.1). The collected samples provide the representation for training and testing. In the following, we first evaluate our classification pipeline on the standard MSRC-9 [15] evaluation dataset. By integrating appearance and height information we illustrate the application to real world aerial imagery and investigate large scale capability.

3.1 Experiments on MSRC-9 Dataset

In our first experiment we use the MSRC-9 dataset with nine on the pixel level labeled classes to provide results for a comparison to state-of-the-art approaches [11,25]. For the training and the testing procedure we randomly split

the dataset including a total number of 240 images, 120 training and 120 test images. The training samples, consisting of the set of *Sigma Points* $\mathcal{S}$ and a target label vector c, are regularly collected on a 5×5 grid with a small spatial neighborhood of $n = m = 21$ pixels. The corresponding label is extracted by considering the available ground truth images. Confirming the observation in [9], the CIELab color space generalizes better than raw RGB values. The first derivatives are computed on the L-channel. We apply small synthetic affine distortions to the training images capturing an invariance to shape deformations [9]. In addition, we extend the test images, according to the spatial neighborhood, to obtain class probabilities at the image borders. Due to randomness of our approach, we repeat the experiment 20 times independently to obtain meaningful averaged classification rates. In this work, we choose a relatively small size of the forest ($T = 15$ trees and a maximum depth of $D = 10$) to provide both, efficiency in testing and classification accuracy.

Our pixel-wise RF classification returns rates of 64.2% using only color and 71.1% integrating both, color and derivative information. The feature representation $\mathcal{S}_i$ at a pixel i integrating only color yields a concatenated vector with a dimension of 21 attributes, while an extension to include derivatives increases the size to 55. In [11] rates of 72.2% are given for only incorporating color information, however using a forest with 20 trees each with a maximum depth of 20. Running the full classification cue including the CRF stage achieves an average classification performance of 84.2%, while in [11] and [25] rates of 87.2% and 84.9% are reported, respectively. Running the full classification cue, consisting of the feature extraction, the evaluation of the classifier at each pixel and the integration of semantic knowledge using the CRF stage, on a single image requires less than 2 seconds on a standard single core PC. Figure 3 depicts a selection of semantic classification results on the MSRC-9 dataset.

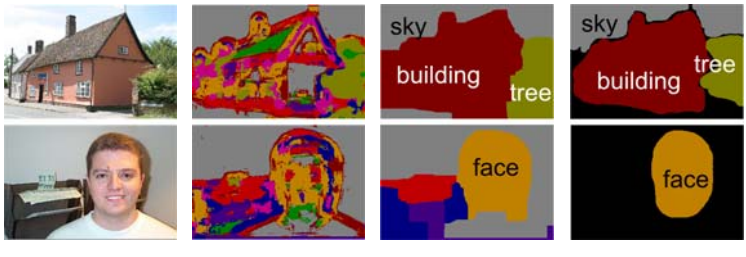


Fig. 3. A selection of results on the MSRC-9 dataset. From left to right: color images, pure pixel-wise classifications, final result using RF and CRF and ground truth labeling.

Considering the results of our first experiment on the MSRC dataset, we conclude that an integration of color and derivatives enhances the classification rates significantly. Including semantic contextual information, using an efficient CRF stage further improves the results. The comparison shows that our throughout simple approach is competitive with existing methods [11,25].

3.2 Experiments on Aerial Images

The second experiment evaluates the classification pipeline on huge real world aerial images. We apply separately trained RF classifies to single aerial images performing a semantic classification into five classes (building, tree, waterbody, green area and streetlayer) on the level of pixels.

In this work, we perform experiments on three different datasets, generated by the *Microsoft Ultracam*: The dataset *Dallas* includes large building structures and gray valued areas, *Graz* shows a colorful characteristic with challenging building blocks, and the images of *San Francisco* have mainly suburban appearance. The color images have a dimension of $11.5K \times 7.5K$ pixels and provide a ground sampling distance (GSD) of 8 cm (*Graz*) and 15 cm (*Dallas*, *San Francisco*). Due to high redundancy a dense matching process [1], taking into account three adjacent images, yields range images representing the surface model. Subtracting the surface model from the extracted ground plane using, e.g., [26] produces the relative height information that is directly applicable to our classification procedure as an additional feature channel. The dimension of the resulting feature vector increases to 78, if CIELab color, derivative, and height information are integrated. Figure 1(a) shows a pixel synchronous pair of a color and the corresponding height image. For each dataset we independently label three images providing the training labels on the pixel level. Additionally, we generate two non-overlapping images as ground truth data for testing. Similar to the MSRC training process the target labels are then collected taking into account these training maps.

In case of the aerial images we compute our feature representation integrating the color, texture, and height information and train an RF classifier with 15 trees and maximum depths of 10 separately for each dataset. The dimension of the spatial neighborhood is set according to the datasets GSD with $n = m = 2(50/GSD + 1)$. The trained RFs are evaluated at each pixel location using a fourth of the full image resolution. The obtained classification rates for the three datasets are summarized as confusion matrices in Fig. 5. A combination of color, derivatives, and height information results in averaged rates of 92% (*Dallas*), 93% (*Graz*), and 88% *San Francisco*. For instance, using only color and derivative cues yields low classification accuracies of 79% (*Dallas*), 78% (*Graz*), and 73% *San Francisco*.

Figure 4 depicts a full semantic classification including the CRF stage of a single image taken from the *Graz* dataset. The feature extraction and pixel-wise classification of a single aerial image of *Graz* covering an area of approximately 0.5 km^2 requires about 35 seconds, the CRF stage increases the computation time to approximately 80 seconds. This scales to an overall computation of about 1.5 hours on a standard PC given a complete dataset, e.g., *Graz* with 155 images. Note that for a full dataset processing the CRF stage can be applied to a fused classification result instead of using the per-image classification, which speeds up the computation drastically. Figure 6 illustrates a selection of classified subimages extracted from full processing steps.

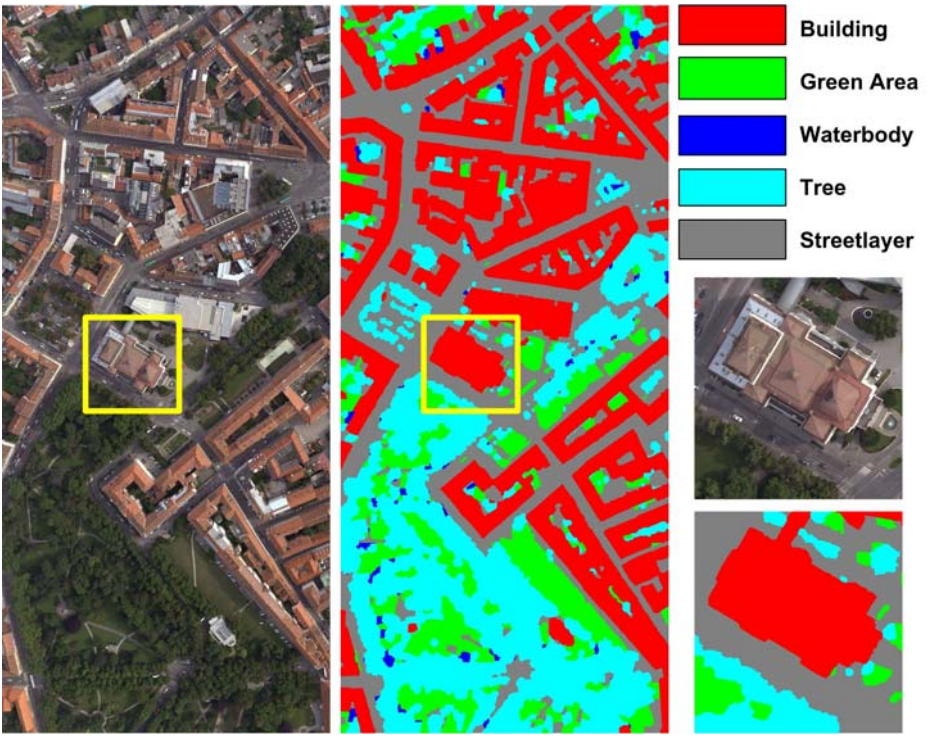
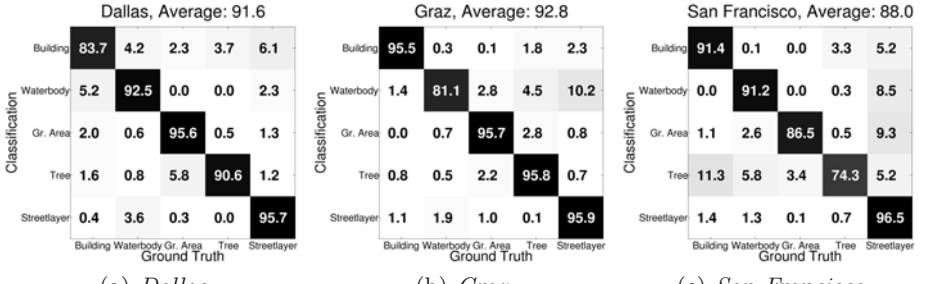


Fig. 4. Full semantic classification of a single image taken of the dataset *Graz*. The image provides a ground sampling of 8 cm and covers an area of approximately 0.5 km^2.



(a) *Dallas* (b) *Graz* (c) *San Francisco*

Fig. 5. Computed confusion matrices on the three aerial image datasets. We obtain classification rates of approximately 90% on the three challenging datasets. The low gray-valued buildings in *Dallas* are sometimes mixed with the streetlayer class which can be caused by inaccurate terrain models. Due to similar spectral ranges small shadow regions in the streets are classified as waterbody in *Graz*. Many small trees inside of courtyards and the hilly terrain in *San Fransisco* explain the relatively low classification rate for trees.

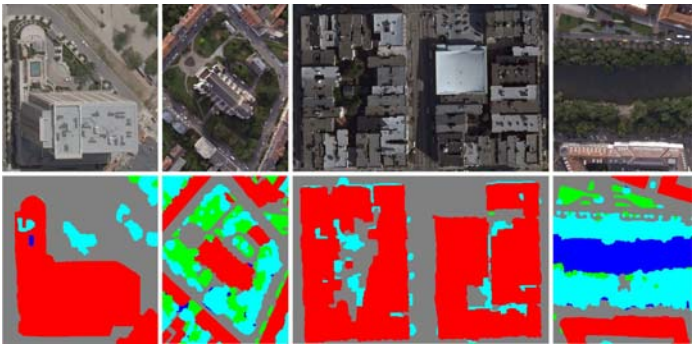


Fig. 6. Representative sub-images extracted from full semantic classification results. From left to right: a hotel complex with a pool/trees on the top in *Dallas*, a church surrounded with vegetation in *Graz*, a typical building block of *San Fransisco*, and a detail showing a river in *Graz*.

4 Conclusion

This work has proposed an efficient approach for semantic classification of images by integrating multiple types of feature modalities, such as appearance, edge responses, and height information. We presented a novel feature representation based on covariance matrices and *Sigma Points*, respectively, that can be directly applied to multi-class RF classifiers. By including contextual information using a CRF stage, we achieved an accurate semantic description of test images on the pixel level. We performed experiments on the MSRC dataset and on huge real world aerial images and demonstrated accurate classification results with low computational costs. Further work will investigate the influence of additional data cues, like infrared and pan-chromatic images, on the classification quality. In addition, we work on exploiting the redundancy by fusing multiple image classification results of different viewpoints.

References

1. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proceedings ICPR (2006)
2. Zebedin, L., Bauer, J., Karner, K., Bischof, H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 873–886. Springer, Heidelberg (2008)
3. Zebedin, L., Klaus, A., Gruber-Geymayer, B., Karner, K.: Towards 3D map generation from digital aerial images. International Journal of Photogrammetry and Remote Sensing 60, 413–427 (2006)
4. Matei, B.C., Sawhney, H.S., Samarasekera, S., Kim, J., Kumar, R.: Building segmentation for densely built urban regions using aerial LIDAR data. In: Proceedings CVPR (2008)

5. Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M.: Automatic building extraction from dems using an object approach and application to the 3D-city modeling. International Journal of Photogrammetry and Remote Sensing 63(3), 365–381 (2008)
6. Hoiem, D., Stein, A., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: Proceedings ICCV (2007)
7. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
8. Cornelis, N., Leibe, B., Cornelis, K., Van Gool, L.: 3D city modeling using cognitive loops. In: International Symposium on 3DPVT (2006)
9. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proceedings CVPR (2008)
10. Breiman, L.: Random forests. Machine Learning, 5–32 (2001)
11. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: Proceedings BMVC (2008)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings CVPR (2005)
14. Shotton, J., Winn, J., Rother, C., Crimininsi, A.: Textonboost: Joint appearance, shape and conext modeling for muli-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
15. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proceedings ICCV (2005)
16. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision (2002)
17. Tuzel, O., Porikli, F., Meer, P.: Learning on lie groups for invariant detection and tracking. In: Proceedings CVPR (2008)
18. Foerstner, W., Moonen, B.: A metric for covariance matrices. Technical report, Department of Geodesy and Geoinformatics, Stuttgart University (1999)
19. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: Proceedings CVPR (2007)
20. Julier, S., Uhlmann, J.K.: A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford (1996)
21. Julier, S., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: International Symposium of Aerospace/Defense Sensing, Simulations and Controls (1997)
22. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: Proceedings ICCV (2007)
23. Komodakis, N., Tziritas, G.: Approximate labeling via graph cuts based on linear programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(8), 1436–1453 (2007)
24. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: Proceedings ICCV (2006)
25. Verbeek, J., Triggs, B.: Scene segmentation with CRFs learned from partially labeled images. In: Advances in NIPS (2007)
26. Champion, N., Boldo, D.: A robust algorithm for estimating digital terrain models from digital surface models in dense urban areas. In: Proceedings ISPRS Commission 3 Symposium, Photogrammetric Computer Vision (2006)

Real-Time Video Matting Based on Bilayer Segmentation

Viet-Quoc Pham[1], Keita Takahashi[2], and Takeshi Naemura[1]

[1] Graduate School of Information Science and Technology, The University of Tokyo
[2] IRT Research Initiative, The University of Tokyo,
Hongo 7–3–1, Bunkyo-ku, Tokyo, 113–8656 Japan
{viet,keita,naemura}@nae-lab.org

Abstract. Most current video matting methods perform off-line with a high calculation cost and require many user inputs for multiple key frames. In this paper, we present an online video matting method that runs in real-time based on bilayer segmentation. For the first step of the method, we introduce an accurate bilayer segmentation method for extracting the foreground region from the background using color likelihood propagation. For the second step, we perform alpha-matting based on the segmentation result. To enable real-time processing, we modify the conventional Bayesian matting method by using down-sampling and smart initialization, which increase the calculation speed by 5 times while maintaining the quality. Experimental results on various test sequences show the effectiveness of our method.

1 Introduction

Video matting is a technique for extracting an alpha matte and foreground from a video sequence, which has been applied widely in commercial television and film production. Different from bilayer segmentation which simply divides a video into two regions (foreground and background), the matting process extracts the foreground and also produces a fractional opacity at every pixel. Combined with the foreground color, the matte allows users to modify the background or to composite the foreground onto a new background. Fractional opacities are especially effective for fuzzy objects like hair, feathers, etc..

This matting problem can be solved easily by using some special assumptions. The most common method for video matting is the chroma keying, which is often used in broadcasting. In this method, foreground elements are filmed in front of a solid color background, which usually requires tightly controlled studio environments. Another common method is difference matting [1], which produces the opacity based on the background subtraction result. Its disadvantage is that the background must be kept fixed during the filming time.

Chuang et al. introduced an effective method for video matting from general scenes [2]. They applied a method called "Bayesian matting" [3] to each frame of the sequence. Bayesian matting begins with a user-supplied trimap, i.e., a segmentation of the scene into three regions: "definitely foreground", "definitely

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 489–501, 2010.

background", and "unknown". By collecting nearby foreground and background statistics, the opacity, as well as foreground and background colors, can be estimated at each pixel in the unknown region. For processing the whole sequence, the trimaps are interpolated across the video volume using forward and backward optical flows. This matting process performs off-line with a high calculation cost, and users must supply correct trimaps for every 10 frames.

In this paper, we address a challenging problem of real-time video matting: extracting a matte from a general video sequence in a real-time process. As far as we know, there are no video matting methods that can run in real-time for general videos with the quality comparable to Bayesian matting. However, real-time video bilayer segmentation, a technique that segments each frame into foreground and background, has been developed for the last five years [4,5,6]. These methods require some manual procedures, but they can perform almost automatically. For example, in [4,5], the first frame is segmented by employing background subtraction or an interactive segmentation method, then the remaining frames are processed automatically. In this paper, we describe a video matting approach that builds upon a real-time bilayer segmentation method and a real-time alpha matting process. This combination can not only save the need for user interaction, but also realize real-time processing.

In our approach, we segment each frame of the input video into foreground and background, then apply matting process to the pixels around the boundary. The entire procedure can be done in real-time. Our method is based partly on the segmentation method from our previous work [4] and the Bayesian matting method [3], but our main contributions lie in the improvements of these methods. For segmentation, we introduce a novel method for propagating color likelihood based on motion vectors. We can attain higher segmentation accuracy than [4], while keeping the processing speed. For matting, we apply down-sampling and smart initialization, which increase the calculation speed by 5 times compared to [3] while keeping the quality. Combining the proposed improvements, we can realize real-time video matting from general scenes with high quality.

The rest of this paper is organized as follows: Section 2 presents the segmentation algorithm. The real-time matting technique is stated in Section 3. Section 4 discusses experimental results. Finally, Section 5 concludes the paper.

2 Segmentation Algorithm

In this section, we describe the first half of our proposed method: the bilayer segmentation step. Our segmentation step supposes that the first frame has already been segmented into two layers (foreground and background) by using some methods, then it performs segmentation for the remaining frames automatically. Here, we employ two methods for segmenting the first frame: interactive segmentation and background subtraction, depending on the type of the target video. In case of existing videos like "Foreman" (see Fig. 8), we use a user interface, based on the Interactive GraphCut algorithm [7], to perform segmentation with very simple user interaction (see Fig. 1). In case of live videos where backgrounds are

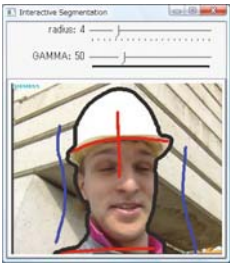


Fig. 1. Interactive segmentation tool

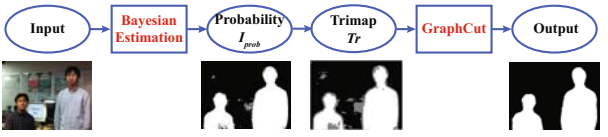


Fig. 2. Our framework for segmentation is based on [4]

kept fixed before filming, we can apply background subtraction to the first frame, making the whole segmentation process smoother without any user interactions.

In this paper, we employ the segmentation framework from our previous work [4], and propose a novel method for propagating color likelihood based on motion vectors. In [4], thermal vision images were employed in addition to color images to attain high quality segmentation results. But for the purpose of this paper, in which only color images are available, the segmentation method of [4] is insufficient. In this paper, by introducing a novel color likelihood propagation technique, we can improve the segmentation quality while maintaining the real-time calculation speed.

Before moving to the next section, it is better to briefly explain the GraphCut optimization algorithm [7]. Let Seg be a segmentation of an image where $Seg(p)$ takes F (foreground) or B (background) for each pixel p. We define the energy function by

$$\mathrm{E}(Seg) = \mathrm{Data}(Seg) + \lambda \mathrm{Sth}(Seg) \tag{1}$$

where the data term $\mathrm{Data}(Seg)$ evaluates the pixel-wise costs, and the smoothness term $\mathrm{Sth}(Seg)$ evaluates the inter-pixel costs. λ is a weighting coefficient. This function can be minimized by the min-cut algorithm [8], and the optimization solution should produce a good segmentation because it considers the balance between the region property and the boundary property of the segments.

2.1 Framework

Our framework for the segmentation step is based on the segmentation algorithm of [4]. The detailed algorithm should be referred to from the original paper [4]. A flow-chart is shown in Fig. 2. The original method requires both color and thermal vision images as the inputs. But in this paper, we employ only color images, so that the method can be adapted to general videos.

First, for each pixel p, we find the probability $I_{prob}(p)$ that p is classified to the foreground, by calculating a conditional probability using Bayes' formula

$$I_{prob}(p) = P(F|\boldsymbol{C}_p) = \frac{P(\boldsymbol{C}_p|F)P(F)}{P(\boldsymbol{C}_p|F)P(F) + P(\boldsymbol{C}_p|B)P(B)} \tag{2}$$

$\boldsymbol{C}_p$ is the color vector of pixel p, and F and B denote the foreground and background. $P(*)$ represents the probability of "$*$". The likelihoods $P(\boldsymbol{C}_p|F)$ and

$P(\boldsymbol{C}_p|B)$ are retrieved from the foreground and background color histograms. The prior $P(F)$ is calculated from the segmentation result of the previous frame, using spatio-temporal coherence. We should notice that the prior depends on the pixel position, while the likelihoods are independent of that.

Next, we create a trimap, $Tr(p)$, which takes one of the three values $\{F$ (foreground), B (background), and U (unknown)$\}$, based on the value of $I_{prob}(p)$

$$\begin{cases} if(I_{prob}(p) < \epsilon): & Tr(p) = B \\ if(I_{prob}(p) > 1 - \epsilon): & Tr(p) = F \\ otherwise: & Tr(p) = U \end{cases}$$

where ϵ is a small real value. GraphCut optimization is then performed on the unknown region. The energy function for GraphCut optimization is defined as

$$\mathrm{Data}(Seg) = \sum_{p \in U} -\log(P(\boldsymbol{C}_p|Seg(p))) + \mu \sum_{p \in U} -\log(I_{prob}(p)) \tag{3}$$

$$\mathrm{Sth}(Seg) = \sum_{(p,q) \in N} [Seg(p) \neq Seg(q)] \frac{e^{-\|\boldsymbol{C}_p - \boldsymbol{C}_q\|^2/(2\sigma^2)}}{\| \mathrm{dist}(p,q) \|} \tag{4}$$

where N represents the set of adjacent pixel pairs, and $\mathrm{dist}(p,q)$ denotes Euclidean distance between pixels p and q. σ can be estimated as "camera noise". The probability map I_{prob} is employed in Eq. (3), to append the spatial-temporal coherence information to the data term.

2.2 Color Likelihood Propagation

In this section, we introduce a likelihood propagation method to the segmentation framework in Section 2.1. By including this new technique, we can attain more stable segmentation results while maintaining the real-time speed.

In most video segmentation researches [4,5,6], color likelihood is represented using a global probability model for the whole image. For example, color likelihood of each pixel was estimated from color histograms in [4,6], and probability density functions in [5]. However, based on many experimental results, we found that such global color models are the main cause of segmentation errors. Figure 3(a) shows the 173th frame from the sample sequence "Video Chatting", which was used in [4]. The segmentation result by directly applying the algorithm in Section 2.1 is shown in Fig. 3(b). In this frame, the black color appears equally in both foreground and background regions. The black laptop appearing near the man (pointed by the arrow) has almost the same color likelihoods for foreground and background, because these likelihoods are estimated globally. As a result, this laptop was misclassified to the foreground.

To fix the above problem, we introduce a novel technique which propagates the local color likelihoods based on motions between two adjacent frames. First, in the current frame, we detect a set of reliable feature points G to track, employing the algorithm of [9]. Then, by using a variation of the Lucas-Kanade optical flow tracker based on image pyramids [10], for each feature point $p_i \in G$, we find an optical flow vector v_i which matches p_i with a correspondent point q_i in the

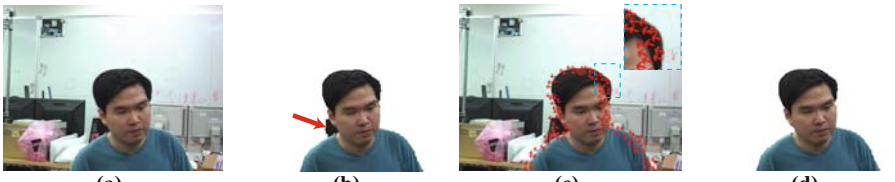


Fig. 3. Color likelihood propagation

previous frame (see Fig. 3(c)). Furthermore, we apply the RANSAC algorithm to remove error vectors. The purpose of this process is to obtain a local color likelihood of each pixel p_i from the known color distribution of q_i based on the propagation.

We first assume that the foreground/ background in the previous frame are correctly segmented. Then, for each point q_i, we build two color distribution models, for foreground and background, using the known foreground/ background colors within this pixel's neighborhood N_i. For details, the mean color $\overline{\boldsymbol{F}_i}$ and covariance $\Sigma_{\boldsymbol{F}_i}$ of the foreground color distribution are calculated as follows:

$$\overline{\boldsymbol{F}_i} = \frac{1}{W_i} \sum_{q \in N_i, Seg(q)=F} w_q \boldsymbol{C}_q \tag{5}$$

$$\Sigma_{\boldsymbol{F}_i} = \frac{1}{W_i} \sum_{q \in N_i, Seg(q)=F} w_q (\boldsymbol{C}_q - \overline{\boldsymbol{F}_i})(\boldsymbol{C}_q - \overline{\boldsymbol{F}_i})^T \tag{6}$$

where W_i is the regularization constant and $\boldsymbol{C}_q$ is the color vector of pixel q. The contribution of each neighborhood pixel q is weighted with a spatial Gaussian fall-off w_q with $\sigma = 8$ to increase the contribution of nearby pixels over those that are further away. The foreground color likelihood of pixel q_i is then estimated locally from the oriented elliptical Gaussian distribution

$$L(q_i|F) = (\boldsymbol{C}_{q_i} - \overline{\boldsymbol{F}_i})^T \Sigma_{\boldsymbol{F}_i}^{-1} (\boldsymbol{C}_{q_i} - \overline{\boldsymbol{F}_i})/2 \tag{7}$$

This foreground color likelihood is then propagated to pixel p_i: $L(p_i|F) = L(q_i|F)$. The background color likelihood $L(p_i|B)$ is found in the same way. The propagated likelihoods are then added to the data term in Eq. (3)

$$\begin{aligned} \text{Data}(Seg) = &\sum_{p \in U} -\log(P(\boldsymbol{C}_p|Seg(p))) + \mu \sum_{p \in U} -\log(I_{prob}(p)) \\ &+ \nu \sum_{p_i \in G} \log(L(p_i|Seg(p_i))) \end{aligned} \tag{8}$$

The likelihood of each feature point is composed of both global and local color distributions. The remaining pixels besides the feature points are not affected by the local color model, but by performing GraphCut optimization, they will be dragged by the nearby feature points, resulting in correct segmentations. Figure 3(d) shows the segmentation results by including the color likelihood propagation method. Some errors appearing in Fig. 3(b) are completely corrected.

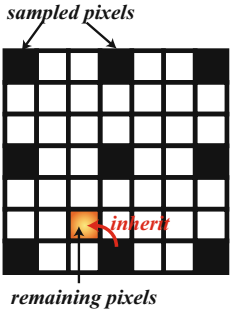


Fig. 4. Down-sampling

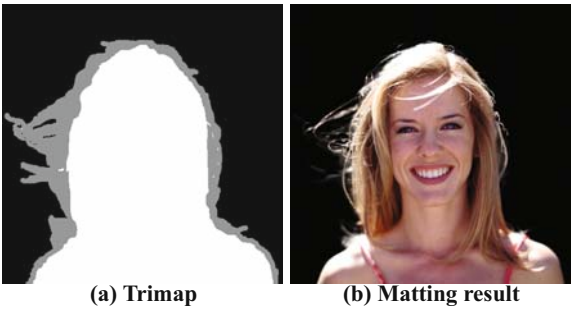


Fig. 5. A natural matte sample

3 Matting Algorithm

In the second step of our method- the matting step, we perform alpha matting based on the segmentation result from Section 2. This alpha matting process, which produces a fractional opacity at every pixel, is performed on a morphological strip around the object contour with 15 pixel width. In our method, we improve the famous Bayesian matting [3] by introducing down-sampling and smart initialization, for the purpose of realizing real-time processing. We can increase the calculation speed by 5 times while keeping the matting quality.

We first explain the original method, then describe our improvements later.

3.1 Bayesian Matting

Bayesian matting [3] method formulates the matting problem in a well-defined Bayesian framework, then estimates its parameters using maximum a posteriori (MAP) method. In this section, we only give a brief explanation of the method. The detailed algorithm should be referred to from the original paper.

The method assumes that the source image $\boldsymbol{C}$ is a composite of two images $\boldsymbol{F}$ and $\boldsymbol{B}$ (foreground and background) with opacity channel α. These values should satisfy the compositing equation in each pixel:

$$\boldsymbol{C} = \alpha \boldsymbol{F} + (1 - \alpha)\boldsymbol{B} \tag{9}$$

This method takes a user-supplied trimap (see Fig. 5(a)) as the input. A trimap for matting (which should be distinguished from the trimap for segmentation in Section 2.1), is a segmentation of the scene into three regions: "foreground" ($\alpha = 1$), "background" ($\alpha = 0$), and "unknown", in which the parameters $\alpha \mathrm{C} \boldsymbol{F} \mathrm{C} \boldsymbol{B}$ for "unknown" pixels need to be estimated. The estimation is performed by maximizing the following posteriori probability, based on the given color $\boldsymbol{C}$

$$P(\boldsymbol{F}, \boldsymbol{B}, \alpha | \boldsymbol{C}) = \frac{P(\boldsymbol{C}|\boldsymbol{F}, \boldsymbol{B}, \alpha) P(\boldsymbol{F}) P(\boldsymbol{B}) P(\alpha)}{P(\boldsymbol{C})} \tag{10}$$

where $P(\boldsymbol{C}|\boldsymbol{F}, \boldsymbol{B}, \alpha)$ is estimated as the distance between $\boldsymbol{C}$ and the mix of $\boldsymbol{F}$ and $\boldsymbol{B}$ (i.e. by the norm of the difference of the left and right hand sides of

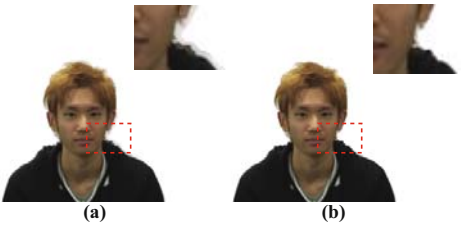


Fig. 6. Matting results after 10 loops: (a)Original method, (b)Our improvement

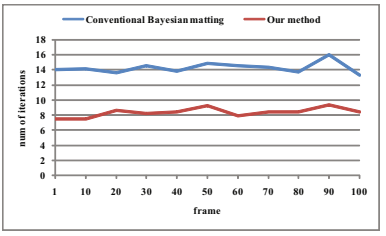


Fig. 7. Number of iterations for convergence

Eq. (9)). $P(\boldsymbol{F})$ and $P(\boldsymbol{B})$, which are assumed to follow the Gaussian distribution, are formulated by estimating their distribution parameters (the means $\overline{\boldsymbol{F}}$, $\overline{\boldsymbol{B}}$ and covariances $\Sigma_{\boldsymbol{F}}$, $\Sigma_{\boldsymbol{B}}$). $P(\alpha)$ is ignored and $P(C)$ is constant. To maximize the posteriori (Eq. (10)), partial derivatives with respect to α and $(\boldsymbol{F}, \boldsymbol{B})$ are set to 0, resulting in the following simultaneous equations

$$\begin{cases} \begin{bmatrix} \Sigma_{\boldsymbol{F}}^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & \Sigma_{\boldsymbol{B}}^{-1} + I(1-\alpha)^2/\sigma_C^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \Sigma_{\boldsymbol{F}}^{-1}\overline{\boldsymbol{F}} + \boldsymbol{C}\alpha/\sigma_C^2 \\ \Sigma_{\boldsymbol{B}}^{-1}\overline{\boldsymbol{B}} + \boldsymbol{C}(1-\alpha)/\sigma_C^2 \end{bmatrix} \\ \alpha = \dfrac{(\boldsymbol{C}-\boldsymbol{B}).(\boldsymbol{F}-\boldsymbol{B})}{||\boldsymbol{F}-\boldsymbol{B}||^2} \end{cases} \tag{11}$$

σ_C is the standard deviation of color $\boldsymbol{C}$. The above Eq. (11) is solved using an iterative method, by repeating two procedures: update $(\boldsymbol{F}, \boldsymbol{B})$ assuming α is fixed, and update α assuming $(\boldsymbol{F}, \boldsymbol{B})$ are fixed. α is initialized by the mean of known α values over the neighborhood.

3.2 Proposed Method

The above algorithm produces good results, but its calculation cost is too high. In this section, we propose two improvements to this algorithm, for the purpose of realizing real-time processing.

The first improvement is down-sampling, which aims to reduce calculation cost for estimating color likelihoods. In the original method, parameters of color models $P(\boldsymbol{F})$ and $P(\boldsymbol{B})$ (the means $\overline{\boldsymbol{F}}$, $\overline{\boldsymbol{B}}$ and covariances $\Sigma_{\boldsymbol{F}}$, $\Sigma_{\boldsymbol{B}}$) are estimated for every pixel based on the distributions of known foreground/ background colors within the neighborhood. Based on the fact that two adjacent pixels have almost the same neighborhoods, we can omit the process of parameter estimation for a pixel by inheriting results from the adjacent pixel. In our first improvement, we estimate parameters for only the sampled pixels where row and column indices are both multiples of 3, and the parameters for each remaining pixel is inherited from its nearest sampled pixel (see Fig. 4). Figure 5(b) shows the matting result of a natural image by including our first improvement. We attain almost the same quality as the result of [3] with only 1/9 calculation cost for the parameter estimation.

Table 1. Calculation time for convergence

	Original method [3]	Proposed method
Likelihood estimation	278ms	33ms
Iteration process	72ms	38ms
Total	352ms	72ms

In our second proposal, we improve the method for estimating the initial value of α, with the objective of reducing the required number of iterations of Eq. (11) for finding the convergent solutions. We found that the convergence speed depends mostly on the initialized α value. The bad choice of this value will make the process longer, and even lead to the local-minimum problem. In the original Bayesian matting, α is initialized by the mean of known α values over the neighborhood. Meanwhile, in our improvement, we estimate the initial value of α using the probability model in Eq. (2). Here, likelihoods $P(\boldsymbol{C}_p|F)$ and $P(\boldsymbol{C}_p|B)$ are calculated from the given trimap. Prior $P(F)$ is estimated from the segmentation result of our previous step, by filtering the segmentation mask with a Gaussian operation. The posterior $P(F|\boldsymbol{C}_p)$ is then used to initialize α. This value is clearly better than the mean value because it considers not only the spatial coherence, but also the color distribution.

Figure 6 shows the matting results after 10 loops of iteration, using the original method (a) and our two improvements (b). To this period, our method has been convergent while the conventional method needs many more loops. In a different evaluation, we set the convergence point as the period when the variance $|\Delta\alpha| < 0.001$, then measure the required number of iterations for convergence (see Fig. 7). The sequence "Headshake" (see Fig. 13) was employed here. Compared to the original method, we need only a half number of iterations for convergence. The calculation time for likelihood estimation and the iteration process is shown in Tab. 1. By introducing the two improvements, we can speed up the likelihood estimation step by 9 times and the iteration process by twice. Our entire matting process is 5 times faster than the original method.

4 Experimental Results

We performed experiments with various sample sequences to show the effectiveness of our method. We first evaluate the effectiveness of each step, segmentation and matting, individually. Then the combination of the two steps, which can realize real-time video matting, is evaluated in the end of this section.

4.1 Evaluation of Segmentation

We compared our segmentation method with three conventional methods [4,5,6]. For a fair comparison, we reused sample videos from these research papers, as the test sequences for our experiments. Because the source code of [5] is not available, we just referred to the experimental results stated in this paper. In

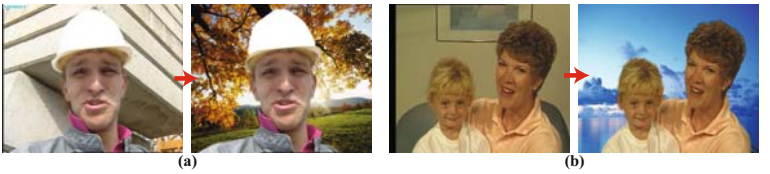


Fig. 8. Segmentation results of sequences: (a)"Foreman", (b)"Mother and Daughter"

Fig. 9. Segmentation results of the sequence"Video Chatting"[4]

case of [4] and [6], the source codes are available, so we performed experiments using them. Some segmentation results using our method are shown in Fig. 8 and Fig. 9. In Fig. 8, we present results for two MPEG-4 sequences "Foreman" and "Mother and Daughter" (352×288 pixels) which were used by [5]. The sequence in Fig. 9 is "Video Chatting" (320×240 pixels) which was employed by [4]. Our segmentation results have such high quality that the composites of the foreground regions on different backgrounds look very natural.

For the quantitative evaluation, we obtained ground truth segmentation data by hand-labeling, then found the *error rate* for each frame which is defined as

$$\text{error rate} = \frac{\text{number of misclassified pixels}}{\text{number of all pixels}} \tag{12}$$

The error rates for all frames are shown in Fig. 10. In Fig. 10(a), we show the comparison between our method and [5], using the sequence "Foreman". We obtained a better result with an average error rate = 0.6%. Figure 10(b) shows the comparisons between our method and two conventional methods [4,6] using the sequence "Video Chatting". This sequence is difficult to process, because the background changes noticably with camera movement. In [4], a thermal vision camera is employed to provide additional information; in [6], motion features are learned in advance from the groundtruth data of the first 100 frames. Meanwhile, in our method, the segmentation process is performed online using a single camera, without requiring any future information. However, in spite of more critical conditions, our segmentation method attains the highest quality. This result proved the effectiveness of the proposed likelihood propagation technique.

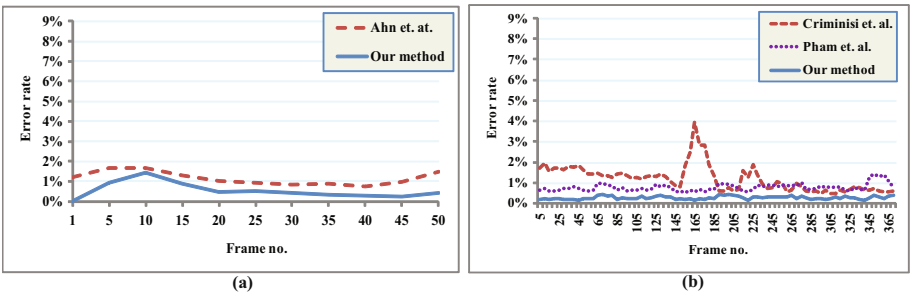


Fig. 10. Comparisons of segmentation accuracy between our method and: (a)Ahn et al.[5] using "Foreman", (b)Pham et al.[4], Criminisi et al.[6] using "Video Chatting".

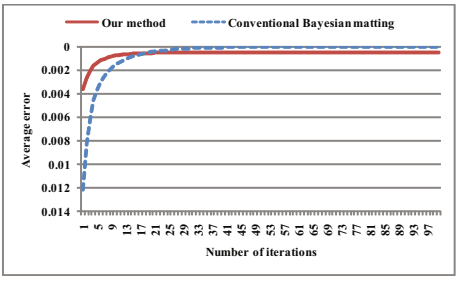


Fig. 11. Matting accuracy evaluation

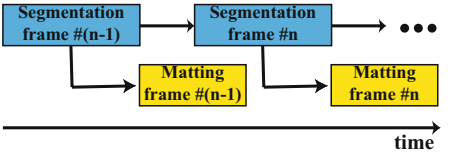


Fig. 12. Parallel processing

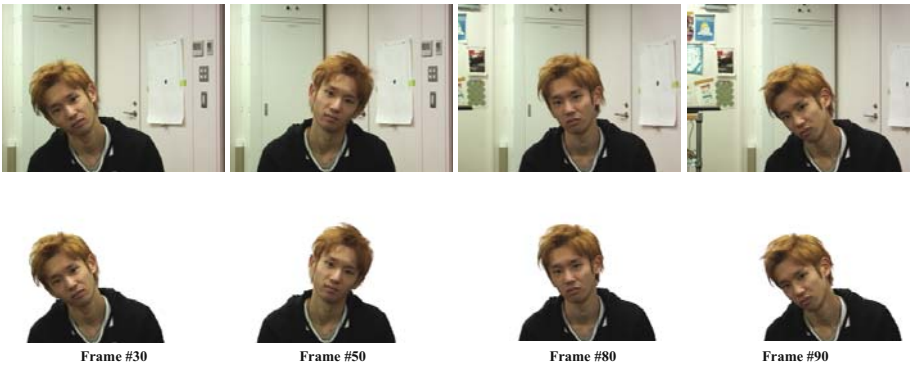


Fig. 13. Matting results for the sequence "Headshake"

4.2 Evaluation of Matting

In Section 3.2, we introduced two improvements to the original Bayesian matting [3], which can increase the calculation speed by 5 times while keeping the quality. In this section, we perform a detailed quantitative evaluation of the matting accuracy. We employed the *average error* as the evaluating function

$$\text{average error} = \frac{\sum_{\text{pixel } i} |\text{resulting } \alpha_i - \text{true } \alpha_i|}{\text{number of pixels}} \tag{13}$$

True α_i was obtained from the convergence result of the iteration process (see Eq. (11)), using the original Bayesian matting [3]. Figure 11 shows the relationship between the number of iterations and the average error. Here, we employed one of our captured sequences, named "Headshake", as the sample sequence. Similar results were attained for other sequences. There are some conclusions that can be drawn from this result:

- Our method converges faster than the original one [3].
- Because our method performs approximations using down-sampling, the result can not reach the true value.

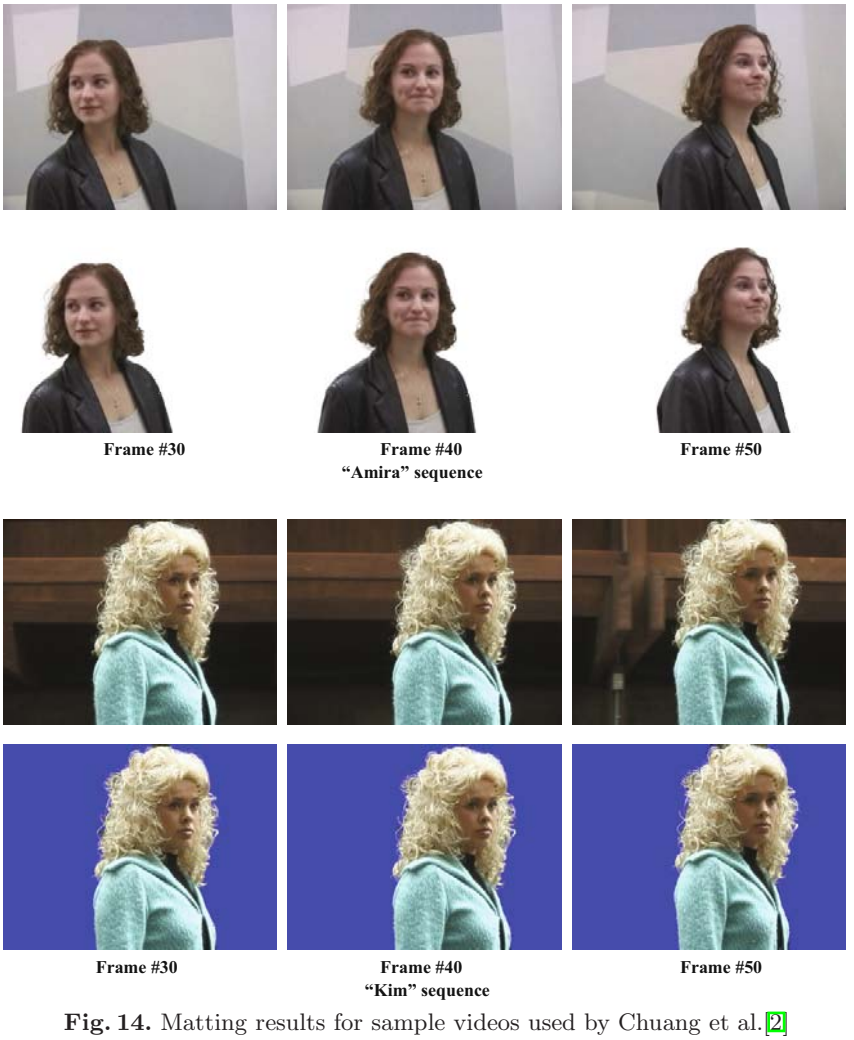


Fig. 14. Matting results for sample videos used by Chuang et al.[2]

– However, for a small number of iterations, our method produces smaller average error than the original one [3]. Therefore, our method is more effective for real-time processing purposes.

Figure 13 shows matting results for the sequence "Headshake". The difficulty of this sequence is that the distribution of background color changes due to camera movement. We also applied our method to the sample videos from Chuang et al.'s paper [2], which is available from their website[1]. The results for some frames are shown in Fig. 14. In [2], the matting process is performed off-line, and requires user-supplied trimaps for every 9-10 frames. In contrast, our method requires hand-made foreground and background for *only the first frame*, and performs matting automatically for the remaining frames. Even though our method can save lots of user interactions and runs much faster, its matting quality is comparable to Chuang's method [2]. Matting results for the whole sequence can be seen from the supplemental file.

4.3 Calculation Time

Our experiments were performed on a PC with Intel Core2 Quad CPU, 2.40GHz and 4GB memory. For a video sequence with resolution 320×240, the processing time for each frame is 108ms for the segmentation step, and 72ms for the matting step. The serial processing, which performs the two steps in sequence, costs about 180ms. However, by employing parallel processing (see Fig. 12), in which the segmentation step for the n^{th} frame and the matting step for the $n-1^{\text{th}}$ frame are performed in parallel, we can decrease the processing time to 120ms ($\sim$*8fps*). This processing time is enough for real-time applications.

5 Conclusions

In this paper, we present an online video matting method that runs in real-time based on bilayer segmentation. We first introduced an accurate bilayer segmentation method for extracting the foreground region from the background using color likelihood propagation. For the second step, we performed alpha-matting based on the segmentation result. To enable real-time processing, we modified the conventional Bayesian matting method by using down-sampling and smart initialization, which increase the calculation speed by 5 times while maintaining the quality. Various experimental results showed the effectiveness of our method.

References

1. Kelly, D.: Digital composition. The Coriolis Group (2000)
2. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D.H., Szeliski, R.: Video matting of complex scenes. ACM Transactions on Graphics 21(3), 243–248 (2002)

[1] http://grail.cs.washington.edu/projects/digital-matting/video-matting/

3. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proc. CVPR, pp. 264–271 (2001)
4. Pham, V.Q., Takahashi, K., Naemura, T.: Live video segmentation in dynamic backgrounds using thermal vision. In: Proc. Pacific-Rim Symposium on Image and Video Technology, January 2009, pp. 143–154 (2009)
5. Ahn, J.K., Kim, C.S.: Real-time segmentation of objects from video sequences with non-stationary backgrounds using spatio-temporal coherence. In: Proc. ICIP, pp. 1544–1547 (2008)
6. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proc. CVPR, vol. 1, pp. 53–60 (2006)
7. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proc. ICCV, vol. I, pp. 105–112 (2001)
8. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. Trans. PAMI, 1124–1137 (2004)
9. Shi, J., Tomasi, C.: Good features to track. In: Proc. CVPR, pp. 593–600 (1994)
10. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker. In: OpenCV documentation (2001)

Transductive Segmentation of Textured Meshes

Anne-Laure Chauve, Jean-Philippe Pons, Jean-Yves Audibert,
and Renaud Keriven

IMAGINE, ENPC/CSTB/LIGM, Université Paris-Est, France

Abstract. This paper addresses the problem of segmenting a textured mesh into objects or object classes, consistently with user-supplied seeds. We view this task as transductive learning and use the flexibility of kernel-based weights to incorporate a various number of diverse features. Our method combines a Laplacian graph regularizer that enforces spatial coherence in label propagation and an SVM classifier that ensures dissemination of the seeds characteristics. Our interactive framework allows to easily specify classes seeds with sketches drawn on the mesh and potentially refine the segmentation. We obtain qualitatively good segmentations on several architectural scenes and show the applicability of our method to outliers removing.

1 Introduction

The generalization of digital cameras, the increase in computational power brought by graphical processors and the recent progress in multi-view reconstruction algorithms allow to create numerous and costless textured 3D models from digital photographs. In this work, we address the problem of segmenting a textured mesh into objects or object classes. The segmentation is an essential step of scene analysis, and can be used for semantic enrichment of architectural scenes, reverse engineering, subsequent recognition of known rigid objects... A meaningful mesh decomposition enables to retrieve objects of interest or remove undesired parts (see Fig.4). This problem raises several interesting challenges: the variability of scenes and object types to handle; the subjectivity of a meaningful segmentation, which depends on the application; the simultaneous classification and segmentation involved by object detection. We took up theses challenges by designing a flexible and interactive framework that conducts collective classification of the mesh.

We propose an easy-to-use, graphical tool to provide labeled training seeds by drawing sketches on the mesh, select appropriate features among the set of available ones, compute the segmentation via our transductive learning algorithm based on Support Vector Machine classification and Laplacian graph regularization, visualize the resulting segmented mesh and refine the training sketches to rerun the segmentation if necessary.

Our work is related to three main research topics: mesh segmentation, collective point cloud classification and transductive image segmentation. Even if we consider meshes in this paper, the type of data and the segmentation technique turn out to be quite different from the ones considered in the literature.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 502–513, 2010.

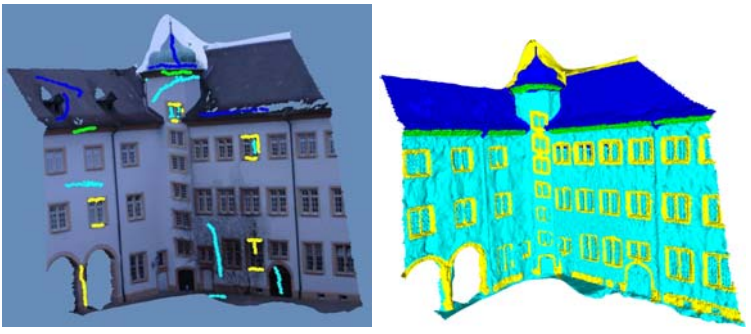


Fig. 1. Segmentation of the mesh into four classes: roof, wall, windows edges, cornice. Left: the input textured mesh with user supplied sketches. Right: the resulting segmentation using our algorithm.

Our issues are actually closer to a series of works on point cloud segmentation, however these are not directly applicable to our meshes whose connectivity and texture are of crucial importance. Our algorithm is rather based on a transductive segmentation method that was developed at first for 2D images.

The problem of mesh segmentation has become an important issue in various computer graphics applications, like parameterization and texture mapping, metamorphosis, 3D shape retrieval or modeling by example. Several segmentation algorithms for mesh partitioning have been compared in [1]. These algorithms deal with non-textured meshes of a single object, that are usually high-quality meshes coming from CAD models or dense scans – thus very different from the image-based meshes of entire scenes we process. These algorithms fall into two main categories. The first one gathers geometric approaches, where the mesh is segmented into patches fitted with simple mathematical surfaces. The typical application of these methods is reverse engineering of CAD models. The second one is rather semantic-oriented: the aim is to decompose meshes of "natural" objects (e.g., a body model) into "meaningful" pieces (e.g., the head, two arms, two legs and the torso). However, these semantic approaches do not involve any learning-based or classification procedure, and do not consider the similarities between different, non connected parts. Mesh decomposition is usually a preprocessing step, and should thus be able to handle various types of input models and of target applications. In order to deal with varying application-dependent requirements and with the subjectivity of a meaningful segmentation, [2] proposed an interactive framework similar to ours: the user draws sketches on the mesh that provide seeds for the algorithm. Nevertheless, as mentioned above and unlike [2], we process textured meshes: this greatly reduces the required amount of user interaction (see for example Fig.1).

A series of works [3,4,5,6] has been carried out on the problems of segmenting scan data into objects or objects classes using Markov networks. This problem differs from the one we address mainly on the type of data to process: textured meshes possess supplementary attributes like color or mesh connectivity that

we exploit, while 3D point clouds are far denser, and more precise, thus targeting distinct applications and requiring different processes. However these works share several characteristics with ours. One is the use of collective classification through graph-based methods: instead of classifying each point or facet independently, the problem is thought of as a global classification task and adjacency relationships are used to enforce spatial contiguity of the labels on the graph. Our graph Laplacian transduction performs such a collective classification. Besides, in both problems, the ability to handle various type of scenes as well as a certain variability inside a given class of objects is crucial: this is achieved by taking advantage of various kinds of features. Thus, the Markov random field segmentation algorithm of [3] has been tested successfully on both outdoor and indoor scenes, for real-world and synthetic scan datasets. In our framework, textured meshes possess both geometric and photometric attributes that should be chosen according to the scene type and jointly exploited. We combine these features in kernel weights of a graph Laplacian regularizer; the inherent modularity of kernel methods hence provides the desired flexibility.

Recently, several works have addressed the image segmentation problem in an interactive framework: a set of seed pixels representative of each region to be segmented is specified by the user, and the segmentation of the entire image is performed consistently with the seeds. The existing algorithms rely on computing weighted geodesic distances [7], graph cuts with discontinuity penalization [8], graph cuts with Gaussian mixture modelling of the segmented regions [9], random walks on a graph and its relation to electrical resistance in a circuit [10,11] and transductive learning [12]. Due to its simplicity and its effective results obtained in the Microsoft GrabCut benchmark, the latter approach is adopted in our mesh segmentation algorithm. The segmentation results are visualized in our interactive framework, allowing for subsequent refinement of the query.

Outline. The rest of the paper is organized as follows. Section 2 presents the Laplacian regularizer on a graph to perform transduction and introduces the energy we minimize in the sequel; section 3 details our algorithm for the segmentation of textured meshes, and section 4 presents our experimental results. We conclude in section 5 with a brief discussion.

2 Graph Laplacian Transduction

2.1 Transductive Inference

The problem we are concerned with is a supervised classification task: given a set of labeled examples (called *training set*), we want to infer the labels of new input points (called *test set*). More specifically, we consider an input space $\mathcal{X}$ and an output space $\mathcal{Y}$ (typically $\mathcal{Y} = \{0, \ldots, K\}$ for a classification problem); given a set of input-output couples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we want to determine the label y of a new point x.

There are two different approaches for this problem, the *inductive* and *transductive* settings. Inductive inference is a two-step process: one tries first to learn a function $f : \mathcal{X} \to \mathcal{Y}$ to map the entire input space to the output space, and then, for every point x of the test set, predicts $y = f(x)$ as its label. In contrast, in transductive inference, the test inputs are known beforehand. Thus, no general input-output mapping is inferred, but only the labels of the test points are predicted. This approach follows Vapnik's principle: "when solving a problem of interest, do not solve a more general problem as an intermediate step", and takes advantage of the unlabeled data to get an idea of the input space distribution. Indeed, transduction relies on this second principle, referred to as the *smoothness assumption*: "outputs vary a lot only on input regions having low density", or, in other words, "the decision boundary should lie in a low-density region".

Transductive inference can be compared to semi-supervised learning (SSL), where the training set is made up of both labeled and unlabeled data points. Like transduction, SSL utilizes the unlabeled points to infer the input distribution. However, in contrast to transduction, the test points are not (necessarily) known beforehand and semi-supervised algorithms can handle unseen data, thus being part of an inductive framework.

A large number of algorithms that have been proposed in the last few years for transductive inference relies on graph-based methods (see [13], [14] and references within), where nodes represent data points and edges encode similarities between them. The graph structure is used to propagate information from the known labels to the unlabeled points. We use the Laplacian graph regularizer of [12] with an unnormalized kernel (case $s = 2, \lambda = 0$): the labeling of the facets is carried out as the minimization of a quadratic cost function derived from the graph. The work [15] studies several cost functions for regularization on a graph and explains the links with label propagation algorithms.

2.2 Graph Laplacian Regularization and Energy Minimization

In the first place, we consider a binary classification problem, where the two classes have respectively 0 and 1 as labels (see 2.3 for generalization to the multi-class problem); 0 is the background class and 1 is the object class. Instead of directly predicting the labels of the test points ($y \in \{0, 1\}$), we consider a real-valued output space ($y \in \mathbb{R}$) and assign each point to the class $\mathbf{1}_{y \geq 1/2}$.

Graph Laplacian. The geometry of the data is represented by a graph $G = (V, E)$ where the nodes $V = \{X_1, \ldots, X_n\}$ correspond to the input points coming from both the p labeled instances $\{X_1, \ldots, X_p\}$ of the training set, and the $n - p$ unlabeled data $\{X_{p+1}, \ldots, X_n\}$ of the test set, and the edges E represent similarities between them, in the form of a *weight* matrix W (of size $n \times n$). The coefficients of this matrix (also called *affinity* or *adjacency* matrix) must satisfy:

- $W_{ij} = 0$ if X_i and X_j are not "neighbours" (*i.e.* are not connected by an edge),

– $W_{ij} \geq 0$,
– $W_{ij} = W_{ji}$.

The meaning of the neighbourhood needs to be specified and depends on the nature of the data (e.g., facet adjacency on a mesh, symmetrized k-nearest neighbour for a point cloud, etc.). We write the weights in the form $W_{ij} = k(X_i, X_j)$ where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric positive function giving the similarity between two input points. A typical example is the Gaussian kernel $k(x, x') = \exp\left(-\frac{d(x,x')^2}{2\sigma^2}\right)$ with d a distance on the input space $\mathcal{X}$ and σ a positive parameter.

Let D be the diagonal matrix $D_{ii} = \sum_j W_{ij}$. The matrix $L = D - W$ is called the *unnormalized graph Laplacian*. It is a discrete analogue of the Laplace-Beltrami operator on a Riemannian manifold (see (3)).

Energy design. The goal is to find a labeling $Y = (y_1, \ldots, y_n)$ of the graph that is consistent with both the labeled training points and the geometry of the entire data (represented by the graph structure).

We partition the vector Y and the matrices W, D and L according to their labeled and unlabeled parts :

$$Y = \begin{pmatrix} Y_\ell \\ Y_u \end{pmatrix} \quad W = \begin{pmatrix} W_{\ell\ell} & W_{\ell u} \\ W_{u\ell} & W_{uu} \end{pmatrix} \quad D = \begin{pmatrix} D_{\ell\ell} & 0 \\ 0 & D_{uu} \end{pmatrix} \quad L = \begin{pmatrix} L_{\ell\ell} & L_{\ell u} \\ L_{u\ell} & L_{uu} \end{pmatrix}.$$

The initial labels of the training points are denoted $Y_\ell^0 = (y_1^0, \ldots, y_p^0)$.

We want to find the optimal labeling $\hat{Y}$ minimizing the following cost criterion, built up three different terms :

$$\hat{Y} = \underset{Y \in \mathbb{R}^n}{\arg\min} \underbrace{c \sum_{i=1}^{p} (y_i - y_i^0)^2}_{(\alpha)} + \underbrace{\frac{1}{2} \sum_{i,j=1}^{n} W_{ij}(y_i - y_j)^2}_{(\beta)} + \underbrace{\lambda \sum_{i=p+1}^{n} (y_i - s_i)^2}_{(\gamma)}, \quad (1)$$

where the scores s_i in (γ) are defined hereafter.

(α) Consistency with the initial labeling: $\sum_{i=1}^{p} (y_i - y_i^0)^2 = \|Y_\ell - Y_l^0\|^2$.

The parameter $c \in [0, +\infty]$ expresses the confidence assigned to the training outputs (it could be different for each point, $(c_i)_{i=1,\ldots,p}$).

(β) Consistency with the geometry of the data: This term penalizes rapid changes in $\hat{Y}$ between points that are close on the graph (as given by the similarity matrix W). It enforces the smoothness assumption along the graph.

$$\begin{aligned} \frac{1}{2} \sum_{i,j=1}^{n} W_{ij}(y_i - y_j)^2 &= \frac{1}{2}\Big(2 \sum_{i=1}^{n} y_i^2 \sum_{j=1}^{n} W_{ij} - 2 \sum_{i,j=1}^{n} W_{ij} y_i y_j\Big) \\ &= Y^\top (D - W) Y = Y^\top L Y \,. \end{aligned}$$

The latter term can be interpreted as a graph Laplacian regularizer in the following sense. Assume that the inputs are real vectors admitting a distribution with density p (with respect to the Lebesgue measure). Let φ be a smooth function such that $y_i = \varphi(X_i)$. From [12], the quantity $Y^T LY$ is a discrete approximation, up to a constant multiplicative factor, of the integral

$$\int \|(\nabla\varphi)(x)\|^2 p^2(x)dx\,, \tag{2}$$

which is equal to

$$\int \varphi(x)(\Delta\varphi)(x)p^2(x)dx\,, \tag{3}$$

for Δ a (weighted) Laplace-Beltrami operator. From (2), we see that the regularization term $Y^T LY$ will be small only when the labels vary in low density regions of the input space, thus fulfilling the principle stated in Section 2.1.

(γ) Consistency with some additional knowledge: $\sum_{i=p+1}^{n} (y_i - s_i)^2 = \|Y_u - S_u^0\|^2\,.$

This terms allows to incorporate either some prior information or the output of another algorithm in the form of a score s_i, measuring for each test point i the "likelihood" that it belongs to the object class. This term can be seen as an initializing score. We note S_u^0 the $(n-p)$-vector $(s_{p+1}, \ldots, s_n)$.

The optimization problem (1) can hence be expressed in the following matrix form:

$$\hat{Y} = \underset{Y\in\mathbb{R}^n}{\arg\min}\ c\|Y_\ell - Y_\ell^0\|^2 + Y^\top LY + \lambda\|Y_u - S_u^0\|^2\,. \tag{4}$$

Sparse linear system giving the predicted labels. Since we assume that the user-supplied seeds are trustworthy, we constrain the labels on the labeled data ($\hat{Y}_\ell = Y_\ell^0$) and thus consider an infinite regularization coefficient $c = +\infty$. Hence the minimization is carried over the labeling Y_u of the test points, and the optimization problem rewrites

$$\begin{aligned}
\hat{Y}_u &= \underset{Y_u\in\mathbb{R}^{n-p}}{\arg\min}\ Y^\top LY + \lambda\|Y_u - S_u^0\|^2 \\
&= \underset{Y_u\in\mathbb{R}^{n-p}}{\arg\min}\ {Y_\ell^0}^\top L_{\ell\ell}Y_\ell^0 + {Y_u}^\top L_{u\ell}Y_\ell^0 + {Y_\ell^0}^\top L_{\ell u}Y_u + {Y_u}^\top L_{uu}Y_u \\
&\qquad\qquad + \lambda\left({Y_u}^\top Y_u - {Y_u}^\top S_u^0 - {S_u^0}^\top Y_u + {S_u^0}^\top S_u^0\right) \\
&= \underset{Y_u\in\mathbb{R}^{n-p}}{\arg\min}\ 2Y_u^\top\left(L_{u\ell}Y_\ell^0 - \lambda S_u^0\right) + Y_u^\top\left(L_{uu} + \lambda I\right)Y_u\,.
\end{aligned}$$

In order to minimize the cost criterion, we compute its derivative with respect to Y_u. Let $A = L_{u\ell}Y_\ell^0 - \lambda S_u^0$ and $B = L_{uu} + \lambda I$. The matrix B is symmetric positive definite matrix when $\lambda > 0$, since L_{uu} is symmetric positive semi-definite:

$${Y_u}^\top L_{uu}Y_u = \frac{1}{2}\sum_{i,j=p+1}^{n} W_{ij}(y_i - y_j)^2 \geq 0\,.$$

Thus the function $f : Y_u \mapsto 2Y_u^\top A + Y_u^\top B Y_u$ is strictly convex and admits a unique minimum where its derivative equals 0:

$$\frac{\partial f(Y_u)}{\partial Y_u} = 2A + 2BY_u = 0 \quad \Longleftrightarrow \quad BY_u = -A\,.$$

Hence, the optimal labeling $\hat{Y}_u$ is the solution of the sparse linear system

$$(L_{uu} + \lambda I)\,\hat{Y}_u = \lambda S_u^0 - L_{u\ell} Y_\ell^0\,. \tag{5}$$

2.3 Multi-class Segmentation

The extension of the previous algorithm to the multi-class case is straightforward using a one-versus-all approach. If there are d different classes, we resolve the linear system (5) for each class k against all other classes as background, thus having as initial label for a training point i, $Y_\ell^{0,k}(i) = 1$ if i is labeled k and $Y_\ell^{0,k}(i) = 0$ otherwise. We obtain d output vectors $\hat{Y}_u^k$, and assign the test point j to the class $\underset{k=1,\ldots,d}{\arg\max}\, \hat{Y}_u^k(j)$.

3 Segmentation of Textured Meshes

3.1 Our Algorithm

We work on textured meshes of a 3D scene built from a set of calibrated images: we use the multi-view stereovision algorithm from [16,17,18] to reconstruct a 3D model of the scene and the multi-band blending algorithm from [19] to compute a texture atlas with minimal color discontinuities or blurring. The meshes presented in our experiments have been reconstructed from datasets provided by C. Strecha *et al.* [20] (castle-P19, castle-P30 and Herz-Jesu-25).

We want to classify the facets of the mesh given seed facets for each class. The seeds are provided by the user who draws sketches on the mesh. Thus we consider a graph G with a node for every facet of the mesh and an edge between any two adjacent facets. The modularity of our kernel method allows to chose various edges weights, depending on the scene type. We use a Gaussian kernel with a distance between facets constructed from one or several features characterizing the facets (see section 3.2 for more details and examples of such features).

The training points are the user-supplied seeds and we use an SVM classification on these training points as additional knowledge on the test points (see section 3.3). The linear system (5) of the transductive classification is sparse due to the sparseness of the facet adjacency relationship on the mesh. We solve it with a conjugate gradient algorithm (we use the IML++ implementation of [21]). For each facet, we obtain a score for each class: the classification is given by the argmax of these scores (see section 2.3) but they carry much more information that can be exploited.

The interactive framework of our method allows to add supplementary seeds depending on misclassified facets and rerun the algorithm in order to improve the segmentation result or add new classes (see Fig 4).

3.2 Features and Kernels

We compare two different facets through a set of attributes extracted from the mesh. We describe each facet by a feature vector of length m. These features are chosen according to the scene type and the discriminant characteristics of such scenes. A major element of this algorithm is the ability to take advantage of several different kinds of features, mixing photometric and geometric informations or any other available attribute (see Fig.2).

Here are some examples of features that we used in the results of Section 4 or that could be used in other experiments. On the one hand, we use photometric features like the mean color of the facet or components of the mean color (for instance, without luminance to account for changes in illumination between cameras). On the other hand, we use geometric features like the normal of the facet (or solely its vertical component), the position of the center of the facet (for instance, height of the center), or the discrete curvature of the mesh. We could also also evaluate less local features that would consequently be smoother and more robust by averaging the previous features on a neighbourhood of the facet.

We do not compare globally the descriptor vectors of the facets, but feature by feature. Then we combine the component-wise distances in the kernel weight by multiplying the Gaussian kernels associated with each feature: if there are n_f different features (each feature being a vector of variable length) and we note $(f_i^1, \ldots, f_i^{n_f})$ and $(f_j^1, \ldots, f_j^{n_f})$ the feature vectors of facets X_i and X_j,

$$W_{ij} = k(X_i, X_j) = \prod_{k=1}^{n_f} \exp\left(-\frac{\|f_i^k - f_j^k\|^2}{2\sigma_k^2}\right) = \exp\left(-\sum_{k=1}^{n_f} \frac{\|f_i^k - f_j^k\|^2}{2\sigma_k^2}\right).$$

The kernel weight is thus parameterized by the n_f values $(\sigma_1, \ldots, \sigma_{n_f})$ which determine the trade-off between the various features.

3.3 Additional Knowledge on the Test Points

We train Support Vector Machines with the user-supplied training points in order to provide initialization information on the test points. We obtain a score s_i^k for each facet i and each class k with a one-versus-all SVM algorithm. We use the LibSVM implementation of the C-SV Classification (cf. [22]).

The incorporation of external knowledge in addition to the graph Laplacian regularization is essential in order to detect several objects of the same class which form several connected components. Indeed, each seed can generate at most one connected component in the segmentation. Besides, the initialization with the SVM scores accelerate the label propagation on the mesh and the convergence of the iterative resolution compared to the transductive segmentation based solely on graph Laplacian.

We can consider either the same kernel as for the graph Laplacian weights or a different one (different features, kernel type, or parameters), if we want to exploit distinct properties of the mesh.We learn the parameters of the SVM

(parameters of the kernel plus parameter C of the SVM) in a cross-validation process. Hence, the tradeoff between the various features combined in the kernel is learned automatically. Moreover, we can employ these selected parameters in the Laplacian kernel for transduction (if we use the same kernel).

The Support Vector Machine approach is a classification method by itself and provides a comparison basis for our algorithm (just as the transductive segmentation without the SVM initialization). However it does not enforce any spatial contiguity on the mesh, and it produces a noisy result (see Fig.3).

Note that we learn this term directly on the scene of interest, but we could incorporate as well a prior learned from other, previously segmented scenes, instead or in addition to this one. In this case, however, the algorithm is not anymore a purely transductive one.

3.4 Computational Complexity and Time

The computational cost of the algorithm can be split up into the cost of the SVM initialization and the cost of the transductive segmentation on the graph. Since in our framework the number of training points p is negligible compared to the total number of points n, we consider the complexity of the algorithm with respect to n only. In the SVM algorithm, the cost of the test prevails over the cost of the train (which is between $\mathcal{O}(p^2)$ and $\mathcal{O}(p^3)$), and its complexity is $\mathcal{O}(n.p) = \mathcal{O}(n)$. In the energy minimization of the transduction, the main cost comes from solving the sparse linear system (5); this can be done in $\mathcal{O}(k.m)$ where k is the number of iterations of the conjugate gradient and m is the number of non-zero entries in the matrix, which is equal to $4n$. We bound the number of iterations to limit the computational time, thus having a total computational complexity in $\mathcal{O}(n)$ (at the expense of precision on the convergence).

In practice, the computational time of the transduction prevails over the one of SVM. The whole segmentation process takes between fifteen seconds and five minutes on a Xeon 2.33 GHz, for meshes with 50,000 to 1,500,000 facets.

4 Experimental Results

Figure 1 in the introduction shows a first example of segmentation into four different classes using the mean color of a facet and the altitude of its center as features. The results are qualitatively good, and mostly agree with perceptual boundaries. Note that every window is detected while only few where initially labeled, illustrating the ability of our algorithm to detect several objects of the same class forming several connected components, thanks to the initialization of the transduction with the classification results of the SVM.

Figure 2 illustrates the importance of combining several different features in order to obtain a relevant segmentation. Indeed the use of mean color or altitude alone produces erroneous results (even if the latter seems less wrong, it is even more naive, and performs a simple thresholding on the altitude), while their combination gives a pertinent result. Hence, the set of features is chosen

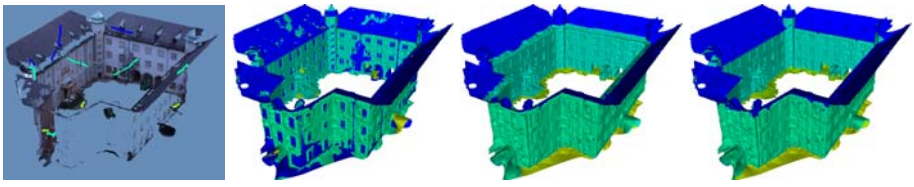


Fig. 2. Combining different features improves the segmentation. 3 classes: roof, wall, ground. From left to right: input textured mesh with seeds; segmentation using mean color; segmentation using altitude; segmentation using both mean color and altitude.

according to the scene of interest: in Fig.3, the segmentation of the top row is performed using color only, while in the bottom row, the color, the curvedness and the vertical normal of the facet are combined.

Figure 3 compares the classification of the SVM and the final result after transduction: as mentioned in section 3.3, the results of the SVM alone are noisy and do not exploit the connectivity of the mesh.

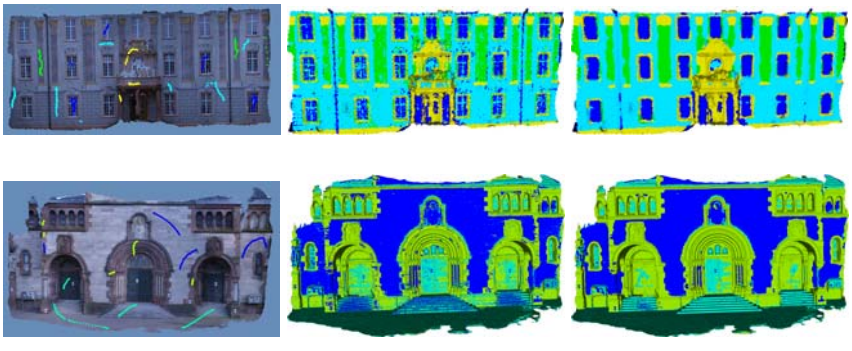


Fig. 3. Comparison of classification results using using SVM alone or SVM plus transductive segmentation. Left: textured mesh with selected facets. Middle: classification obtained with SVM; the labels are noisy and do not provide a decomposition of the mesh. Right: classification obtained with SVM initialization plus transduction.

Our algorithm can be used to remove outliers in a scene. In the original mesh of the castle of Figure 4, obtained by multi-view reconstruction, a portion of the sky has been reconstructed running on from the roof. It can be easily remove in our interactive framework, using only two classes, one for the outliers (sky facets), one for the inliers (castle facets), and color and altitude as features. The first segmentation *(middle column)* is not entirely satisfying (some portions of sky remain), so we add a few labeled points and rerun the algorithm, then obtaining a good segmentation *(right column)*.

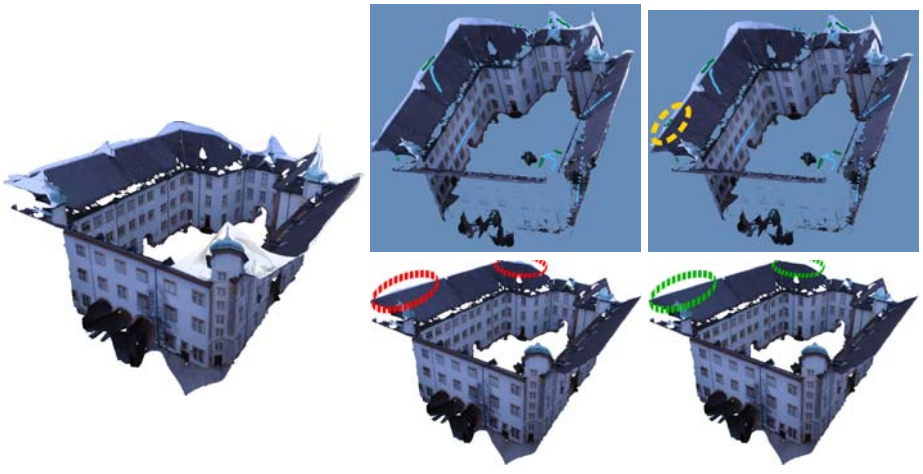


Fig. 4. Removing outliers. Left: original textured mesh of the castle with portions of "sky" mistakenly reconstructed by stereo. Middle: removing sky facets using selected facets. Right: refinement of the previous segmentation by adding sky facets to the selection (*yellow outline*). Top row: selected facets (*green: sky outliers, blue: castle inliers*), bottom row: textured meshes.

5 Conclusion

We have presented an efficient procedure for the segmentation of textured meshes: we designed a sketch-based interactive framework which produces a meaningful segmentation according to the user's aims, thanks to a possible refinement of the training sketches. Our experiments demonstrate that we can take advantage of geometric and photometric features at the same time, combining a various number of appropriate features in our kernel. The graph Laplacian regularizer enforces the spatial contiguity of the labels on the mesh, producing robust decompositions of the mesh for subsequent applications while the SVM initialization allows to detect non-connected, similar objects. Future work will focus on the development of supplementary features and kernels, in order to apply this algorithm to a larger range of scenes as well as to other types of data like point clouds.

References

1. Attene, M., Katz, S., Mortara, M., Patanè, G., Spagnuolo, M., Tal, A.: Mesh Segmentation - A Comparative Study. In: SMI (2006)
2. Wu, P., Yang, M.: A sketch-based interactive framework for real-time mesh segmentation. In: CGI (2007)
3. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.Y.: Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In: CVPR (2005)
4. Triebel, R., Kersting, K., Burgard, W.: Robust 3D Scan Point Classification using Associative Markov Networks. In: ICRA (2006)

5. Triebel, R., Schmidt, R., Martínez Mozos, O., Burgard, W.: Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Raneg Data. In: IJCAI (2007)
6. Munoz, D., Vandapel, N., Hebert, M.: Directional Associative Markov Network for 3-D Point Cloud Classification. In: 3DPVT (2008)
7. Bai, X., Sapiro, G.: A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In: ICCV (2007)
8. Boykov, Y.Y., Jolly, M.P.: Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In: ICCV (2001)
9. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.H.S.: Interactive Image Segmentation Using an Adaptive GMMRF Model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
10. Grady, L.: Random Walks for Image Segmentation. PAMI 28, 1768–1783 (2006)
11. Kim, T.H., Lee, K.M., Lee, S.U.: Generative Image Segmentation Using Random Walks with Restart. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 264–275. Springer, Heidelberg (2008)
12. Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J., Ségonne, F.: Segmentation by transduction. In: CVPR (2008)
13. Belkin, M., Niyogi, P.: Semi-Supervised Learning on Riemannian Manifolds. Machine Learning 56, 209–239 (2004)
14. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge (2006)
15. Bengio, Y., Delalleau, O., Roux, N.L.: Label Propagation and Quadratic Criterion. In: Chapelle, Schölkopf, Zien (eds.) Semi-supervised Learning, pp. 193–216. MIT Press, Cambridge (2006)
16. Labatut, P., Pons, J.P., Keriven, R.: Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In: ICCV (2007)
17. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view Stereo Reconstruction and Scene Flow Estimation with a Global Image-Based Matching Score. IJCV 72, 179–193 (2007)
18. Vu, H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: CVPR (2009)
19. Allène, C., Pons, J.P., Keriven, R.: Seamless Image-Based Texture Atlases using Multi-band Blending. In: ICPR (2008)
20. Strecha, C., von Hansen, W., Van Gool, L.J., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-view Stereo for High Resolution Imagery. In: CVPR (2008)
21. Dongarra, J., Lumsdaine, A., Pozo, R., Remington, K.: A Sparse Matrix Library in C++ for High Performance Architectures. In: Proc. of the Second Object Oriented Numerics Conference (1992)
22. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Levels of Details for Gaussian Mixture Models

Vincent Garcia[1], Frank Nielsen[1,2], and Richard Nock[3]

[1] Ecole Polytechnique
Laboratoire d'informatique LIX
91128 Palaiseau Cedex, France
[2] Sony Computer Science Laboratories, Inc.
3-14-13 Higashi Gotanda
141-0022 Shinagawa-Ku, Tokyo, Japan
[3] Université des Antilles-Guyane, CEREGMIA
Campus de Schoelcher, BP 7209
97275 Schoelcher, Martinique, France

Abstract. Mixtures of Gaussians are a crucial statistical modeling tool at the heart of many challenging applications in computer vision and machine learning. In this paper, we first describe a novel and efficient algorithm for simplifying Gaussian mixture models using a generalization of the celebrated k-means quantization algorithm tailored to relative entropy. Our method is shown to compare experimentally favourably well with the state-of-the-art both in terms of time and quality performances. Second, we propose a practical enhanced approach providing a hierarchical representation of the simplified GMM while automatically computing the optimal number of Gaussians in the simplified mixture. Application to clustering-based image segmentation is reported.

1 Introduction and Prior Work

A mixture model is a powerful framework to estimate the probability density function of a random variable. For instance, the Gaussian mixture models (GMMs for short) – also known as mixture of Gaussians (MoGs) – have been widely used in many different area domains such as image processing. For a given mixture model f, the probability density function evaluated at $x \in \mathbb{R}^d$ is given by

$$f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) \tag{1}$$

where $0 \leq \alpha_i \leq 1$ denotes the weight of each mixture component f_i such as $\sum_{i=1}^{n} \alpha_i = 1$. Given a GMM f, each function f_i is a multivariate Gaussian function

$$f_i(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}{2}\right) \tag{2}$$

parametrized by its mean $\mu_i \in \mathbb{R}^d$ and its covariance symmetric positive-definite matrix $\Sigma_i \succ 0$. It is common to estimate model parameters from independent and

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 514–525, 2010.

identically-distributed observations using the expectation-maximization (EM) algorithm [1].

A typical operation on mixture models is the estimation of statistical measures such as Shannon entropy or the Kullback-Leibler divergence. With large number of components in the mixture model (*e.g.* arising from a kernel-based Parzen density estimation [2]), the estimation of these measures is prohibitive in terms of computation time. The computational time can be strongly decreased by reducing the number of components in the mixture model. The simplest method to obtain a compact representation of f is to re-learn the mixture model directly from the source dataset. However, this may not be applicable for two reasons. First, the estimation of a mixture model is computationally expensive if we consider large datasets. Second, the source dataset can be unavailable. Thus, the most appropriated solution is to simplify the initial mixture model f.

Given a mixture model f composed of n components (see equation (1)), the problem of mixture model simplification consists in computing a simpler mixture model g

$$g(x) = \sum_{j=1}^{m} \alpha'_j g_j(x) \tag{3}$$

with m components ($1 \leq m < n$) such as g is the "best" approximation of f with respect to a similarity measure.

Some GMM simplification methods have been proposed in the last decade. Zhang and Kwok [3] have proposed to simplify a GMM by first grouping similar components together and then performing local fitting through function approximation. By using the squared loss to measure the distance between mixture models, their algorithm naturally combines the two different tasks of component clustering and model simplification. Goldberger *et al.* [4] have proposed a fast GMM simplification algorithm named UTAC (Unscented Transform Approximation Clustering) based on the Unscented Transform (UT) method [5] [6]. The UTAC algorithm proceeds by maximizing the UTA (Unscented Transform Approximation of the negative cross-entropy) criterion computed between the two GMMs f and g. The authors have shown that the UTA criterion can be maximized with a standard EM-like algorithm. Davis and Dhillon [7] have proposed a hard clustering algorithm based on the decomposition of the relative entropy as the sum of a Burg matrix divergence with a Mahalanobis distance parametrized by the covariance matrices. Goldberger and Roweis [8] have proposed a GMM simplification algorithm based on the k-means hard clustering.

These methods have two disadvantages. First, they only consider the problem of GMM simplification. However, other kind of mixture models have been successfully used in different applications such as multinomial mixture models in text classification [9]. Proposing a simplification algorithm working not only on GMMs but on a generic wider class of mixture models, called exponential families, is necessary. Second, they require the user to specify the number of Gaussians (denoted m) used in the simplified model g, the optimal value of m depending both on the initial GMM and on the application.

In this paper, we first describe a novel and efficient algorithm for simplifying GMMs using a generalization of the celebrated k-means quantization algorithm tailored to relative entropy (see section 2). Our algorithm extends easily to *arbitrary* mixture of exponential families. The proposed method is shown to compare favourably well with the state-of-the-art UTAC algorithm both in terms of time and quality performances. Second, we describe an algorithm based on the G-means algorithm [10] who (1) allows to automatically learn the *optimal* number of Gaussians m in the simplified model and (2) provides a progressive representation of the GMM (see section 3).

2 Entropic Quantization of GMMs

2.1 Relative Entropy and Bregman Divergence

The fundamental measure between statistical distributions is the relative entropy, also called the Kullback-Leibler divergence (denoted by KLD). Given two distributions f_i and f_j, the KLD is an oriented distance (asymmetric) and is defined as

$$\mathrm{KLD}(f_i||f_j) = \int f_i(x) \log \frac{f_i(x)}{f_j(x)} \mathrm{d}x. \tag{4}$$

This fastidious integral computation yields for multivariate normal distributions

$$\begin{aligned} \mathrm{KLD}(f_i||f_j) =& \frac{1}{2} \log \left(\frac{\det \Sigma_j}{\det \Sigma_i} \right) + \frac{1}{2} \mathrm{tr} \left(\Sigma_j^{-1} \Sigma_i \right) \\ &+ \frac{1}{2} (\mu_j - \mu_i)^T \Sigma_j^{-1} (\mu_j - \mu_i) - \frac{d}{2} \end{aligned} \tag{5}$$

where $\mathrm{tr}(\Sigma)$ is the matrix trace operator. We can avoid the integral computation using the canonical form of exponential families [11]

$$f_F(x|\tilde{\Theta}) = \exp \left\{ \langle \tilde{\Theta}, t(x) \rangle - F(\tilde{\Theta}) + C(x) \right\} \tag{6}$$

where $\tilde{\Theta}$ are the *natural parameters* associated with the *sufficient statistics* $t(x)$. The *log normalizer* $F(\tilde{\Theta})$ is a strictly convex and differentiable function that specifies uniquely the exponential family, and the function $C(x)$ is the carrier measure. The relative entropy between two distribution members of the same exponential family is equal to the Bregman divergence defined for the log normalizer F on the natural parameter space:

$$\mathrm{KLD}(f_i||f_j) = D_F(\tilde{\Theta}_j||\tilde{\Theta}_i) \tag{7}$$

where

$$D_F(\tilde{\Theta}_j||\tilde{\Theta}_i) = F(\tilde{\Theta}_j) - F(\tilde{\Theta}_i) - \langle \tilde{\Theta}_j - \tilde{\Theta}_i, \nabla F(\tilde{\Theta}_i) \rangle. \tag{8}$$

The $\langle \cdot, \cdot \rangle$ denotes the inner product and ∇F is the gradient operator. For multivariate Gaussian distributions, we consider mixed-type vector/matrix parameters (μ, Σ). The sufficient statistics is *stacked* into a two-part d-dimensional

vector/matrix entity $t(x) = (x, -\frac{1}{2}xx^T)$ associated with the natural parameters $\tilde{\Theta} = (\theta, \Theta) = (\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1})$. The log normalizer specifying the exponential family is [12]

$$F(\tilde{\Theta}) = \frac{1}{4}\mathrm{tr}(\Theta^{-1}\theta\theta^T) - \frac{1}{2}\log\det\Theta + \frac{d}{2}\log\pi. \tag{9}$$

The inner product $\langle\tilde{\Theta}_p, \tilde{\Theta}_q\rangle$ is then a composite inner product obtained as the sum of two inner products of vectors and matrices: $\langle\tilde{\Theta}_p, \tilde{\Theta}_q\rangle = \langle\Theta_p, \Theta_q\rangle + \langle\theta_p, \theta_q\rangle$. For matrices, the inner product is defined by the trace of the matrix product $\Theta_p\Theta_q^T$: $\langle\Theta_p, \Theta_q\rangle = \mathrm{tr}(\Theta_p\Theta_q^T)$. The gradient ∇F is given by

$$\nabla F(\tilde{\Theta}) = \left(\frac{1}{2}\Theta^{-1}\theta\ ,\ -\frac{1}{2}\Theta^{-1} - \frac{1}{4}(\Theta^{-1}\theta)(\Theta^{-1}\theta)^T\right). \tag{10}$$

2.2 Bregman *k*-Means

Banerjee *et al.* [11] extended Lloyd's k-means algorithm to the class of Bregman divergences, generalizing also the former Linde-Buzo-Gray clustering algorithm. They proved that the simple Lloyd's iterative algorithm minimizes *monotonically* the Bregman (right-sided) loss function:

$$\mathrm{LossFunction}_F(\{x_1, ..., x_n\}; k) = \min_{c_1,\ldots,c_k} \sum_k \sum_i D_F(x_i||c_k).$$

where x_i are the source point sets and c_k the respective cluster centroids. A right-sided Bregman k-means is a left-sided differential entropic (*i.e.* KLD) clustering, and vice-versa. Thus, we propose a GMM simplification algorithm based on Bregman k-means. The k-means algorithm is the repetition until convergence of two steps: First, determine membership in clusters (repartition step); second, recompute the centroids. The algorithms 1 and 2 respectively present our right-sided and left-sided Bregman k-means clustering algorithms (denoted BKMC). For these algorithms, $\tilde{\Theta}$ and $\tilde{\Theta}'$ denote natural parameters respectively for GMMs f and g.

2.3 Symmetric Bregman *k*-Means

The BKMC algorithm can be modified in order to use the symmetric Bregman divergence instead of a sided one. Indeed, the use of a symmetric similarity measure is required for common applications such as content-based image retrieval. Given two Gaussians $\tilde{\Theta}_p$ and $\tilde{\Theta}_q$ (natural parameters), the symmetric Bregman divergence SD_F (used in the repartition step) is defined as the mean of the right-sided and left-sided Bregman divergences:

$$SD_F(\tilde{\Theta}_p, \tilde{\Theta}_q) = \frac{D_F(\tilde{\Theta}_q||\tilde{\Theta}_p) + D_F(\tilde{\Theta}_p||\tilde{\Theta}_q)}{2} \tag{15}$$

Algorithm 1. BKMC right-sided(f,m)

1: Initialize the GMM g.
2: **repeat**
3: Compute the cluster C: the Gaussian f_i belongs to cluster C_j if and only if

$$D_F(\tilde{\Theta}_i \| \tilde{\Theta}'_j) < D_F(\tilde{\Theta}_i \| \tilde{\Theta}'_l),\ \forall l \in [1,m] \setminus \{j\} \tag{11}$$

4: Compute the centroids: the weight and the natural parameters of the j-th centroid (*i.e.* Gaussian g_j) are given by:

$$\alpha'_j = \sum_i \alpha_i, \quad \theta'_j = \frac{\sum_i \alpha_i \theta_i}{\sum_i \alpha_i}, \quad \Theta'_j = \frac{\sum_i \alpha_i \Theta_i}{\sum_i \alpha_i} \tag{12}$$

The sum $\sum_i$ is performed on $i \in [1,m]$ such as $f_i \in C_j$.
5: **until** the cluster does not change between two iterations.

Algorithm 2. BKMC left-sided(f,m)

1: Initialize the GMM g.
2: **repeat**
3: Compute the cluster C: the Gaussian f_i belongs to cluster C_j if and only if

$$D_F(\tilde{\Theta}'_j \| \tilde{\Theta}_i) < D_F(\tilde{\Theta}'_l \| \tilde{\Theta}_i),\ \forall l \in [1,m] \setminus \{j\}$$

4: Compute the centroids: the weight and the natural parameters of the j-th centroid (*i.e.* Gaussian g_j) are given by:

$$\alpha'_j = \sum_i \alpha_i, \quad \tilde{\Theta}'_j = \nabla F^{-1}\left(\sum_i \frac{\alpha_i}{\alpha'_j} \nabla F\left(\tilde{\Theta}_i\right)\right) \tag{13}$$

where

$$\nabla F^{-1}(\tilde{\Theta}) = \left(-\left(\Theta + \theta\theta^T\right)^{-1}\theta\ ,\ -\frac{1}{2}\left(\Theta + \theta\theta^T\right)^{-1}\right) \tag{14}$$

The sum $\sum_i$ is performed on $i \in [1,m]$ such as $f_i \in C_j$.
5: **until** the cluster does not change between two iterations.

Similarly, the symmetric centroid c_s is computed from the right-sided and left-sided centroids (respectively denoted c_r and c_l). The symmetric centroid c_s belongs to the geodesic link between c_r and c_l. A point on this link is given by

$$c_\lambda = \nabla F^{-1}\left(\lambda \nabla F(c_r) + (1-\lambda)\nabla F(c_l)\right) \tag{16}$$

where $\lambda \in [0,1]$. The symmetric centroid $c_s = c_\lambda$ verifies

$$SD_F(c_\lambda, c_r) = SD_F(c_\lambda, c_l). \tag{17}$$

A standard dichotomy search on λ allows to quickly find the symmetric centroid c_s for a given precision.

3 Hierarchical GMM Representation

Hamerly and Elkan [10] proposed to adapt the k-means clustering algorithm to learn automatically the number of clusters (parameter k) during the process. Their algorithm, called G-means for Gaussian-means, starts with a small number of centroids (usually 1) and splits iteratively the centroids. G-means repeatedly makes decisions based on the statistical Anderson-Darling test [13]: If the data currently assigned to a centroid follow a normal distribution, then the data are represented by their centroid; otherwise, the data are split into two subsets. The G-means algorithm directly provides a hierarchical clustering of the input data.

In this section, we propose a GMM simplification algorithm based on G-means and BKMC algorithms. This algorithm, named Bregman G-means clustering algorithm (BGMC for short) and described in algorithm 3, first allows to automatically learn the *optimal* number of Gaussians m in the simplified model, and second provides a progressive representation of the GMM. The problem here is to determine if a set of Gaussians (GMM) follows a Gaussian distribution. If so, the set is represented by one Gaussian: its centroid (right-sided, left-sided, or symmetric). Otherwise, the Gaussian set is divided in two subsets. We reasonably assume that a GMM (Gaussian set) is a Gaussian distribution if a large set of l points drawn from this GMM verify the Anderson-Darling test. In our experiments, l was set to $l = 10000$ and the confidence level (here denoted β) used in the Anderson-Darling test was set to $\beta = 95\%$. The algorithm 3 starts with BGMC(N, f, c, α) where N is the root of an empty binary tree, f is a GMM, c is the centroid (right-sided, left-sided, or symmetric) of f, and $\alpha = \sum_{i=1}^{n} \alpha_i = 1$. N_{left} and N_{right} respectively denote the left-child and the right-child of the node N.

The hierarchical structure of the simplified GMM g allows us to introduce the notion of *resolution*, the successive resolutions given a progressive representation of g. Each node of the tree contains a weighted Gaussian. The resolution r corresponds to all the weighted Gaussians contained in nodes of depth r. The resolution 0 corresponds to a GMM containing only one Gaussian: the tree root. The maximal resolution (*i.e.* the tree height) contains all the leafs of the tree. The *optimal* value of m is given by the GMM size at the maximal resolution.

4 Experiments

4.1 Bregman k-Means Clustering

In this section, we compare the influence of the Bregman divergence type (right-sided, left-sided, or symmetric) on the quality of the simplified GMM g. This quality is evaluated through the standard right-sided Kullback-Leibler divergence (KLD) between f and g estimated with a classical Monte-Carlo algorithm [14] since it does not admit any closed-form solution. For this experiment, the initial GMM f is composed of 32 Gaussians and is computed from the image Baboon: First we perform a standard k-means algorithm to gather RGB pixels

Algorithm 3. Calculate BGMC(N, f, c, α)

Store the centroid c and the weight α in the node N.
Draw a set of l points $X = \{x_1, \cdots, x_l\}$ from f.
Split the centroid c into two centroids c_1 and c_2.
Perform a Bregman k-means on c_1 and c_2. Let f_1 (resp. f_2) be the set containing the weighted Gaussians of f closer to c_1 (resp. c_2) than c_2 (resp. c_1). Let α_1 (resp. α_2) be the sum of all the weights of the Gaussians contained in f_1 (resp. f_2).
Compute the projection vector $v = \mu_1 - \mu_2$ where μ_1 and μ_2 are respectively the mean of c_1 and c_2.
Given X and v, use the Anderson-Darling statistical test [13] to detect if f is a normal distribution (at confidence level $\beta = 0.95$).
if f is a normal distribution **then**
    Stop the process; the current node N is a leaf (N_{left} and N_{right} are null).
else
    Compute BGMC(N_{left}, f_1, c_1, α_1).
    Compute BGMC(N_{right}, f_2, c_2, α_2).
end if

in 32 classes, and second we compute each f_i with a standard EM algorithm. The dimension of the Gaussians is 3 (components RGB: red, green, blue).

The figure 1 shows the evolution of the KLD as a function of m (number of the Gaussians in the simplified GMM) for the different Bregman divergence types. First, the KLD decreases with m as expected whatever the Bregman divergence type used. Indeed, the quality of the approximation of the initial GMM f increases with the number of Gaussians in the simplified model g. Second, the left-sided Bregman divergence gives the best results and the right-sided the worst. Indeed, the measure used to evaluate the quality of the simplification is the right-sided KLD. The left-sided Bregman clustering on natural parameters amounts to compute a right-sided KLD clustering on corresponding probability measures. The symmetric BKMC provides better results than right-sided BKMC but worse than left-sided BKMC. In the paper remainder, we will use the left-sided BKMC.

4.2 Method Comparison

4.3 BKMC versus UTAC

The figure 2 shows the evolution of the KLD as a function of m (number of components in the simplified GMM) for algorithms UTAC and BKMC (left-sided). Both algorithms are written in Java. The initial GMM f is computed as in section 4.1. First, whatever the algorithm used (UTAC and BKMC), the KLD decreases with m. Second, BKMC provides the best results and is faster than UTAC: for $m = 16$, the clustering process is performed in 20 milliseconds for BKMC and 107 milliseconds for UTAC on a Dell Precision M6400 laptop (Intel Core 2 duo @ 2.53GHz, 4Go DDR2 memory, Windows Vista 64 bits, Java 1.6). Indeed, BKMC is based on a k-means algorithm which generally quickly converges. UTAC uses a EM method known to slowly converge (*i.e.* within a

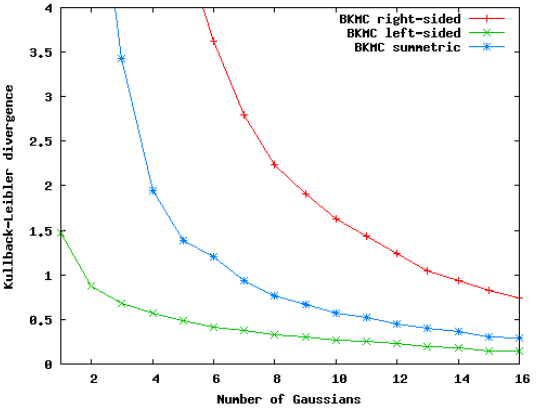


Fig. 1. Evolution of the KLD as a function of m for algorithms right-sided, left-sided, and symmetric BKMC. The left-sided BKMC provides the best approximation of the initial GMM.

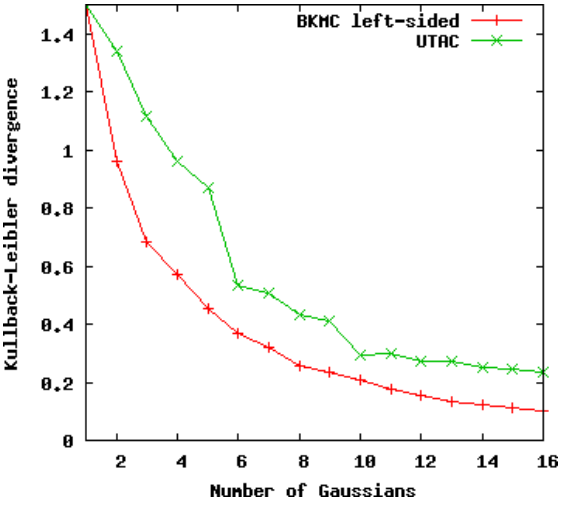


Fig. 2. Evolution of the KLD as a function of m for algorithms BKMC and UTAC

threshold after a large number of iterations). We automatically stop the UTAC process after 30 iterations if the process has not converged.

4.4 Clustering-Based Image Segmentation

In this section, we apply the GMM simplification methods in the context of clustering-based image segmentation problem. Given an image, a pixel x can be considered as a point in $\mathbb{R}^3$. Given a GMM g of m Gaussians, the segmentation is performed by classifying each pixel x to the most probable class C_i:

$$g_i(x) > g_j(x) \ \forall j \in [1, m] \setminus \{i\}$$

This segmentation is illustrated by assigning to the pixel x the value of the class representative μ'_i (see figure 3). The images used for the experiments are Baboon, Lena, Colormap, and Shantytown. The first and second rows show respectively the input image and the segmentation computed from the initial GMM f composed 32 Gaussians. The third and fourth rows show the segmentations computed after the simplification of f respectively with the algorithms UTAC and BKMC. With all images tested, the algorithm BKMC provides the best results (visually and according to the KLD value).

Fig. 3. Application of GMM simplifying algorithms (UTAC and BKMC) to clustering-based image segmentation. The BKMC algorithm provides the best results (visually and according to the KLD value).

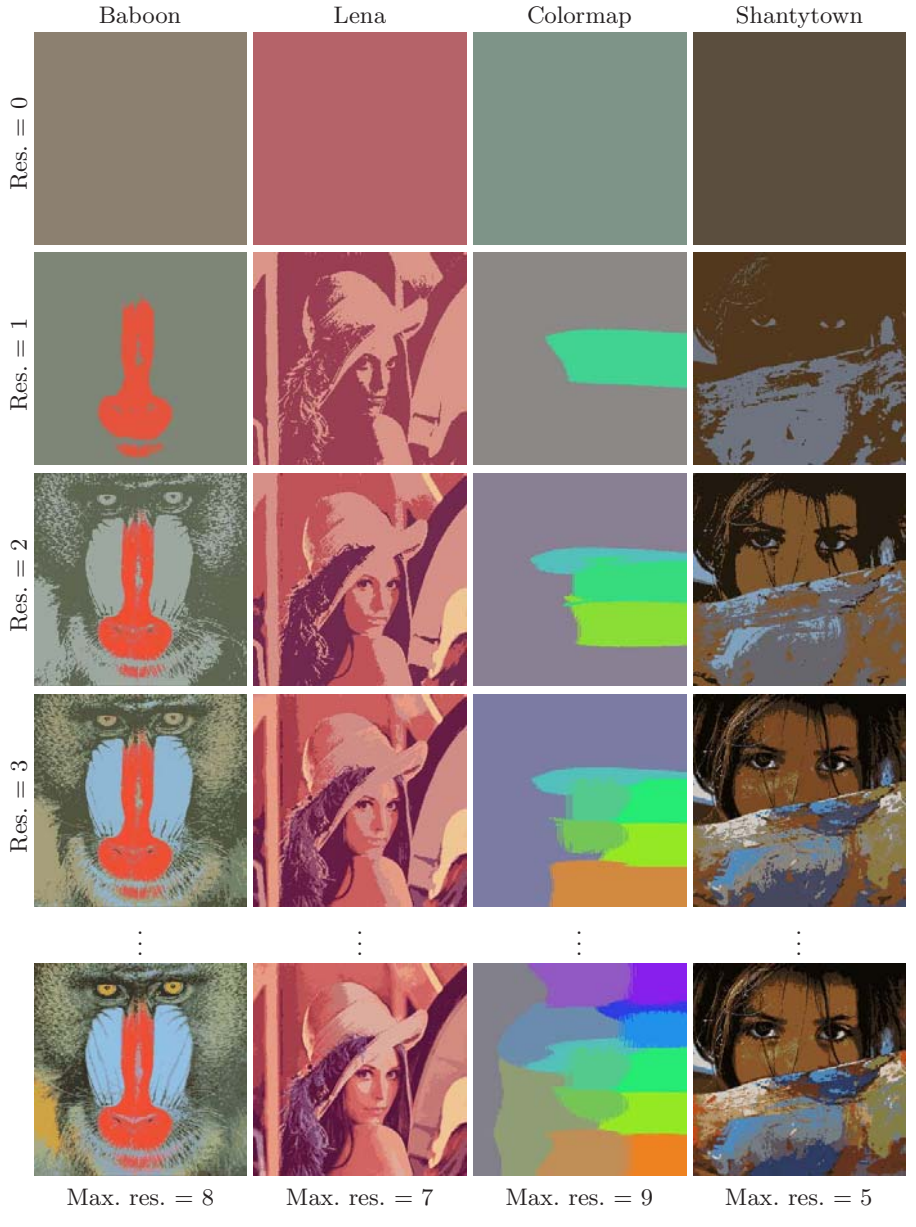


Fig. 4. Application of BGMC algorithm to clustering-based image segmentation. The figure shows (from top to bottom) the simplified GMM from resolution 0 to the maximal resolution. The GMM simplification quality increases with the resolution.

4.5 Hierarchical GMM Representation

In this section, we apply the BGMC algorithm (hierarchical GMM) in the context of clustering-based image segmentation. The figure 4 shows the segmentation obtained from different resolution of the hierarchical GMM. The segmentation quality increases with the resolution. A resolution equal to 0 provides a GMM composed only of one Gaussian: all the pixel of the input image belongs to the same class. The *optimal* value of m is given by the GMM at the maximal resolution. For each image, we give below this optimal value m, the maximum resolution, and the KLD between the initial GMM f and the *optimal* simplified GMM:
- Baboon: $m = 14$, max. res.=8, KLD=0.18
- Lena: $m = 14$, max. res.=7, KLD=0.13
- Colormap: $m = 14$, max. res.=9, KLD=0.59
- Shantytown: $m = 13$, max. res.=5, KLD=0.28

On average, the construction of the hierarchical GMM is performed in 466 ms.

5 Concluding Remarks

In this paper, we have proposed two algorithms for the simplification of Gaussian mixtures models. The first one, named BKMC, is based on the k-means algorithm. Experiments corroborate that BKMC yields better results in shorter computational time in comparison to the state-of-the-art. The second proposed algorithm, named BGMC, is based on the G-means algorithm. BGMC allows to automatically learn the *optimal* number of Gaussians in the simplified model and provides a progressive representation of the initial GMM. Note that although we have presented our algorithms to simplify GMM, our framework is generic and applies to any mixture model of an exponential family. The Java library implementing these algorithms is available at `www.lix.polytechnique.fr/~nielsen/MEF`.

References

1. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society B 39, 1–38 (1977)
2. Parzen, E.: On the estimation of a probability density function and mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)
3. Zhang, K., Kwok, J.T.: Simplifying mixture models through function approximation. In: Neural Information Processing Systems (2006)
4. Goldberger, J., Greenspan, H., Dreyfuss, J.: Simplifying mixture models using the unscented transform. IEEE Transactions Pattern Analysis Machine Intelligence 30, 1496–1502 (2008)
5. Goldberger, J., Gordon, S., Greenspan, H.: An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In: IEEE International Conference on Computer Vision (2003)
6. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE 92, 401–422 (2004)

7. Davis, J.V., Dhillon, I.: Differential entropic clustering of multivariate gaussians. In: Neural Information Processing Systems (2006)
8. Goldberger, J., Roweis, S.: Hierarchical clustering of a mixture model. In: Neural Information Processing Systems (2004)
9. Novoviov, J., Malk, A.: Application of multinomial mixture model to text classification. In: Pattern Recognition and Image Analysis (2003)
10. Hamerly, G., Elkan, C.: Learning the k in k-means. In: Neural Information Processing Systems (2003)
11. Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with bregman divergences. Journal of Machine Learning Research 6, 234–245 (2005)
12. Nielsen, F., Boissonnat, J.D., Nock, R.: On Bregman Voronoi diagrams. In: SIAM Symposium on Discrete Algorithms (2007)
13. Anderson, T.W., Darling, D.A.: Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. In: Annals of Mathematical Statistics (1952)
14. Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (2007)

A Blind Robust Watermarking Scheme Based on ICA and Image Dividing Blocks

Yuqiang Cao[1,2] and Weiguo Gong[2]

[1] Chongqing Communication Institute, China
[2] College of Optoelectronic Engineering, Chongqing University , China
caoyuqiang@live.cn

Abstract. We propose a new scheme for the oblivious/blind robust watermarking of digital images based on independent component analysis (ICA). Most watermarking schemes based on ICA need additional information in watermark extraction process. But it is not encouraged because storing and transferring these additional information is not very convenient in some situations. A novel dividing image blocks is utilized in the paper that overcomes the shortcoming that watermark extraction based on ICA need additional information. The watermarking scheme, undergoing a variety of experiments, has shown its robustness against many attacks, e.g. JPEG, filter, gray scale reduction etc; it also exhibits a capability in image authentication.

1 Introduction

With the rapid spread of computer networks and the further development of multimedia technologies, digital contents can be accessed easily, and the protection of intellectual property becomes more and more important every day. A digital watermark embeds an imperceptible signal into data such as audio, video and images, for a variety of purposes, including copyright control and data authentication. As watermarking is increasingly used for a wide variety of applications, various properties of watermarks, such as how they respond to common signal transformations or deliberate attack, have become important considerations. The watermarking techniques can be fragile, robust, or semi-fragile [1]. Fragile watermarks do not survive lossy transformations to the original host signal and their purpose is tamper detection of the original signal. Ideally, an effective, robust watermarking scheme provides a mark that can only be removed when the original content is destroyed as well. Typically, many of the applications for copyright protection involve relatively high quality original content and the imperceptibility criterion is critical for such applications. Semi-fragile watermarking techniques differentiate between common signal transformations and deliberate attack. It is robust to common signal transformations attack but is fragile to deliberate attack. In order to estimate the watermark, some existing methods require the original image at the extraction process. Some others need a priori knowledge of the watermark for the extraction. It is not encouraged since original work should be restricted from public access, and watermark information is not always fixed in advance [2]. This is called non-blind watermarking that watermark extraction algorithms need use the original unwatermarked data to find the watermark. When the watermark

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 526–535, 2010.

extraction algorithm has access to the original unwatermarked data, this renders the watermark extraction more difficult. Watermarking algorithms of this kind are referred to as public, blind, or oblivious watermarking algorithms.

In this paper, we develop a simple method for the blind robust watermarking of digital images based on independent component analysis (ICA). Independent component analysis (ICA) is an important technique in signal processing field for estimating unknown signals from their observed mixtures [3]. With its blind separation capability, several authors have tried to apply ICA to watermarking. When applied to watermarking, ICA presumes the watermarked work as a mixture of the original work and the watermarks, and therefore, it can do separation to estimate the watermark. In the past ICA based framework has been used for multimedia watermarking [4–6]. Thang Viet Nguyen et al in [4] have proposed a approach called WMicaT that employs ICA. But their detector uses a 'public image' during watermark extraction process. The size of the public image is as big as the size of the original image, therefore, storing and transferring this supporting image is not very convenient in some situations. J.J. Murillo-Fuentes in [5] has presented the blind robust watermarking of digital images based on ICA. But watermarking capability was too small that 64 bits watermarking had only hid in the 512×512 gray-scale images. So it didn't meet the requirements of significative watermarking, e.g. logos image, signature, fingerprint etc. In the paper the blind watermarking is not only robust against many attacks included Jpeg compress, filtering, gray-scale reduction, tampering attack, but has a large capability so that the watermark can be any meaningful image, logo etc.

The outline of this paper is as follows: In section 2, we review basic ICA theory and present a general ICA watermarking scheme. In section 3, the new blind watermark embedding scheme based on ICA is proposed. In section 4, we provide watermark extraction scheme. In section 5, experimental results are presented. Finally, a conclusion is given, and future research direction is proposed.

2 Watermarking Using ICA

The ICA technique [3], which consists of recovering a set of unknown sources from their instantaneous mixtures, is an important technique in signal processing. An ICA model shown in Fig.1 can be divided into two sub-models: mixing model and demixing model. The observations $x_1, x_2 \cdots x_N$ are assumed to be linear mixtures of M hidden statistically independent signals $s_1, s_2 \cdots, s_M$. The mixing model can be expressed as

$$x = As\,. \tag{1}$$

Where $x = [x_1, x_2, \cdots, x_N]^T$, $s = [s_1, s_2, \cdots, s_M]^T$ and $A_{M\times N}$ is an unknown mixing matrix.

To estimate the unknown signals s_i, we have to build a demixing model, i.e., to compute a demixing matrix B. ICA carries out this task by maximizing the statistical independence criteria among the outputs $y_1, y_2 \cdots y_M$. When converged, the outputs

$y_1, y_2 \cdots y_M$ will be a permutation of the unknown sources $s_1, s_2 \cdots, s_M$. The demixing model, therefore, can be formulated as

$$y = Bx \,. \tag{2}$$

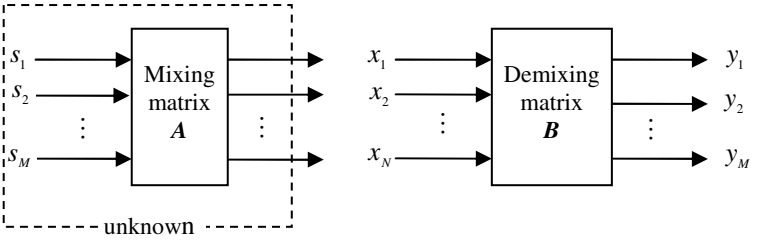


Fig. 1. The ICA mixing and demixing model

To able to acquire optimal estimate y, the identifiably constraints are outlined as,

(1) Unknown source $s_1, s_2 \cdots, s_M$ are statistically independent and non-gaussian;

(2) Number of observed linear mixtures (sensors) N m must be greater or equal to the number of independent components (sources) M, i.e. $N \geq M$;

(3) The rank of the mixing matrix, A, must be full column rank.

The ICA model and a watermarking model is similar so we can design watermarking scheme based ICA. Watermark signal can be significative logo, image or audio etc, and media signal protected can be image, audio or video etc. they most are non-gaussian and statistically independent. It is also easy to construct a full column rank A. Watermark embedding is looked as different sources mixing, the number N of observations $x_1, x_2 \cdots x_N$ must be greater or equal to the number of independent components (sources) M, how to produce $x_1, x_2 \cdots x_N$ in the watermarking scheme become more important.

3 Watermark Embedding Scheme

A complete watermark embedding scheme is shown in Fig. 2. Assume matrix I be a gray-scale image of size $N \times M$. This matrix can be divided into 4 blocks $S_p(i, j)$, p=1, 2, 3, 4. Matrix I is divided into 4 blocks $S_p(i, j)$ according to,

$$\begin{aligned} S_1(i, j) &= I(2\times i, 2\times j)\,, \\ S_2(i, j) &= I(2\times i-1, 2\times j)\,, \\ S_3(i, j) &= I(2\times i, 2\times j-1)\,, \\ S_4(i, j) &= I(2\times i-1, 2\times j-1)\,. \end{aligned} \tag{3}$$

Where $i = 1, 2, \cdots, N/2$, $j = 1, 2, \cdots, M/2$.

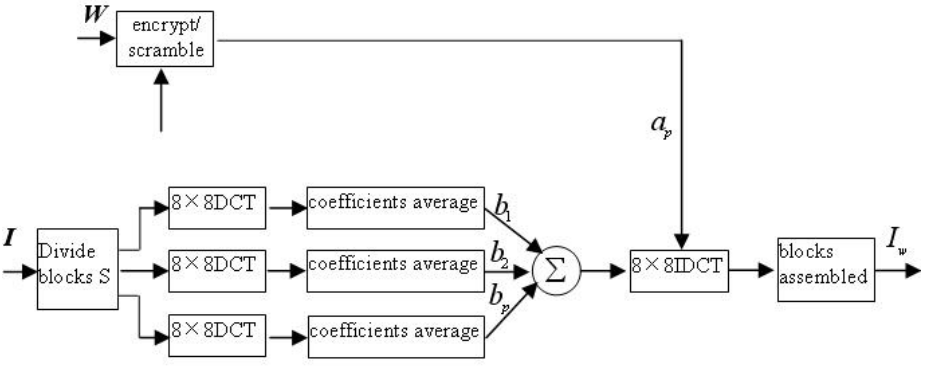


Fig. 2. Watermark embedding scheme

The four sub-images are very similar, so we designed that the sub-images mixed linearly with watermark signal. It is the base of detect watermark using ICA. The four sub-images are decomposed into non-overlapping 8×8 blocks $B_{m,n}^{p}$ according to,

$$B_{m,n}^{p}(i,j) = S_p((m-1)\times 8+i,(n-1)\times 8+j). \tag{4}$$

Where $i, j = 1,2\cdots 8;\ \ m = 1,2\cdots M/8$, $p = 1, 2, 3, 4$.

The DCT is performed independently for every block $B_{m,n}^{p}$, and the DCT coefficients matrixes are denoted by $C_{m,n}^{p}$. Four coefficients are selected to embed watermarks in every $C_{m,n}^{p}$, to balance the robustness and transparency, we select intermediate frequency coefficients (3, 1), (4, 1), (3, 2), (2, 3). Calculate J according to,

$$\begin{aligned}
J_{m,n}(3,1) &= (C_{m,n}^{1}(3,1)+\cdots+C_{m,n}^{4}(3,1))/4,\\
J_{m,n}(4,1) &= (C_{m,n}^{1}(4,1)+\cdots+C_{m,n}^{4}(4,1))/4,\\
J_{m,n}(3,2) &= (C_{m,n}^{1}(3,2)+\cdots+C_{m,n}^{4}(3,2))/4,\\
J_{m,n}(2,3) &= (C_{m,n}^{1}(2,3)+\cdots+C_{m,n}^{4}(2,3))/4.
\end{aligned} \tag{5}$$

The watermark W values are 0 or 1; it is encrypted or scrambled with secret key k to enhance security of watermark. The watermark W are inserted into the coefficients to form the new coefficients given by

$$V_{m,n}^{p} = J_{m,n}\times a_p + sign(J_{m,n})\times b_p \times W(k). \tag{6}$$

$sign(\ )$ is sign function, a_p, b_p denote embedding strengths factors, the values of these two parameters will determine how strongly the watermarks are embedded. $V_{m,n}^{p}$ Replace $C_{m,n}^{p}$ and IDCT is performed for every block to obtain watermarked image I_w.

4 Watermark Extraction Scheme

The goal of the extraction scheme is to extract the watermark W from the watermarked image I_w. Besides the watermarked image, the other information available to us is the secret key k, but k is non-correlative to watermark extraction. So the watermark extraction algorithm is blind. The extraction scheme can be divided into two stages. The first stage is to extract the estimated watermark W from the watermarked image I_w. In the second stage, a post processing scheme is applied to obtain the optimal W from the estimated watermark.

4.1 Estimate Watermark Based on ICA

For estimating watermark from watermarked image, the steps for watermarking extraction are as follows:

(1) like as embedding process, I_w is divided into 4 blocks, and 8×8 block DCT is performed independently for every block to obtain coefficient matrixes $C'^{p}_{m,n}$;

(2) The $C'^{p}_{m,n}(3,1)$, $C'^{p}_{m,n}(4,1)$, $C'^{p}_{m,n}(3,2)$, $C'^{p}_{m,n}(2,3)$ are converted into one dimension vectors V^1, V^2, V^3 and V^4;

(3) Applying Fastica[8] technique on V^1, V^2, V^3 and V^4 each other. Equation (7) clearly matches the ICA mixing model $y = Bx$. Though signal sources are only W and J in embedding process. IDCT is performed for every block, integer quantization must be applied to save image format. So quantization noise n is imported, signal source are not only W and J. Different V components are performed Fastica to obtain different estimate watermark w_i.

$$\begin{bmatrix} W \\ J \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ a_j & b_j \end{bmatrix}^{-1} \begin{bmatrix} V^i \\ V^j \end{bmatrix},$$

$$\begin{bmatrix} W \\ J \\ n \end{bmatrix} = \begin{bmatrix} a_i & b_i & x_i \\ a_j & b_j & x_j \\ a_q & b_q & x_q \end{bmatrix}^{-1} \begin{bmatrix} V^i \\ V^j \\ V^q \end{bmatrix}, \tag{7}$$

$$\begin{bmatrix} W \\ J \\ n_1 \\ n_2 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & x_1 & y_1 \\ a_2 & b_2 & x_2 & y_2 \\ a_3 & b_3 & x_3 & y_3 \\ a_4 & b_4 & x_4 & y_4 \end{bmatrix}^{-1} \begin{bmatrix} V^1 \\ V^2 \\ V^3 \\ V^4 \end{bmatrix}.$$

4.2 The Post-processing Scheme

The ICA technique, however, only provides a set of images that contains the watermark, but is not able to identify which one is the estimate of the watermark. It means

that the output Y_1 does not necessarily correspond to the estimate of watermark W. It can be the estimate of any one of the four source signals W, J, n_1 or n_2. For this reason, in the second stage of the extraction scheme, we develop a post-processing algorithm to obtain the watermark from the images Y_1, Y_2, Y_3 and Y_4. We apply the correlation coefficients in post-processing scheme to identify which output, Y_i, corresponds to which source signal.

The detail scheme includes two stages, an identifying stage and a refining stage. The first stage filters out the watermarks from the four image inputs. The second stage uses the estimated watermarks to extract the optimal watermark. To identify the watermarks, we use the correlation coefficients between each output and the watermarked image. Let us consider two images X and Y, each of size $M \times N$. The absolute correlation coefficient $|r_{X,Y}|$ measures the similarity between two images X and Y. When two images are totally different $|r_{X,Y}| \approx 0$, and, on the other hand, when X and Y are identical to each other $|r_{X,Y}| \approx 1$. Assume the threshold T, when $|r_{X,Y}| > T$, we judge it is the watermark component w. After successfully estimating the every watermark w_i, we compute the estimate of the optimal watermark as the average of these watermarks, given by

$$W' = \frac{1}{l}(w_1 + w_2 + \cdots + w_l). \tag{8}$$

5 Experimental Performance Analysis

5.1 Simulate Setup

The original images lena are gray-scale images of size 512 × 512 with 256 intensity levels. The watermark are binary images: the Chongqing university logo is of size 64×64. The embedding strengths a_p and b_p were controlled so that the watermarked images have a high quality in term of the peak signal-to-noise ratio (*PSNR*). The peak signal-to-noise ratio between an original image I and the modified image I_w is computed by

$$PSNR = 20\log 10\left(\frac{255}{\sqrt{\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - I_w(i,j))^2}}\right). \tag{9}$$

Where $I(i,j)$ and $I_w(i,j)$ denote the (i,j) th pixel intensity (gray) level of the original and watermarked images, respectively. The watermark embedding values used for different parameters in the experiments are provided in Table 1. With the chosen parameter values, the watermark was generated and embedded into the host images. We are able to produce a high quality watermarked images (PSNR > 38 dB). The

Table 1. Experimental parameters

	$p=1$	$p=2$	$p=3$	$p=4$	*PSNR*	$\lvert r_{W,W'} \rvert$
a_p	5	7	9	30		
b_p	0.9	0.85	0.8	0.2	38.1779	0.995

Fig. 3. The images used in the experiments. From left to right: original image (I), watermark (W), watermarked image(I_w),optimal watermark W .

watermarked images are almost identical to the original ones and the embedded marks are invisible to normal eyes.

Next, we evaluated and compared the robustness of embedded watermark. In the experiments, we applied various types of attacks at different strengths to the watermarked images, i.e. JPEG compression, Gray scale reduction, filtering, brightness & contrast enhancement, and computed the $\lvert r_{W,W'} \rvert$ from the retrieved watermarks. For further investigation, we compared the proposed scheme with other watermarking techniques that work on different processing domains. These techniques include discrete cosine transform algorithms Cox-DCT [9], spatial-domain algorithms Thang Viet Nguyen[4], and discrete wavelet transform algorithms Kundur-DWT [10].

5.2 JPEG Compression Test

The watermarked image I_w was compressed using JPEG compression tool with different quality factors (from 100% down to 10%). The watermark extraction was done on the compressed image. The performance index $\lvert r_{W,W'} \rvert$ was computed for each quality factor and is shown in Fig. 4. As it is shown, the proposed algorithm provided very good performance on all experiments. The qualities of the estimated watermark were high even when the JPEG quality factor was lowered drastically. The proposed scheme outperforms the blind watermarking scheme Kundur-DWT[10], and inferior to on DCT-based method (the Cox-DCT)[9] and WMICA[4], because the last two schemes are non-blind.

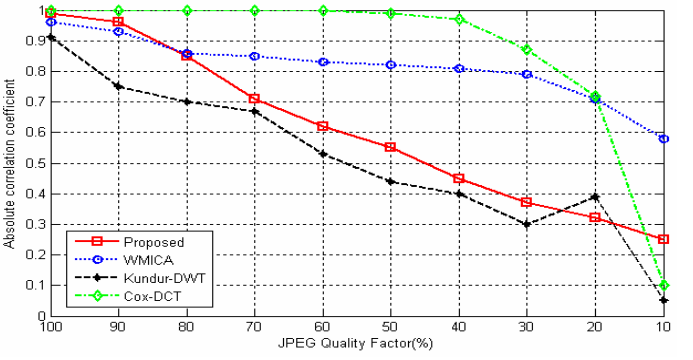


Fig. 4. Performance comparison for JPEG compression test

5.3 Gray Scale Reduction Test

In this test, the intensity (gray) level of the watermarked image I_w was reduced from original 256 level down to 128, 64, …, 4 level. As it is shown in Fig.5, the algorithm offered excellent results in the gray-scale reduction test. The performance index $|r_{W,W'}|$ was high, showing a strong correlation between the estimated image and the watermark. It can be seen that our scheme was able to extract the watermark successfully where the gray level is greater or equal 4, and its performance outperforms other schemes.

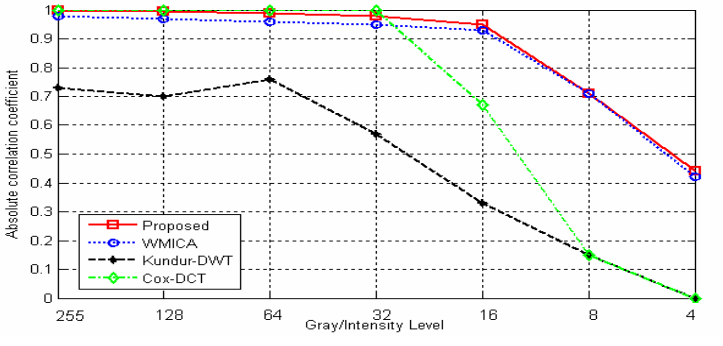


Fig. 5. Performance comparison for gray-scale reduction test

5.4 Other Attacks

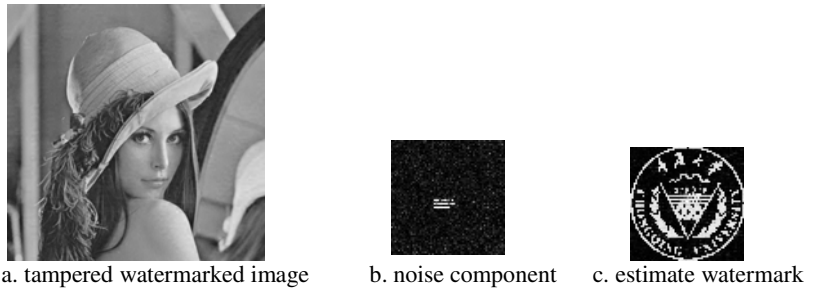
Unlike some methods that are only robust against several specific attacks, the proposed method provides a steady performance throughout all the tests. The Table.2 summarizes the detection results against contrast enhancement, gaussion noise, median filtering and resize attacks respectively. The results show our scheme is robust.

Table 2. The watermark detect result for other attacks

attack	Contrast 50%	Gaussion noise	Median filtering	Scale 50%
PSNR	19.3	27	32	30
$\lvert r_{w,w'} \rvert$	0.89	0.75	0.76	0.78

5.5 Authentication Test

To evaluate the validity of the proposed image authentication scheme, the tampered image, shown in Fig. 6, was used. To forge a tampered image of the embedded image, the hat in the watermarked image is inserted into a flower. The detection result of the tampered image is shown in Fig.6. The Fig denote noise component, it shall distribute averagely if the watermarked image didn't been modified. When watermarked image did been modified, the tampered area is clear noticeable, with the pixel values of the tampered area are much higher than the others.



a. tampered watermarked image b. noise component c. estimate watermark

Fig. 6. Tampering attack test

6 Conclusion

In this work, we have proposed a robust watermarking scheme based on ICA technique. The most advantage is that watermarking scheme blind and don't need any additional signal, and the robustness and capability don't been reduced. The watermark embedding and extraction process is simple the proposed scheme also makes it quite a useful tool for authentication. Experimental results have demonstrated that the proposed watermarking is robust with respect to some important attacks. The next work is that the quality of watermarked images need further improvement, geometrical attack in the watermarking scheme need further research.

Acknowledgements

The work on this paper was supported by Natural Science Foundation Project of CQ CSTC, 2008BA0018.

References

1. Miller, M.L., Cox, I.J.: A review of watermarking principles and practices. In: Digital Signal Processing in Multimedia Systems, pp. 461–485. Marcell Dekker Inc., New York (1999)
2. Langelaar, G.C., Setyawan, I., Lagendijk, R.L.: Watermarking digital image and cideo data. IEEE Signal Processing Magazine 9, 1053–5888 (2009)
3. Cichocki, A., Amari, S.-I.: Adaptive Blind Signal and Image Processing. John Wiley & Sons, Chichester (2002)
4. Nguyen, T.V., Patra, J.C.: A simple ICA-based digital image watermarking scheme. In: Digital Signal Process (2007), doi:10.1016/j.dsp.2007.10.004
5. Murillo-Fuentes, J.J.: Independent component analysis in the blind watermarking of digital images. Neurocomputing 70, 2881–2890 (2007)
6. Malik, H., Khokhar, A., Ansari, R.: Improved Watermark Detection for Spread-Spectrum Based Watermarking Using Independent Component Analysis
7. Tong, M., Feng, W., Ji, H.: A Robust Geometrical Attack Resistant Digital Image Watermarking Based on FastICA Algorithm. In: IEEE 2008 Congress on Image and Signal Processing (2008), doi:10.1109/CISP.2008.47
8. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. 10(3), 626–634 (1999)
9. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.G.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. 6(12), 1673–1687 (1997)
10. Kundur, D., Hatzinakos, D.: Digital watermarking using multiresolution wavelet decomposition. In: Int. Conf. Acoust. Speech Signal Process (ICASSP 1998), Washington, USA, vol. 5, pp. 2969–2972 (1998)

MIFT: A Mirror Reflection Invariant Feature Descriptor

Xiaojie Guo, Xiaochun Cao, Jiawan Zhang, and Xuewei Li

School of Computer Science and Technology
Tianjin University, China
{xguo,xcao,jwzhang,lixuewei}@tju.edu.cn

Abstract. In this paper, we present a mirror reflection invariant descriptor which is inspired from SIFT. While preserving tolerance to scale, rotation and even affine transformation, the proposed descriptor, MIFT, is also invariant to mirror reflection. We analyze the structure of MIFT and show how MIFT outperforms SIFT in the context of mirror reflection while performs as well as SIFT when there is no mirror reflection. The performance evaluation is demonstrated on natural images such as reflection on the water, non-rigid symmetric objects viewed from different sides, and reflection in the mirror. Based on MIFT, applications to image search and symmetry axis detection for planar symmetric objects are also shown.

1 Introduction

Local image feature is a prerequisite in the areas of computer vision such as image retrieval [1], 3-D reconstruction, object recognition [2], camera calibration, robot navigation, gesture recognition, image search [3] and building panoramas [4]. To handle image variability caused by rotations, scale changes, occlusions, varying illuminations and even perspective distortions, numerous techniques such as Harris corner detector [5], SIFT [6], SURF [7], GLOH [8] and DAISY [9] are purposed. Harris corner detector can find feature points with invariance of rotation and illumination. Actually, what it detects are not just corners but also points that have great gradients in multiple directions at a fixed scale. In other words, scale changing is not handled. SIFT makes full use of the gradient property and utilizes a spatial weighting scheme to differentiate pixels in the corresponding collection, i.e. the histogram of each pixel in the collection is weighted by the gradient magnitude and a Gaussian weighted window. SURF adopts Fast-Hessian detector to quickly detect features within images which owes much to the integral images and uses the Harr wavelet response to capture the texture properties around interest points. Histogram of Oriented Gradients (HOG) [10] uses the distribution of gradients within a set of pixels nearby the point of being calculated. DAISY [9] is another fast solution of local image descriptor designed for dense wide-baseline matching purpose, and is intrinsically tolerant to rotations due to its circular interest region design.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 536–545, 2010.

(a) (b)

Fig. 1. Comparison of matching between MIFT and SIFT in the mirror reflection situation. The left image is the Red Wings logo from movie "Ferris Bueller's Day Off", the right one is the horizontally reflected version of it. (a) Matching result of MIFT. (b) Matching result of SIFT.

Although so many different schemes of feature extraction have been proposed, it is far from complete and robust enough to solve all of the problems encountered in the real life. While those above descriptors are successful in handling most of distortions and illumination variances, they fail in the situation of mirror reflection as shown in Fig. 1. Since the invariance of SIFT is remarkably robust, many works have been carried out based on it, such as the 3D SIFT descriptor [11] that is proposed to recognize actions and PCA-SIFT [12] which aims at creating a more robust and shorter descriptor to image deformations than SIFT. The purposed descriptor MIFT improves the invariance to mirror reflection and performs as well as SIFT when there is no mirror reflection. Particularly, we reorganize the structure of SIFT descriptor, and also adjust the matching strategy accordingly.

2 MIFT

MIFT improves SIFT to advance the invariance of features in mirror reflection situations. Note that the proposed solution is able to be used in other SIFT-like descriptors with minor changes. We choose SIFT as basis, because of its outstanding stability, robustness and distinctiveness.

2.1 Mirror Reflection

In the real world, mirror reflection generally appears in three different ways: horizontal reflection, vertical reflection, and combined reflection which is horizontally and vertically reflected. However, for rotation invariant descriptors, the horizontal reflection and the vertical reflection are equivalent by rotating the coordinate system. The same argument holds true for the original and the combined reflection. Therefore two cases, i.e. the original image and the horizontal reflected one, need to be handled. The relationship between the same region after specifying the dominant orientation in the original image and in the horizontally reflected one is that the row order of pixels and cells is identical and the column order inverse, i.e. the gradients of the same pixel in the interest region after orientation assignment in different reflection cases are approximate[1] bilateral symmetry.

[1] The noises and distortions might influence the gradients.

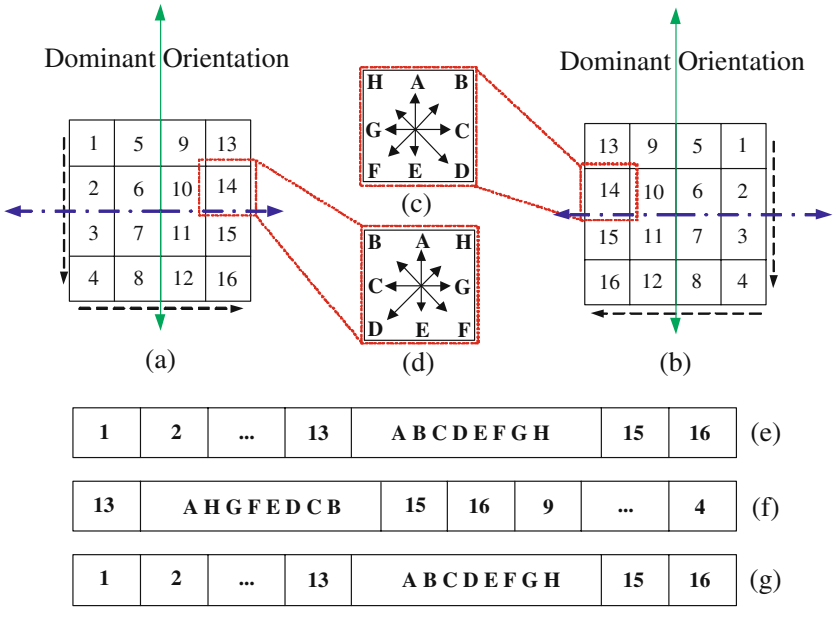


Fig. 2. Illustration of the descriptor organizations of MIFT and SIFT in the situations with and without mirror reflection. (a) A keypoint and its interest region in the original image. (b) (a) in the horizontally reflected image. (c) Distribution of eight orientations in the 14^{th} cell of (b). (d) Distribution of eight orientations in the 14^{th} cell of (a). (e) SIFT and MIFT descriptor for (a). (f) SIFT descriptor for (b). (g) MIFT descriptor for (b).

2.2 Descriptor Reorganization

A vector containing the magnitudes of all the orientation histogram entries in a region around the keypoint forms a descriptor, and a 4×4 array with 8 orientation bins in each is proved to be the successful format in [6]. As SIFT, a 128-D, i.e. $4 \times 4 \times 8$, vector is also chosen to support MIFT descriptor, although 128-D is not compulsory. Figure 2 (a) is a keypoint with its interest region in the original image, and Figure 2 (b) is (a) in the horizontally reflected image, both of which are after specifying dominant orientation. Note that there is no limitation about which direction the mirror reflection axis is alone because of the rotation invariance.

Order of 16 Cells. SIFT uses a fixed order to organize the 16 cells in the interest region after specifying the dominant orientation out of 36 candidates. Note that there might be multiple descriptors for the same combination of location and scale, because it is possible that multiple peaks of 36 orientations are close to the highest peak. As shown in Fig. 2 (a), SIFT might adopt the column-major-order encoding strategy. As a result, the 16 cells are ordered as Fig. 2 (e). However, the column order is reversed after mirror reflecting as shown in Fig. 2 (b). The original fixed encoding strategy used in SIFT would arrange the 16 cells as Fig. 2 (f). Although this encoding strategy is invariant to rotation and scale,

and even tolerant to affine transformation, it does not result in the same order in the situation of mirror reflection. We introduce an adaptive encoding technique that is able to preserve the order of the 16 cells in the mirror reflection case. Fortunately, there are only two directions to choose from, from left to right and vice versa, as the order of the rows are the same in the column-major-order encoding strategy. Intuitively, two magnitudes of the directly left- and right-pointing orientations (the blue dashed arrows) can be used to select which direction goes first. However, this method is sensitive to be relied on due to noises and distortions. The direction is instead decided based on the summations of the magnitudes of all left-pointing and right-pointing orientations,

$$m_r = \sum_{k=1}^{(N_{bin}-2)/2} L_{(n_d-k+N_{bin})\%N_{bin}}, \tag{1}$$

$$m_l = \sum_{k=1}^{(N_{bin}-2)/2} L_{(n_d+k+N_{bin})\%N_{bin}}, \tag{2}$$

where N_{bin} is the number of orientation bins that is 36 for MIFT, n_d is the dominant orientation index and L_i is the gradient magnitude in the $\frac{i\times 2\pi}{N_{bin}}$ direction. According to this measurement, we adaptively change the encoding strategy from the fixed order to the one indicated by the winner of m_l and m_r. Theoretically, this new encoding approach makes the sequence of cells unique in whichever case as shown in Fig. 2 (g). Nevertheless, m_l and m_r might be close to each other due to noises, lighting variations and other factors. To be more robust in such situations, an extra descriptor is created in opposite order to support the feature, when $\min\{m_l, m_r\} > \tau \max\{m_l, m_r\}$. In our experiments, τ is 70% for all the evaluations.

Order of 8 Orientations. Reorganization of cells described above is just one of two essential steps. The other equally significant procedure is to restructure the order of orientation bins in each cell. As shown in Fig. 2 (c) and (d), all gradients in each cell are divided into their nearest bins of eight directions. Figure 2 (c) and (d) present the same cell in the cases with and without mirror reflection. Main difference between them is order change of bins without serious influence on their strengths. Therefore, we encode them in anticlockwise order beginning with 'A' in the case of Fig. 2 (a), and in clockwise beginning with 'A' in the case of Fig. 2 (b) based on the comparison of m_l and m_r.

Finally, for the same keypoint, we obtain a unique descriptor in different mirror reflection cases by the method described above. For the features in the situations of Fig. 2 (a) and (b), the descriptors are identical which are shown in Fig. 2 (e) and (g).

2.3 Matching

Due to the change of the signature strategy described in Section 2.2, every descriptor from SIFT may map to one or two from MIFT, decided by the

similarity between m_l and m_r. Recall the original matching method of SIFT just compares the closest and the second closest descriptors. Therefore, it may miss pairs that should be on the list of good matches since MIFT might have multiple similar descriptors for one keypoint. To reduce the miss rate, we introduce an improved matching method. Algorithm 1, IMM, gives pseduo-code for the improved matching method.

Algorithm 1. Improved matching method (IMM)

Require: $k = 1$
Ensure: **match**(i)=IMM (**des1**(i), **des2**)
  [**vals**, **indices**]= Sort (arccos(**des1**$(i)^T$**des2**), ascend)
  repeat
    $k = k + 1$
    if **vals**$(1) <$ *distRatio* $\times$ **vals**(k) **then**
      match(i) = **indices**(1)
      break
    else
      match$(i) = 0$
    end if
  until The location and scale combinations of **vals**(1) and **vals**(k) are not identical

In the IMM, distRatio is set the same as SIFT, **des1**(i) is the i^{th} descriptor in the set of the comparing image descriptors and **des2** is the matrix containing all of the compared image descriptors. According to the characteristic of the matching strategy, the IMM aims at preventing potential true matches from ignoring due to similar matched descriptors for the same keypoint, we consider different combinations of location and scale as different keypoints, alternatively the descriptors at the same location and scale may be probably very close. Theoretically, the matching strategy needs to constrain the influence from close matched descriptors of the same keypoint. The improved matching performance is demonstrated in Section 3.

3 Results

MIFT aims at advancing state-of-the-art descriptors to be invariant to the mirror reflection. In order to show the benefit of MIFT, we tested the performance on natural images with and without mirror reflection against SIFT.

To quantify the performance of MIFT compared to SIFT in mirror reflection situations, we choose some typical scenes including symmetric image pairs, reflection on the water, non-rigid symmetric objects viewed from different sides and reflection in the mirror as shown in Fig. 3. The comparison results are shown in Fig. 4 (a). For the same four pairs of natural images with mirror reflection, SIFT fails to find reasonable numbers of matches, but MIFT finds (78, 70, 57, 36) matches, with very few mismatches (6, 2, 2, 0).

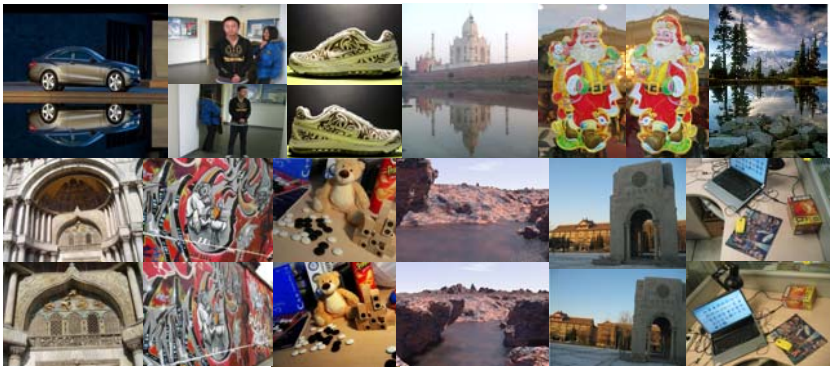
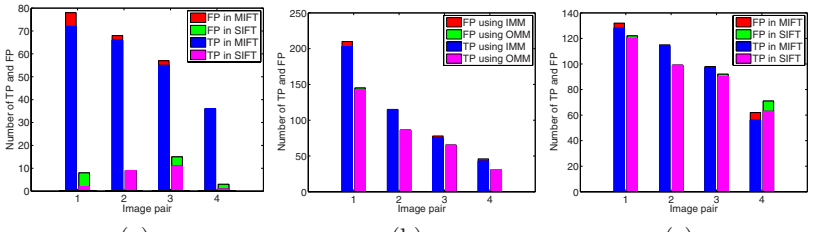


Fig. 3. Example images from image data set for the comparison between MIFT and SIFT. **Top two rows:** test images with mirror reflection. **Bottom two rows:** test images without mirror reflection.



(a) (b) (c)

Fig. 4. Comparisons between true positive matches (TP) and false positive ones (FP). (a) Comparison between MIFT and SIFT for image pairs with mirror reflection. (b) Comparison between the improved matching method (IMM) and the original matching method (OMM). (c) Comparison between MIFT and SIFT for image pairs without mirror reflection.

As described in Section 2.3, the improved matching method aims at reducing the miss rate. As shown in Fig. 4 (b), we plot the differences between improved matching method and the original method referring to the method used in SIFT. For the same four pairs of images which are randomly selected from the data set with and without mirror reflection, the true positive matches increase a lot from (143, 86, 65, 31) to (203, 115, 75, 43). Simultaneously, the mismatches increase also for the test pair 1, 3, 4. However, the positive numbers are significantly bigger than the negative ones.

We also test and verify the performance of MIFT compared with SIFT on natural images without mirror reflection. Some of them are shown in Fig. 3. As shown in Fig. 4 (c), the numbers of the true positive and false positive matches of MIFT are similar to those of SIFT. However, for most cases of Fig. 4, MIFT performs slightly better than SIFT due to the characteristic of the IMM, i.e. the descriptors obtained at the same location and scale are considered as the same keypoint and are ignored.

4 Applications

4.1 Application to Image Search

Image feature is the basis of image understanding for computer, and it is the fundament of many multimedia applications such as image search from mass image data. No doubt directly comparing query images with the images in database is the simplest method. In addition, in [3], the authors present two ways to search images. 1) Hamming embedding (HE) and 2) weak geometric consistency constraint (WGC). These two approaches are based on the bag-of-feature method with k-means algorithm. However, all these methods can't solve the flip detection effectively but implement SIFT twice for every query image ($SIFT_1$ for the original query images, and $SIFT_2$ for the manually flipped ones).

Duplicate computation of SIFT for every query image is rather laborious and inflexible, we apply MIFT to address this tough problem. Note that we just implement the directly comparing method, however HE and WGC can be easily transplanted to MIFT from SIFT.

The image data are from INRIA Holidays dataset[2], and we randomly select 300 images from INRIA Holidays dataset as our database, and another 36 images as query set. The one third of the query images are without mirror reflection (NM), another one third are with horizontal mirror reflection (HM) and the remaining one third are with vertical mirror reflection (VM).

We separately use MIFT and SIFT to detect and describe features for every image from our database and query set. As shown in Table 1, MIFT successfully searches 35 images out of 36 in which 34 successfully registered images are first-ranking and 1 image second-ranking. There is only 1 failed to find the related images. This very image and its corresponding one in the database are shown in Fig. 5. For SIFT, as shown in Table 1, it only correctly finds 11 non-reflection images out of 36 for the first process which contains 12 images without mirror reflection, and for the second process it finds 23 first-ranking images with horizontal or vertical mirror reflection, and 1 third-ranking which is second-ranking using MIFT. The incorrect registration image pair is the same as MIFT, as shown in Fig. 5.

The reason why the left image pair in Fig. 5 is not first-ranking is that there are too many repetitive structures within the images that lead to close arc cosine values which are rejected by OMM and IMM. To handle this problem caused by repetitive structures, W. Zhang and J. Kosecka [13] propose an additional criterion which increases true positive rate while keeps false positive low. For the right image pair in Fig. 5, MIFT and SIFT can not obtain reasonable matches since the image pair are changed too much with respect to the scale and occlusion. Some search results are shown in Fig. 7, the images from the database with top 3 scores are shown for every query image.

[2] http://lear.inrialpes.fr/∼jegou/data.php

Table 1. The results and their ranks of image search using MIFT and SIFT

#Image pairs	MIFT	$SIFT_1$	$SIFT_2$
12(NM)	TP, 11, 1^{st}; FP, 1	TP, 11,1^{st}; FP, 1	FP, 12
12(HM)	TP, 12, 1^{st}	FP, 12	TP, 12, 1^{st}
12(VM)	TP, 11, 1^{st} , 1, 2^{nd}	FP, 12	TP, 11, 1^{st}, 1, 3^{rd}

Fig. 5. Left image pair: the image pair with vertical reflection is queried at the second rank using MIFT, and at the third rank using SIFT. **Right image pair:** the image pair failed to be matched by both MIFT and SIFT.

4.2 Application to Detection of Symmetry Axis

In the real life, symmetric objects are ubiquitous, especially in urban area. Symmetry detection has been regarded as a useful and important technique in computer vision. Many symmetry detection schemes are proposed including human gait detection [14] and a focused version of Generalized Symmetry Transform [15]. As is known, the rules for planar symmetric objects are simple, symmetric points in the symmetric object lie on a single line perpendicular to the symmetry axis and their perpendicular distances to the symmetry axis are identical. However, the symmetric object under perspective projection is an exception of these rules. Although the ratios of lengths are not preserved under perspective projection, the following cross ratio constraint [16] is still valid,

$$\{\mathbf{v_x}, \mathbf{l_i}; \mathbf{c_i}, \mathbf{r_i}\} = \frac{(\mathbf{v_x} - \mathbf{l_i})(\mathbf{c_i} - \mathbf{r_i})}{(\mathbf{v_x} - \mathbf{r_i})(\mathbf{l_i} - \mathbf{c_i})} = 1 \tag{3}$$

where $\mathbf{v_x}$ is the vanishing point calculated by the parallels linking matched pixels, $\mathbf{l_i}$ and $\mathbf{r_i}$ are the endpoints of the line and $\mathbf{c_i}$ is the point on the symmetry axis.

Fig. 6. Example results of symmetry axis detection. The symmetry axes are marked with green straight line. Please zoom in the picture to see MIFT matches.

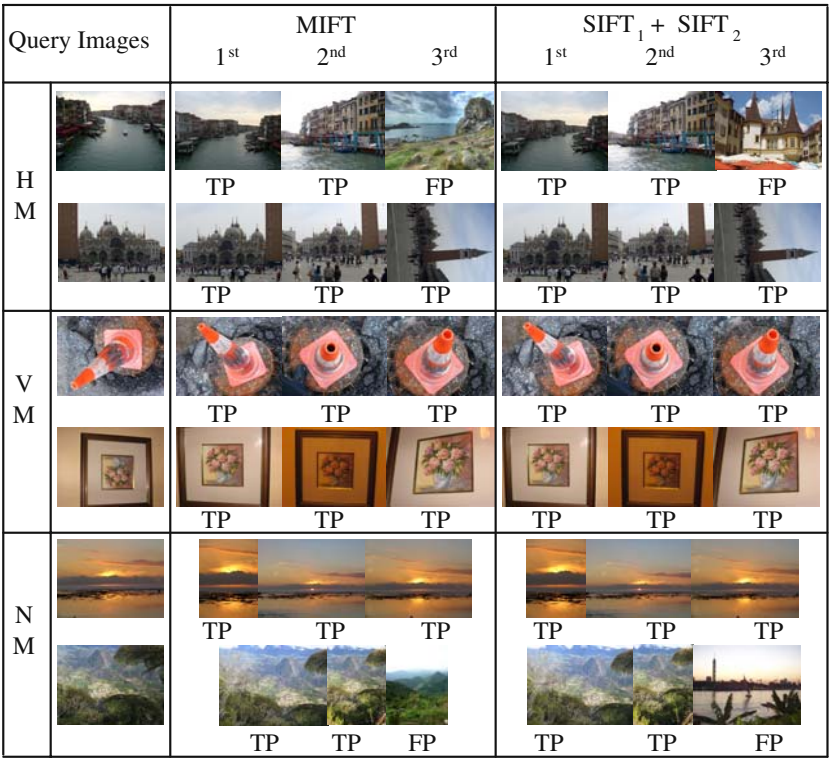


Fig. 7. Example results of image search. SIFT results are obtained by implementing query procedure twice, while MIFT only once. **Top two rows:** The query images with horizontal mirror reflection (HM). **Middle two rows:** The query images with vertical mirror reflection (VM). **Bottom two rows:** The query images without mirror reflection (NM). The results are labeled as true positive (TP) or false positive (FP) blow the searched images.

We use MIFT to detect the endpoints $\mathbf{l_i}$ and $\mathbf{r_i}$. $\mathbf{v_x}$ is then estimated from concurrent lines that link $\mathbf{l_i}$ and $\mathbf{r_i}$. Finally we use RANSAC [17] to find the optimal symmetry axis. Some results are shown in Fig. 6. Light colorful lines link matched pixel pairs, outliers of matched symmetric points have been eliminated by RANSAC.

5 Conclusion

In this paper, we have proposed a mirror reflection invariant descriptor, MIFT, and have verified its improved performance over the state-of-the-art SIFT in mirror reflection situations. MIFT performs comparably in general cases when compared with SIFT. We elaborate the new strategy for organizing descriptor, and describe the matching method used to reduce recall loss. MIFT instead of SIFT applied to the image search can capably handle the mirror reflection

images rather than implementing SIFT twice for every query image. And the symmetry axis of a planar symmetric object can be easily detected using MIFT with RANSAC.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 50735003), Tianjin University 985 research fund, and State Key Laboratory of Precision Measuring Technology and Instruments open fund.

References

1. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: ICCV, vol. 1, pp. 525–531 (2001)
2. Kleban, J., Xie, X., Ma, W.: Spatial pyramid mining for logo detection in natural scenes. In: ICME, pp. 1077–1080 (2008)
3. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
4. Brown, M., Lowe, D.: Recognising panoramas. In: ICCV (2003)
5. Harris, C., Stephens, M.J.: A combined corner and edge detector. In: Alvey Vision Conference, vol. 20, pp. 147–152 (1988)
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. In: IJCV, vol. 60, pp. 91–110 (2004)
7. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
8. Mikolajczyk, K., Schmid, C.: A performance evalution of local descriptors. PAMI 27, 1651–1630 (2004)
9. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: CVPR, pp. 1–8 (2008)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
11. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: ACM International conference on Multimedia, pp. 357–360 (2007)
12. Ke, Y., Suktnankar, R.: Pca-sift: A more distictive representation for local image descriptors. In: CVPR, vol. 2, pp. 506–513 (2004)
13. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT, pp. 33–40 (2006)
14. Hayfron-Acquah, J.B., Nixon, M.S., Carter, J.N.: Automatic gait recognition by symmetry analysis. In: Pattern Recognition Letters, vol. 24, pp. 2175–2183 (2003)
15. Choi, I., Chien, S.I.: A generlized symmetry transfor with selective attention capability for specific corner angels. IEEE Signal Processing Letters 11, 255–257 (2004)
16. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
17. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. Assoc. Comp. Mach. 24, 381–395 (1981)

Detection of Vehicle Manufacture Logos Using Contextual Information

Wenting Lu, Honggang Zhang, Kunyan Lan, and Jun Guo

Pattern Recognition and Intelligent System Lab,
Beijing University of Posts and Telecommunications, Beijing, China, 100876
Luwenting1983@163.com

Abstract. Besides the decorative purposes, vehicle manufacture logos can provide rich information for vehicle verification and classification in many applications such as security and information retrieval. Detection and recognition of vehicle manufacture logos are, however, very challenging because they might lack of discriminative features themselves. In this paper, we propose a method to detect vehicle manufacture logos using contextual information, i.e., the information of surrounding objects near vehicle manufacture logos such as license plates, headlights, and grilles. The experimental results demonstrate that the proposed method is more effective and robust than other methods.

1 Introduction

The vehicle manufacture logos can contain important information about vehicles besides their decorative purposes. A vehicle logo can provide not only the identity for the manufacture, but also the information for vehicle verification, identification, and recognition. For example, a manufacture logo of a certain type vehicle appears in a fixed location. It could indicate a problem if the logo appears at a wrong location. Association of a manufacture logo with a license plate can be used for vehicle verification. Manufacture logos can also be used for index to search a vehicle, e.g., we can search for "a red Toyota sedan" or "a black Ford SUV". Both "Toyota" and "Ford" can be identified from their manufacture logos. Therefore, detection and recognition of vehicle manufacture logos have many potential applications. However, unlike the license plate, which is designed for identification purposes, vehicle manufacture logos are mainly designed for decorative purposes such that they can be embedded into surrounding objects or have low contrast against background, as shown in Fig.1. From a detection and recognition point of view, the differences between a license plate and a manufacture logo are as follows:

1. While all license plates are installed in the specified locations (e.g., in the central part of the front and/or back of a vehicle), locations of vehicle logos can differ from manufactures. For example, as shown in Fig.1(a), the Mercedes Benz logo is almost separated from the main body of the car with the windshield glass as its background, which creates a great challenge for

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 546–555, 2010.

detection and recognition. The vehicle logo in Fig.1(b) is surrounded by the grille while the vehicle logo in Fig.1(c) is embedded into body.
2. While the shape, text, and color of license plates are standardized by the authority, a manufacture has freedom in designing appearance of a logo which makes a great variation in appearances of different logos. Some logos are even very similar, e.g., a Honda logo vs. a Hyundai logo, as shown in Fig.1(d) and Fig.1(e).
3. While it is illegal to mask or modify a license plate, the regulations on manufacture logos are much loser. A vehicle logo can be occluded by other objects, as shown in Fig.1(f).
4. While the content of a license plate consists of letters and digits within a rectangular plane, manufacture logo can contain letters, digits and any symbols on complex surfaces.

(a) Separated from body (b) Surrounded by grille (c) Embedded into body

(d) Honda logo (e) Hyundai logo (f) Partial occlusion

Fig. 1. An illustration of some challenging cases for manufacture logo detection and recognition

Therefore, it is very difficult, if not impossible, to robustly detect vehicle logos in real world applications. Fortunately, there is some contextual information, i.e., the information of surrounding objects near vehicle manufacture logos such as license plate, headlights, and grille, that can help to detect vehicle manufacture logos. In this paper, we propose a new method to localize vehicle manufacture logos using contextual information. The basic idea is to make full use of the spatial relationship among license plates, headlights, grilles and the vehicle manufacture logo. Firstly, we use an Adaboost classifier to detect license plate and use SIFT (Scale Invariant Feature Transform) algorithm to localize headlights. Secondly, we analyze the relationship among license plate, headlights and the vehicle manufacture logo by statistical laws to obtain the coarse position of the manufacture logo. Finally, we use ALBP (Advanced Local Binary Pattern) operator to detect whether exists grille in the coarse region and if it exists, we eliminate the disturbance to get the accurate position of the logo. The experimental results show that this method is more effective and robust than other methods.

The rest of this paper is organized as follows. Section 2 presents some related work. Section 3 illustrates how to use the contextual information among license

plate, headlights, and vehicle manufacture logos to detect the vehicle manufacture logos. Section 4 discusses experimental results. Finally we conclude the paper in section 5.

2 Related Work

Vehicle manufacture logo detection and recognition is a relative new research area, which has little previous existing work. Furthermore, since manufacture logos might lack of discriminative features themselves, it is very difficult, if not impossible, to use state-of-art algorithm directly for vehicle logo detection and recognition. However, the proposed approach is related some previous work in literature. As one of the most important studies on computer vision and pattern recognition in the application field of Intelligent Transportation Systems (ITS), License Plate Recognition (LPR) is now a mature technology, and has been applied in many applications such as intersections monitoring, parking lots management, over-speed detection on freeway ,vehicle verification [1], [2], [3].

Contextual information has been widely used in solving different AI (Artificial Intelligence) problems. It is well known that context plays a very important role in representing knowledge. In general, researchers consider the COIN program [4], [5] of MIT as the representative for the application of context in the field of information integration. Combining the existing literatures with our research emphases, we give the definition of context as follows: context—a set of hypothesis, from which we can make secret semantic information specific. According to this theory, we consider using the other information except manufacture logos themselves in the vehicle frontal area to localize manufacture logo, i.e., the information of surrounding objects near vehicle manufacture logos, such as spatial relationship among plate, headlights, grille, and the logo. And then vehicle manufacture logo recognition will become a 2D shape recognition problem [6], [7]. In this way, we decompose manufacture logo detection that is a challenging problem into several easier problems such as license plate detection, headlight detection, and grille detection.

Currently available methods for license plate detection can be categorized based on input source (gray or color images) and type of classifiers. There are some successful methods based on the gray-scale image. In [8], Mahini et al. introduced a feature-based license plate localization algorithm that coped with the multi-object detection problem in different image capturing conditions. In [9], Le et al. presented a hybrid method for extracting license plates from cluttered images based on an edge density map. Although in some cases, the methods based on the gray-scale image can get the region of license plate. However, when the license plate has regional distortion, or when the image is defaced, efficiency decreases significantly, accompanied by slow speed, a high miss rate and false alarm rate. Besides the methods that based on gray-scale image, color information is also exploited, such as [10], [11]. Color-based detection methods can obtain more accurate location, lower miss rate, but poor adaptive capability to

the size of license plate. It can be impacted by abrasion and tilt. In addition, the license plate can be confused by the regions having similar color, shadows and illumination changes in environment that has a dramatic effect on the definition of the color space. Thus, in order to localize manufacture logo accurately, we need to adopt a method that not only provides fast and precise position of license plate but also has a strong robustness. In order to achieve this goal, we adopt the detection method based on Adaboost classifier [12], [13] in this paper, which will be described in 3.1. Furthermore, we use SIFT features to detect headlights in this research.

3 Vehicle Logo Detection Based on Contextual Information

Vehicle manufacture logo localization is prerequisite for logo recognition, whose accuracy is crucial to follow-up processes and has a direct impact to the speed and accuracy of logo identification. In this paper, We focus on vehicle manufacture logo localization only. Fig.2 shows the flow chart of the processing.

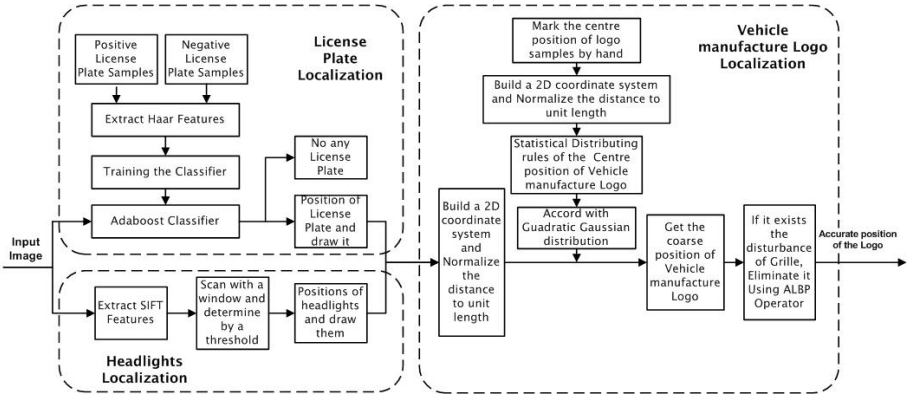


Fig. 2. The flow chart of vehicle logo localization based on contextual information

3.1 License Plate Localization

Based on the previous analysis, we adopt the detection method based on the Adaboost classifier. This target detection method was first introduced by Viola and Jones in 2001 [12] and is improved by Lienhart [14], [15]. They use an image representation method known as the "integral image", which can quickly calculate the features used by the detector. We use the Haar features of samples to train classifier, resulting in a cascade of the boosted classifier. For the classifier training, we use 3000 positive sample images and 10000 negative sample images. The classifier system we choose is similar to the Discrete Adaboost classifier in the form. Given a set of N weak classifiers, a strong classifier is computed as a

linear combination of a series of weighted weak classifiers [13], as is shown in Equation (1) and (2).

$$H^{strong}(x) = sign(conf(x)), \tag{1}$$

$$conf(x) = \sum_{n=1}^{N} \partial_n \cdot h_n^{weak}(x), \tag{2}$$

where, the value $conf(\cdot)$ (which is related to the margin) can be interpreted as a confidence measure. Moreover, each weak classifier is created by using only one Haar feature. The results of the license plate localization are shown in Fig.3.

Fig. 3. An illustration of the license plate localization based on the Adaboost classifier

3.2 Headlights Localization

In recent years, the method based on Local Invariant Descriptor makes great progress on object recognition and image matching in computer vision. In 1999, Lowe presented a new algorithm called SIFT to extract feature points [16], and improved in 2004[17], which made good performance in solving image distortion problem caused by sheltering, scale, rotation, zooming or the focus changing, and successfully applied in object recognition, image restoration and image mosaic.

Vehicle headlights are usually in complex light-reflecting multi-faceted form. Thus, regardless of the lights are in the state of operation (at night) or clearance (daytime), the headlights areas are more complex and frequent changing areas, with very rich in all orientations of the gradient information. Therefore, we can use the SIFT features to perform the headlights localization. These features extracted from headlights are local and robust to rotation, scale, changes in illumination and addition of noise. Moreover, they are invariant to full Affine Transformation and viewing angle change. Space lacks for a detailed description of it, the detail steps of SIFT algorithm please refer to reference [17]. Fig.4 shows examples of the detected features of frontal vehicles using SIFT algorithm. Each keypoint has three coordinates: location (x, y), scale and orientation.

As shown in Fig.4, the feature points detected in the area of headlights using SIFT algorithm are very dense and relatively concentrated. Thus we can set a detection region (including headlights and vehicle logo), in which we use a rectangular window and set a reasonable threshold to search, and then confirm whether this area is headlights or not according to the density of the feature points in this detection region. Therefore, we can localize headlights accurately. The density threshold of the feature points can be obtained by the experiment analysis. The results of headlights localization are shown in Fig.5.

Fig. 4. Examples of SIFT features in the front of vehicles

Fig. 5. An illustration of the results of headlight localization using SIFT algorithm

3.3 Contextual Information from a License Plate and Headlights

According to 3.1 and 3.2, we have obtained the exact position of license plate and headlights. In order to make better use of contextual information like license plate and headlights to localize manufacture logos, we use 200 images of vehicles as samples, and obtain the center position of manufacture logos using the statistical rules. At first, it needs to set up a two-dimensional coordinate system in the vehicle head, and take the center of license plate as the origin of the coordinate system, marking it as $(0, 0)$. According to the pixel coordinates of the license plate and headlights in original image, we can calculate the center coordinates of the headlights in this two-dimensional coordinate system easily, marking the coordinates of left headlight as (x_1, y_1) and right as (x_2, y_2), shown in Fig. 6. Because of the different scales, we need to map the distances between the headlights and the origin both vertically and horizontally by normalizing the distances to unit length for all the sample images, and then calculate the coordinates (x, y) of the center position of vehicle manufacture logo using the statistical rules. The result reveals that the center position of logo in these 200 images focus on the some specific area showing as being dense in the center and sparse in the margin, which can be approximately described by two-dimensional Gaussian distribution.

By statistical laws, such center positions can be expressed by the mean value of two-dimensional Gaussian distribution as Equation (3):

$$\overline{z} = \frac{1}{n}\sum z = \frac{1}{n}\sum(\overrightarrow{x}, \overrightarrow{y})^T, \tag{3}$$

where, z represents the two-dimensional random variable of the center point of vehicle manufacture logos, $\overrightarrow{x}$ and $\overrightarrow{y}$ respectively represents the vertical or horizontal coordinates of the center point of the logo.

The covariance matrix of two-dimensional Gaussian distribution is also of great importance to vehicle manufacture logo localization, which represents dispersion of the logo center, expressed as Equation (4):

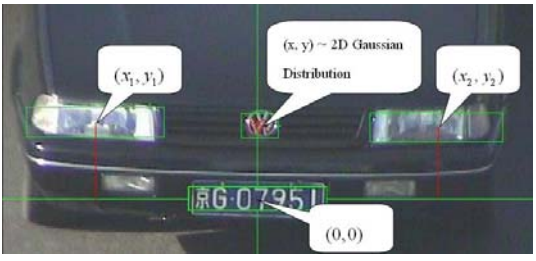


Fig. 6. A two-dimensional coordinate system in the vehicle front, the center position of vehicle manufacture logo can be approximately described by two-dimensional Gaussian distribution

$$C_z = \frac{1}{n}\sum(z - \overline{z})(z - \overline{z})^T, \tag{4}$$

When inputting a new image, it utilizes the Adaboost classifier and the SIFT algorithm to get the center coordinates of license plate and headlights first, mapping the distance by normalizing the distance to unit length, and then get the coordinates of manufacture logo center by Equation(3). Finally, it needs to turn the coordinates of the vehicle logo center in the two-dimensional coordinate system to the pixel coordinates of the logo in original image according to anti-mapping distance normalized unit. Then we get the center point of the logo. Finally, we draw a rectangle of reasonable size centered with this point and this is the coarse region manufacture logo exists.

3.4 Contextual Information Using Grille

However, it is possible that a grille may link with vehicle manufacture logos in that coarse region, which brings more difficulties to vehicle manufacture logo recognition. Considering that grilles of most cars are arrayed regularly in either horizontal or vertical direction (shown in Fig.7), we can use ALBP operator to detect whether exists grille in the coarse region and if it exists, we eliminate the disturbance of grille, so as to get the exact position of vehicle manufacture logos.

The original LBP operator introduced by Timo Ojala proved to be a powerful texture descriptor [18]. However, it demands a little more time and storage space. Thus, in this paper, we adopt a more powerful and low-computation ALBP

Fig. 7. An illustration of well-regulated array of grille

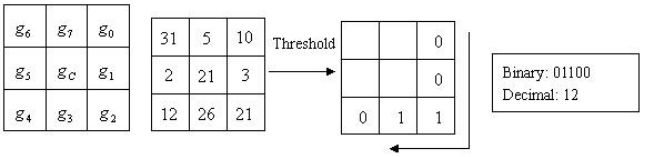


Fig. 8. An example of ALBP operator

Fig. 9. An illustration of the final results of vehicle manufactures logo detection

operator [19] to detect and eliminate the disturbance of grille. As shown in Fig.8, the operator labels the pixels of an image with the binary thresholding result of the 3×3 neighborhood of each pixel with the center value. The ALBP operator only uses the right and bottom pixels of the original LBP operator and then the histograms of the labels can be used as texture descriptor. The express of the ALBP operator is as Equation (5).

$$ALBP = \sum_{p=0}^{4} s(\mathrm{g}_p - \mathrm{g}_c)2^p, \qquad \text{where} \quad s(z) = \begin{cases} 1, z \geq 0 \\ 0, z < 0 \end{cases} \tag{5}$$

After the image input, we scan the coarse region of the logo with a 3×3 sub-window to get the ALBP map, then divide the map into 6×6 non-overlapping sub-regions, and compute the histogram for each block. If the histogram is symmetrical, it means that the distribution of the pixels in the block is even, and it can be identified as grille. For it has obvious distribution features as grille and can be disregarded. Take the horizontal grille as example. It should be searched from the left and right rim to the center until the histogram is not symmetrical. When it stops, the reserved area in the middle is just the final detection result of vehicle manufacture logo, as is shown in Fig.9. The position of the vertical grille is similar.

4 Experimental Results

In order to evaluate our method, we use an image database with 1000 color vehicle head images in various real life applications, and the size of each image is 720×288. In our experiments, we consider the correct detection as localizing the complete vehicle logo in a region without any other disturbance of grille. Fig.10 shows a comparison on vehicle manufacture logo correct detection rate of 5 different methods: (1) localizing manufacture logos directly in whole images without any contextual information, mark N; (2) only using the information of license plate to localize the accurate region of logos, for short P; (3) method only using the information of headlights, for short L; (4) method only using

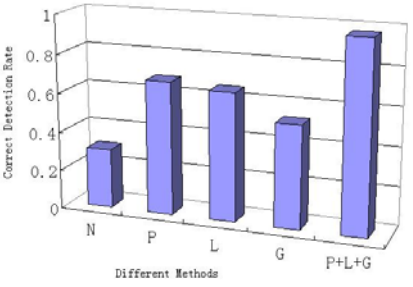


Fig. 10. Comparison on vehicle manufacture logo correct detection rate of 5 different methods

the information of grille, for short G; (5) the proposed method using contextual information in this paper, for short P+L+G.

As shown in Fig.10, the correct detection rate of vehicle manufacture logo without any contextual information or only using one certain contextual information is too low to satisfy the requirement of application. While the method based on contextual information we proposed in this paper performs better and more effective.

5 Conclusions

The main contribution of this paper is to have successfully proposed a new method to detect vehicle manufacture logos accurately. This method is based on contextual information, i.e., the information of surrounding objects near vehicle manufacture logos such as the spatial relationship among license plate, headlights and the manufacture logo, and well-regulated array of grille. The experimental results indicate that the proposed method not only can localize vehicle logo accurately, but also is robust against environmental changes. Therefore, it can provide a solid foundation for manufacture logo recognition and can provide rich information for vehicle verification and classification in many applications such as security and information retrieval.

Acknowledgements

This research was partially supported by 863 High-project under Grant No. 2007AA01Z417 and 111 Project under Grant No. B08004.

References

1. For, W.-K., Leman, K., Eng, H.-L., Chew, B.-F., Wan, K.-W.: A multi-camera collaboration framework for real-time vehicle detection and license plate recognition on highways. In: IEEE Transactions on Intelligent Vehicles Symposium, June 2008, pp. 192–197 (2008)

2. Ge, G., Bao, X., Ge, J.: Study on automatic detection and recognition algorithms for vehicles and license plates using ls-svm. In: The 7th World Congress on Intelligent Control and Automation, June 2008, pp. 3760–3765 (2008)
3. Arth, C., Limberger, F., Bischof, H.: Real-time license plate recognition on an embedded dsp-platform. In: CVPR, June 2007, pp. 1–8 (2007)
4. Firat, A.: Information integration using contextual knowledge and ontology merging. MIT Sloan School of Management, Cambridge (2003)
5. Hongwei, Z., Madnick, S.E.: Context interchange as a scalable solution to interoperating amongst heterogeneous dynamic services. In: Proc. of the 3rd Workshop on eBusiness (2004)
6. Doerman, D.: Applying algebraic and differential invariants for logo recognition. Machine Vision Apply 9(2), 73–86 (1996)
7. Dlagnekov, L.: Video-based car surveillance: License plate make and model recognition, masters thesis. University of California at San Diego (2005)
8. Mahini, H., Kasaei, S., Dorri, F.: An efficient features - based license plate localization method. In: ICPR, vol. 2, pp. 841–844 (2006)
9. Le, W., Li, S.: A hybrid license plate extraction method for complex scenes. In: ICPR, vol. 2, pp. 324–327 (2006)
10. Wei, W., Wang, M., Huang, Z.: An automatic method of location for number-plate using color features. In: ICIP, vol. 1, pp. 782–785 (2001)
11. Jia, W., Zhang, H., He, X., Piccardi, M.: Mean shift for accurate license plate localization. ITS, 566–571 (2005)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
13. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR, vol. 1, pp. 260–267 (2006)
14. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: ICIP, vol. 1, pp. 900–903 (2002)
15. Kuranov, A., Lienhart, R., Pisarevsky, V.: An empirical analysis of boosting algorithms for rapid objects with an extended set of haar-like features. Intel Technical Report MRL-TR-July02-01 (2002)
16. Lowe, D.G.: Object recognition from local scale - invariant features. In: ICCV, pp. 1150–1157 (1999)
17. Lowe, D.G.: Distinctive image features from scale - invariant interest points. IJCV, 91–110 (2004)
18. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI 24(7) (2002)
19. Wang, Y., Zhang, H., Fang, X., Guo, J.: Low-resolution chinese character recognition of vehicle license plate based on albp and gabor filters. In: ICAPR, pp. 302–305 (2009)

Part-Based Object Detection Using Cascades of Boosted Classifiers

Xiaozhen Xia[1], Wuyi Yang[2], Heping Li[1], and Shuwu Zhang[1]

[1] Hi-tech Innovation Center, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
{xiaxiaozhen,hpli,swzhang}@hitic.ia.ac.cn
[2] Key Laboratory of Underwater Acoustic Communication and Marine Information Technology, Ministry of Education, Xiamen University, Xiamen, China
wyyang@xmu.edu.cn

Abstract. We present a new method for object detection that integrates part-based model with cascades of boosted classifiers. The parts are labeled in a supervised manner. For each part, we construct a boosted cascade by selecting the most important features from a large set and combining more complex classifiers. The weak learners used in each level of the cascade are gradient features of variable-size blocks. Moreover, we learn a model of the spatial relations between those parts. In detection, the cascade of classifiers for each part compute the part values within all sliding windows and then the object is localized within the image by integrating the spatial relations model. The experimental results demonstrate that training a cascade of boosted classifiers for each part and adding spatial constraints among parts improve performance of detection and localization.

Keywords: object detection, part-based model, boosted cascade.

1 Introduction

The problem of object detection has drawn considerable attention in computer vision and artificial intelligence. The concentrated effort of researchers in the last few years resulted in many novel approaches for object detection. But it still remains a challenging problem because of the large variance of objects in the class. Such variance may be due to the view point variation, illumination and occlusion, change in the scale, background clutter, and deformable object shape, etc.

A variety of models and methods exist for representing, learning and recognizing objects in images. Recent papers have demonstrated that part-based models can achieve good performance in object detection. The pictorial structure models [9, 10] represented an object by a collection of parts arranged in a deformable configuration, where the spatial relationships between parts are represented by spring-like connections between pairs of parts. The constellation model [6, 7] represented objects as flexible constellations of rigid parts. The variability within a class is represented by a

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 556–565, 2010.

joint probability density function on the shape of the constellation and the output of part detectors. In star model [8] and k-fan model [11], the location of the model part is conditioned on the location of a reference part. While these models are appealing in their generality, it has been complex for learning and difficult to establish their value in practice.

Recent work has also shown that boosted object detectors can be effective at detecting object classes. Schneiderman and Kanade [12] train each classifier based on the statistics of localized parts to minimize classification error on the training set by using AdaBoost. Torralba et al. [5] present a multi-class boosting procedure by finding common features that can be shared across the classes. Viola and Jones [2] construct a boosted cascade of simple feature classifiers for rapid face detection. Zhu et al. [3] integrate the boosted cascade-of-rejectors approach with the HOG features to achieve a fast human detection system. The cascade of boosted classifiers has proven to be powerful in detection, however, the learning process increases the computational cost.

In this paper, we develop a new object detection framework that combines the part-based model and cascades of boosted classifiers. The basic idea is an extension of the boosted cascade algorithm [3], which has been shown to be useful for human detection. Different from previous works, we train a cascade of boosted classifiers for each part of the object and add spatial information for the models. The parts are labeled in a supervised manner. For each part, we extract HOG features [1] of the variable-size blocks, and each orientation bin of the HOG features corresponds to one feature in our scheme. Then we use AdaBoost to select a small number of important features viewed as weak learners and yield a strong classifier for each level of the cascade. Finally, a boosted cascade is constructed by combining the strong classifier in each stage. The spatial model for the part locations is similar to 1-fan [11] or star graph [8] with a reference part that is connected to all other parts. In addition, our part structure model are most closely related to [13], which use latent SVM to train deformable part model.

The rest of this paper is organized as follows. In section 2, we describe the model framework. In section 3, we provide a detailed description of learning method. Section 4 gives the procedures of detection and localization. The experimental results are presented in section 5. Finally, we conclude the paper in section 6.

2 Model

Our models are formed by integrating part structure with cascades of boosted classifiers. For each part, gradient features are extracted from all possible blocks that vary in sizes, locations and aspect ratio, then AdaBoost is used to select a small number of important features from a larger set and yield strong classifier for each stage. Finally, we combine the strong classifiers to construct a boosted cascade. The spatial relations between parts are tree-structured graph with a root part and other parts conditioned on the root part.

2.1 Integral Histogram of HOG and Gradient Features

The "Integral Histogram" [14] developed by Porikli allows for fast computation of histograms over arbitrary rectangular image regions. Inspired by his work, Zhu et al. [3] exploit a rapid way of calculating the HOG features. The gradient at each pixel is discretized into one of 9 orientation bins, and an integral image for each bin of the HOG is computed and stored. Then the HOG features for any rectangular regions can be computed quickly from those integral images. We follow the way to extract HOG features from any rectangular image regions.

A 48×48 fixed-size patch surrounding the labeled part location is extracted from each training image. To encode more information for each patch of part, we consider blocks whose size ranges from 8×8 to 48×48. The ratio between block width and block height can be any of the following ratios (1 : 1), (1 : 2) and (2 : 1). In total, we get 189 blocks in each patch of the part. The 9 dimensional HOG features of the blocks can be computed quickly from the integral images. Each feature in our scheme corresponds to one orientation bin of the HOG features of the blocks. Thus the total number of features in each patch of the part is 9×189.

2.2 Cascades of Boosted Classifiers

We use a cascade of boosted classifiers for each part. We choose to base our algorithm on the version of boosting called "gentleboost" [4] because it has been shown experimentally [15] to outperform other boosting algorithms. In gentleboost, the optimization of the cost function is done using adaptive Newton steps, which corresponds to minimizing a weighted squared error at each step.

In the boosting process, several critical features from a large set are selected to form weak learners. Then the strong classifier for each stage is constructed by those weak learners. For each 48×48 pixel region we have 189 blocks and 9×189 features. We select the most informative features from these features.

A boosted cascade is a combination of the strong classifier in each stage which reject many of the negative sub-windows at the earliest stage while detecting almost all positive instances. Stages in the cascade are constructed by training classifiers using gentleboost and then adjusting the threshold to meet the predefined quality requirements. Our cascade of boosted classifiers contains 4 stages for each part in that the first few stages of the cascade rejects the majority of detection windows.

2.3 Spatial Relations

We use the star-graph model proposed in [8] to model the spatial relations between parts. Let $G = (V, E)$ be a star graph with central node v_r and other independent nodes $v_i (i \neq r)$ conditioned on the value of v_r. Let $S = \{s_1, ..., s_n\}$ be parameters of spatial relationships. Let $L = \{l_1, ..., l_n\}$ denote the location for each part, among which l_r is the location of the central part and l_i is the location of other part except

for the central part. The spatial relations can be written in terms of conditional distributions as,

$$p(L \mid S) = p(l_r \mid s_r) \prod_{v_i \neq v_r} p(l_i \mid l_r, s_i). \tag{1}$$

We model the conditional distribution of other part location given the central part location $p(l_i \mid l_r, s_i)$ as a Gaussian with mean μ_{ilr} and covariance $\sum_{ilr}$,

$$p(l_i \mid l_r, s_i) = N(l_i - l_r, u_{ilr}, \sum_{ilr}). \tag{2}$$

3 Learning

In learning we are given a set of images annotated with bounding boxes for each part. In training the cascade, we need to construct a strong classifier for each level of the cascade to meet the predefined quality requirements. For spatial relations model, the parameters discussed above should be estimated.

3.1 Training the Cascade

We train a boosted cascade of classifiers similar to the one proposed in [2] and [3], with the following difference. First, we use 9 dimensional HOG feature to describe a block and each feature selected in each round of boosting process corresponds to one orientation bin of the HOG features. Second, for each part we select the most informative features from all the 1701 features in the 189 possible blocks and construct a rejection cascade. Finally, since our models are based on part structure, for each part we will train a rejection cascade and the levels of the cascade are less than that of [3] even it will result in low rejection rate.

The boosting process for training each stage is a standard Gentle AdaBoost algorithm. For each level of the cascade we construct a strong classifier consisting of several weak learners. Each stage is trained by adding weak learners until the target detection rate and false positive rate are met. Subsequent stage is trained using those samples which pass through all the previous stages. In our system we require the minimum detection rate to be 0.9975 and the maximum false positive to be 0.7 in each stage.

3.2 Training Spatial Relations

We learn the spatial relations model from labeled images using a maximum likelihood (ML) criterion. The goal is to find the ML estimate S^* which best explain the data from all the training images. We are given a set of images $\{I_1, ..., I_m\}$ and corresponding object configurations $\{L_1, ..., L_m\}$ for each image, $L_k = \{l_{k,1}, ..., l_{k,n}\}$, then the ML estimate of S is,

$$S^* = \arg\max_S \prod_{k=1}^{m} p(L_k \mid S) = \arg\max_s \prod_{k=1}^{m} p(l_{k,r} \mid s_r) \prod_{v_i \neq v_r} p(l_{k,i} \mid l_{k,r}, s_i). \quad (3)$$

For a fixed set of central part, estimating the ML parameters involves estimating the mean and covariance in (2). These can be obtained from the sample mean and covariance of the labeled configurations.

4 Detection and Localization

In detection, the boosted cascade detector for each part scan across the image at multiple scales and compute the boosting score within all sliding windows, and then localize the object within the image by integrating the spatial relations model. Let $L = \{l_1, ..., l_n\}$ denote the location for each part, $c_i(l_i)$ denote the score of the presence of the i^{th} part in the candidate windows given the location l_i . $c_i(l_i)$ can be derived from boosted cascade detector. For getting the location of the object within each image, we look for an object configuration L^* with maximum posterior probability,

$$L^* = \max_L p(l_r \mid s_r) c_r(l_r) \prod_{v_i \neq v_r} p(l_i \mid l_r, s_i) c_i(l_i). \quad (4)$$

There is a large number of placements for the parts of the object. We use distance transforms technique [10] to compute the best location for the parts of the object as a function of the central part location. We score central part locations according to the best possible placement of the parts and threshold this score. In this case the running time of the localization algorithm is reduced to $O(nk)$, where n is the number of parts and k is the number of detection windows (number of locations) for each part within the image.

5 Experiments

We carried experiments to investigate how integrating part structure with cascades of boosted classifiers affects object detection and localization accuracy. In our first set of experiments, we applied our learning method to the INRIA person dataset [1], which consists of 2,416 positive training images and 1,126 negative training images. Positive training images were scaled so that object size was approximately uniform across the set of images. For human six parts were labeled by hand, which were the head, left and right arm, middle part of the body, left and right leg.

To learn the boosted cascade of classifiers for a given part, a 48×48 fixed-size patch surrounding the labeled part location was extracted from each training image, and then HOG features of variable-size blocks in the patch were computed. The

cascade of boosted classifiers was trained using the procedure described in Section 3.1. We trained only four stages for each part, which enabled the detection rate to be 0.99 and the rejection rate to be 0.76. The training process took 5 to 6 hours using a PC with 2.66GHz CPU and 1GB memory. The number of features in the four layers of the detector was 5, 13, 17, 25 features respectively. In total, about 240 weak learners were selected for all the parts.

We inspected the most informative blocks among which features are selected by our system for each part. The first column of figure 1 show the best 6 blocks among which features are selected with minimum error rates in each part. We observe that the best block size for each part are about 32×32, 16×16, 48×48, 32×32, 24×24, 32×16 respectively. The last two columns show the blocks selected in our cascade of level-1,2 as well. Another observation is that most selected blocks cluster in the central area of each part, which demonstrates that gentleboost is very efficient in selecting the most informative blocks as opposed to the blocks in background.

For detection, we found an optimal configuration for the object in each test image. Distance transforms technique was used to compute the best location for the parts of the object. Figure 2 shows some detection and localization results of our algorithm for the person dataset. We compare our method to the way of using boosted cascade without part information, with the following details. Each stage is trained in gentleboost manner, the weak learner selected in the boosting process corresponds to one orientation bin of the HOG features, and the cascade has 20 stages and about 435 weak learners. We also implemented the Dalal & Triggs [1] algorithm using the same training and testing databases they provided.

Figure 4(a) shows the resulting precision-recall curves for the person dataset. From the curves we can see that our method based on part structure and 4 levels of cascade for each part performs better than the way based on 20 levels of cascade without parts and the Dalal & Triggs method. The result demonstrate that training a cascade of boosted classifiers with less stages for each part and adding spatial information into the framework are helpful in improving detection accuracy.

In the next experiment we evaluated our system on the Caltech-101 database [16]. Five categories were selected from the database, which were motorbikes, airplanes, faces, leopards, and cars. Each dataset was split randomly into two separate sets. We used the first for training and the remaining for testing. Six parts were labeled by hand in each training images. The patch size of each part is 48×48 for all classes but the leopard for which the patch size is 36×36. Figure 3 illustrates some detection and localization results on the motorbike, airplane, face, leopard and car dataset, showing precise localization of the parts despite substantial variability in their appearances and locations.

We trained a boosted cascade without part structure for each class and compared our method to it. Figure 4(b-f) show the resulting precision-recall curves on motorbike, airplane, face, leopard and car dataset respectively. Table 1 presents the average precision scores for each experiment on the INRIA person dataset and five categories of Caltech-101 datasets. The results show that our model provided a substantial improvement in accuracy for the person, motorbike and airplane dataset,

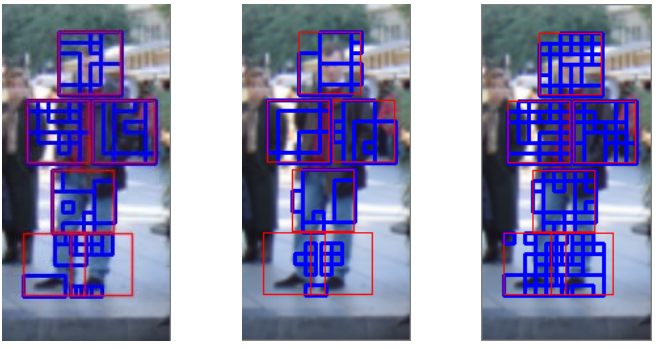


Fig. 1. Visualizing the selected blocks from training set for each part. The first column show top 6 blocks among which features are selected with minimum error rates in each part. The second and third column show the blocks selected in the cascade of level-1 and level-2 respectively.

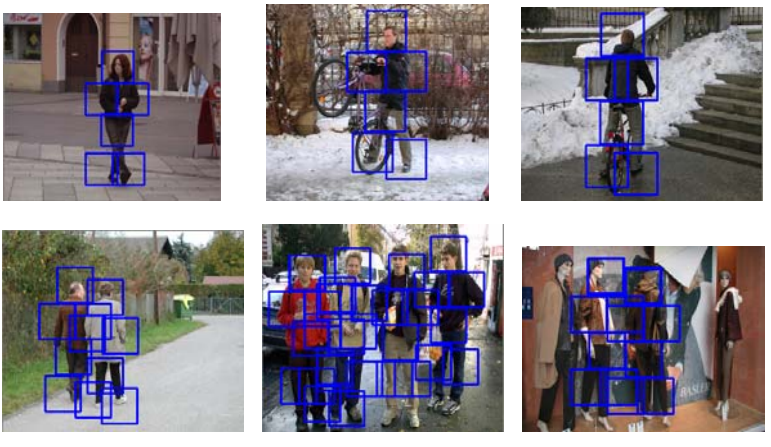


Fig. 2. Some results from INRIA person dataset. The first five show correct detections while the last shows false positive.

little improvement for the face and leopard dataset, no improvement for the car dataset. For the person, motorbike and airplane dataset, each part has rich information and adding spatial constraints among parts for these classes is helpful for detection and localization. For the face and leopard dataset, there are little improvement for the detection accuracy, the reason for which is that hand-labeling the training images may add noise for the part information, and wide variation for the leopard dataset affects the spatial relationships between parts. The lack of improvement for car may be due to the problem of overfitting. Furthermore, a supervised approach is limited by the quality of the parts chosen and the accuracy of the hand-labeled ground truth.

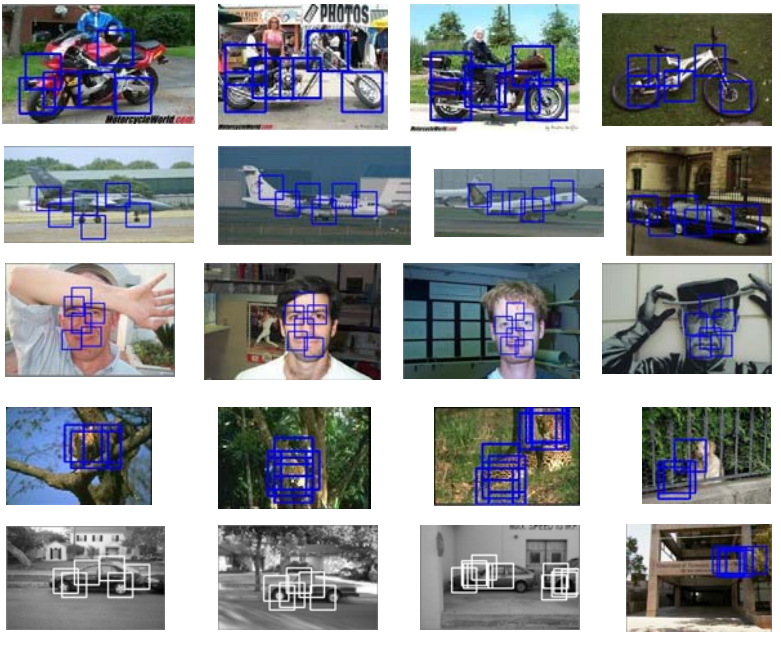


Fig. 3. Some results from Caltech-101 database. Each row shows detections using a model for a specific class (Motorbike, Airplane, Face, Leopard, Car). The first three columns show correct detections while the last column shows false positives.

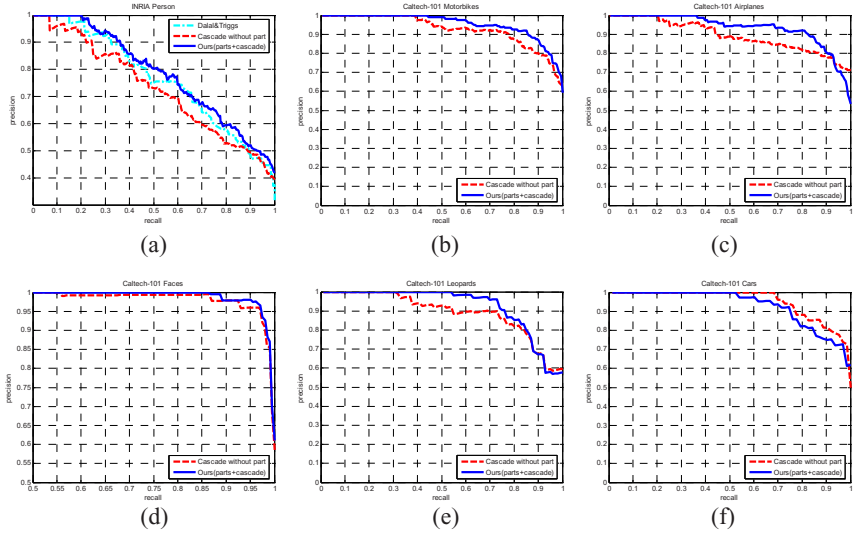


Fig. 4. Precision-recall curves for the INRIA person dataset and Caltech-101 motorbikes, airplanes, faces, leopards, and cars dataset

Table 1. Average precision scores for each experiment on the 6 categories of INRIA person dataset and Caltech-101 dataset

	Person	Bikes	Airplanes	Faces	Leopards	Cars
Cascade without part	0.7295	0.9162	0.8925	0.9588	0.8981	**0.9233**
Ours (parts+cascade)	**0.7787**	**0.9357**	**0.9151**	**0.9627**	**0.9137**	0.9172
Dalal & Triggs [1]	0.7549					

6 Conclusion

We have proposed a new method for object detection and localization in images. Our approach relies on a learning framework that combines the part structure and cascades of boosted classifiers. We investigate whether training a cascade of boosted classifiers for each part and adding spatial constraints among parts are actually helpful in detection and localization. Experimental results on a variety of categories demonstrate the power of our system in object detection and localization.

Currently, we use a supervised learning process which is limited by the quality of the parts chosen and the accuracy of the hand-labeled ground truth. Moreover, the framework do not use the context analysis, where the presence of a certain object class in an image probabilistically influences the presence of a second class. We are adapting our implementation to weakly supervised procedure and working on integrating scene context to improve detection.

Acknowledgments. The research was supported by National Sciences & Technology Support Program of China (Grant No. 2008BAH21B03, Grant No. 2008BAH26B02, and Grant No. 2008BAH26B03).

References

1. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, pp. 886–893 (2005)
2. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR, pp. 511–518 (2001)
3. Zhu, Q., Avidan, S., Yeh, M.C., Cheng, K.T.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: CVPR, pp. 1491–1498 (2006)
4. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Annals of Statistics, 337–374 (2000)
5. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing Visual Features for Multiclass and Multiview Object Detection. PAMI, 854–869 (2007)
6. Weber, M., Welling, M., Perona, P.: Unsupervised Learning of Models for Recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
7. Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: CVPR, pp. 264–271 (2003)

8. Fergus, R., Perona, P., Zisserman, A.: A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In: CVPR, pp. 380–387 (2005)
9. Fischler, M.A., Elschlager, R.A.: The Representation and Matching of Pictorial Structures. IEEE Transactions on Computer, 67–92 (1973)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial Structures for Object Recognition. IJCV, 55–79 (2005)
11. Crandall, D.J., Felzenszwalb, P.F., Huttenlocher, D.P.: Spatial Priors for Part-Based Recognition using Statistical Models. In: CVPR, pp. 10–17 (2005)
12. Schneiderman, H., Kanade, T.: Object Detection Using the Statistics of Parts. IJCV, 151–177 (2004)
13. Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: CVPR (2008)
14. Porikli, F.: Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. In: CVPR (2005)
15. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In: Proc. DAGM 25th Pattern Recognition Symp. (2003)
16. 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: CVPR Workshop on Generative-Model Based Vision (2004)

A Novel Self-created Tree Structure Based Multi-view Face Detection

Xu Yang, Xin Yang, and Huilin Xiong

Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200240, China
feilang@foxmail.com, {yangxin,hlxiong}@sjtu.edu.cn

Abstract. This paper proposes a self-created multi-layer cascaded architecture for multi-view face detection. Instead of using predefined *a priori* about face views, the system automatically divides the face sample space using the kernel-based branching competitive learning (KBCL) network at different discriminative resolutions. To improve the detection efficiency, a coarse-to-fine search mechanism is involved in the procedure, where the boosted mirror pair of points (MPP) classifiers is employed to classify image blocks at different discriminatory levels. The boosted MPP classifiers can approximate the performance of the standard support vector machines in a hierarchical way, which allows background blocks to be excluded quickly by simple classifiers and the 'face like' parts remained to be judged by more complicate classifiers. Experimental results show that our system provides a high detection rate with a particularly low level of false positives.

Keywords: Face detection; Support Vector Machines; Competitive learning network; Machine learning.

1 Introduction

Face detection plays important roles in wide range of practical applications. In recent years, the frontal face detection has made remarkable progress with the use of the cascade classifiers [1], and the multi-view face detection has attracted a lot of research attentions, since more than 75% faces in real images are non-frontal [2]. Compared with frontal face detection, detecting multi-view faces is a more challenging task, because profile and half profile faces tend to be less informative, more diverse, and more sensitive to noise. Moreover, the ubiquitous concomitance of pitch and yaw of faces further compounds the diversities of the training face. Some powerful detector structures are proposed in the literature to cope with these more diverse patterns with large intra-class variation, including Wu et al.'s parallel cascade [3], Fleuret et al.'s scalar tree [4], Li et al.'s pyramid [2], Jones et al.'s decision tree [5], and Huang et al.'s Width-First-Search tree [6]. The strategies taken by these methods are the same. Before the training process to construct the multi-view face detector, the face samples with similar view information are first grouped together by hand. Therefore, the whole face sample space is partitioned into subspaces manually, and then the view dependent specific face classifiers are trained on each of them. The final detection

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 566–575, 2010.

result is obtained by merging the parts passing all the view dependent classifiers. However, despite the notable contributions of the schemes outlined above, these view-partition approaches need *a prior* and experience to construct their tree architecture, and assign the sub-category labels manually for each face image in the training set. This could be time consuming, and furthermore, difficult to generalize from one application to another.

In this paper, we proposed a method that can automatically construct tree-structured classifiers. Instead of using predefined information for view partition based on people's knowledge, we divide the multi-view face samples by the kernel-based branching competitive learning (KBCL) method at different discriminative resolution levels. We call this structured detector the KBCL-based multi-resolution tree detector. The SVM is adopted to classify the face and non-face patterns for its robustness to noise and distortions affecting the face patterns. The clustering feature of our KBCL-based multi-resolution tree makes the learning task for each SVM more specific and simpler. In order to achieve high detection speed further, we simplified the SVM by a series of boosted mirror pair of points (MPP) classifiers of different accuracies, which was taken as the member classifier for each tree node at different levels. The boosted MPP classifiers with different complicate hierarchies allows the background patches to be filtered away quickly by the top few layers and the 'face like' parts remained to be judged by more complicate classifiers at later layers.

The paper is organized as follows: Section 2 firstly introduces the BCL method, and then explains why we run BCL on the kernel induced empirical feature space to cluster the multi-view face samples in a multi-resolution manner. In Section 3 the SVM and the boosted MPP classifier are formulated. Experimental results are given in Section 4 and the conclusions are in the last Section.

2 Kernel-Based Branching Competitive Learning and Multi-resolution Clustering Tree

BCL [7] is a self-creating clustering model of a neural network. We take it as the measure to divide the multi-view face samples automatically for its efficiency in capturing the statistical distributions of the data under different resolutions.

2.1 Branching Competitive Learning Network

Assume there are N data vectors in d-dimensional space, $\{\vec{x}_n\}_{n=1}^{N}$ and $\vec{x}_n \in R^d$. Then the process of data clustering can be defined as follows: Find $\{\vec{\omega}_m\}_{m=1}^{M}$ in R^d, to minimize the average distortion or the mean squared error (MSE) given by

$$MSE^{d} = \frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn}\left\|\vec{x}_n - \vec{\omega}_m\right\|^2 \tag{1}$$

where M is the cluster number, $N_m = \sum_{n=1}^{N} z_{mn}$ is the number of data belonging to cluster C_m and $N = \sum_{m=1}^{M} N_m$, where the variable z_{mn} indicates the membership of data

$\vec{x}_n$ to cluster C_m, i.e., $z_{mn}=1$ if $\vec{x}_n \in C_m$ and 0 otherwise. $\vec{\omega}_m$ is the cluster center of C_m, and $\|\cdot\|$ is the L_2 norm.

Like other competitive learning models, the BCL network takes a winner-take-all iterative scheme to update the cluster centers (also called the synaptic vectors), and thus to optimize the MSE measure. The main algorithm can be described as

$$\begin{aligned} &\vec{\omega}_c(t+1)=\vec{\omega}_c(t)+\alpha_c(\vec{x}-\vec{\omega}_c(t)) \\ &\text{for} \quad c=\arg\min_j \left\|\vec{x}-\vec{\omega}_j\right\| \end{aligned} \tag{2}$$

where $\vec{x}$ is a randomly selected input data point, t represents the current step of competitive learning, $\vec{\omega}_c$ denotes the synaptic vector nearest to $\vec{x}$, and α_c is the learning rate. If the geometrical measurements of $\vec{\omega}_c$ between the current competition step t_c and the previous activated step t_l surpass the current level's thresholds,

$$\begin{cases} ang\left(\vec{x}(t_c)-\vec{\omega}_c, \vec{x}(t_l)-\vec{\omega}_c\right) > \varphi_0 \\ \min\left(\left\|\vec{x}(t_c)-\vec{\omega}_c\right\|, \left\|\vec{x}(t_l)-\vec{\omega}_c\right\|\right) > d_0 \end{cases} \tag{3}$$

it is usually an appropriate moment to split the synaptic vector. The nodes that are seldom activated are judged as the 'dead unit', which will be pruned. This heuristic mechanism of automatic insertion of new nodes and deletion of superfluous nodes assures appropriate cluster number can be used at different hierarchical resolution level. For more details about BCL, one can refer to [7].

2.2 Kernel-Based BCL in Reduced Empirical Feature Space

The MSE based criteria employed for clustering implicitly imposes the assumption of hyper-spherical or hyper-ellipsoidal enclosed distribution shapes for each cluster [8]. If the sub-clusters are not linearly classifiable, the results got by this kind of clustering method sometimes don't make sense. Methods, including the kernel-K-means [9], support vector clustering [10], and their variants are proposed to solve this problem by adopting the strategy of nonlinearly transforming the data into a high-dimensional feature space and then performing the clustering algorithm within this feature space. However, the computational complexity for these kinds of kernel-based clustering algorithms may become huge for large data sets. The situation may become more serious for clustering algorithms based on competitive learning of neural networks, where the winner synaptic vector needs to be updated for each randomly selected sample $\vec{x}_n$. If the number of data set is large, the complexity of the expression for the synaptic vector could become tremendous as the iteration process goes on. So it's not practical to run the BCL algorithm for clustering the multi-view face samples directly in the kernel induced feature space. While for a given data set, the algorithm only needs to perform in a subspace of the feature space spanned by the images of the training data $\{\phi(x_n)\}_{n=1}^N$. This subspace can be embedded into an Euclidean space with all the geometrical measurements between each pair of $\phi(x_n)$ unchanged [11]. From

both the theoretical and practical points of view, it is more convenient to access the empirical feature space than the feature space.

We now formulate how to explicitly map samples from data space to the empirical feature space in detail. The kernel matrix K can be decomposed as

$$K = [k_{ij}]_{N\times N} = P_{N\times r}\Lambda_{r\times r}P^T_{r\times N} \tag{4}$$

where $k_{ij} = k(x_i, x_j)$, $\Lambda_{r\times r}$ is a diagonal matrix containing only the r positive eigenvalues of K in decreasing order and P consists of the eigenvectors corresponding to the positive eigenvalues. The map $\phi^e : R^d \to R^r$ from the d-dimensional input data space to the r-dimensional empirical feature space can be expressed explicitly as

$$x \to \Lambda^{-1/2}P^T\left(k(x,x_1),k(x,x_2),\ldots,k(x,x_N)\right)^T \tag{5}$$

The rank r of the kernel matrix may still be very high for data sets where the number of examples N is large. For example, for our multi-view face sample clustering task, there are more than 10000 examples. So practically we select the first l eigenvectors in P, and map the face samples into the l-dimensional reduced empirical feature space where the BCL algorithm can be carried out explicitly. Determining the reduced dimension of the empirical feature space is a tricky problem. Usually it is a trade-off between clustering accuracy and computation efficiency.

2.3 Kernel BCL Based Multi-resolution Clustering Tree

Among the parameters of BCL, the distance threshold d_0 represents the resolution level at which the BCL partitions the face data set. With a large value of d_0, the BCL model will give a coarse clustering result. On the other hand, a relatively small value of d_0 means that the BCL model "views" the data set under a fine resolution.

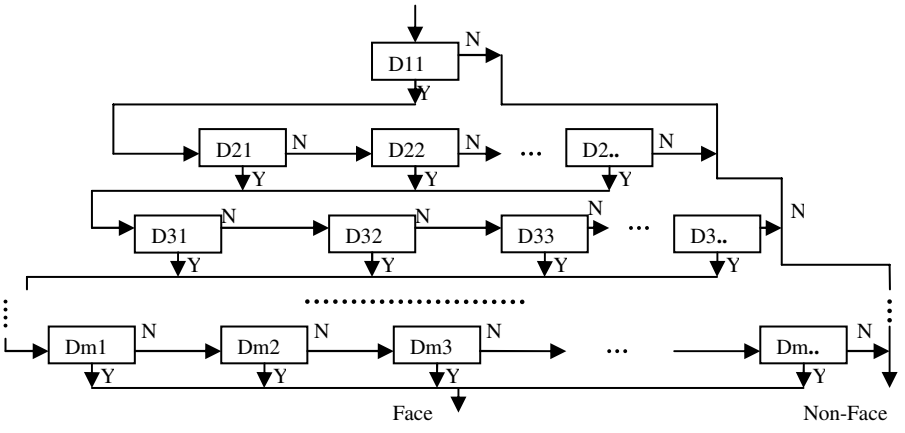


Fig. 1. Cascade tree structure of our self-created multi-resolution face detector

The construction of multi-resolution clustering tree is an iterative algorithm. At the beginning, the tree contains only one node. At later iteration round under a given d_0, it dynamically splits and prunes its nodes along with the BCL until the network reaches a dynamical equilibrium. Thus presents a good clustering result with an appropriate number of nodes under the given detection level. Using an appropriately decreasing sequence of $d_0(t)$ (t = 1, 2,…, m), a multi-resolution hierarchical cluster of the face images is created automatically, as shown in Fig. 1, where m is the number of layers, i.e., the depth of the multi-resolution tree, D is the whole face training set, at any layer t, $D = \bigcup_k D_{tk}$ *and* $D_{tk} \cap D_{tj} = 0$, for $k \neq j$.

3 Support Vector Machines and Boosted MPP Classifiers

Support vector machines maps the data into the high-dimensional feature space F via a map $\phi : X \to F$, $x \to \phi(x)$. This map is implicitly determined by defining the dot products in F by kernel functions $k(x, x') = \phi(x) \cdot \phi(x')$. The SVM algorithm finds a linear separating hyperplane $H_{\omega,b} : \omega^T \phi(x) + b = 0$ which separates the data in F by the largest margin. ω is the normal vector of $H_{\omega,b}$, which has the following form

$$\omega = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{6}$$

Those training examples x_i with $\alpha_i > 0$ are called Support Vectors (SVs). The classification rule of the SVM is

$$f(x) = \operatorname{sgn}\left(\omega^T \phi(x) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + b\right) \tag{7}$$

Although SVM has shown potential and promising performance in face detection, the common drawback of SVM-based methods lies in their detection speed due to the relatively complicated computation of kernel functions. Specifically, the time complexity of a SVM classification is characterized by the number of SVs. In [12] Chen et al. proposed to use the mirror point pairs and a multiple classifiers system to reduce the classification time of SVM. Here we employ Chen's method and modify it to fit the case where the two data classes are highly asymmetric.

For $v \in X$, the distance from its image $\phi(v)$ to the hyperplane $H_{\omega,b}$ can be denoted by

$$d\left(\phi(v), H_{\omega,b}\right) = \frac{\left|\omega^T \phi(v) + b\right|}{\|\omega\|} = \frac{\left|\sum_{i=1}^{N} \alpha_i y_i k(v, x_i) + b\right|}{\left(\sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j k(x_i, x_j)\right)^{\frac{1}{2}}} \tag{8}$$

Furthermore, its mirror vector, m_v with respect to the hyperplane $H_{\omega,b}$ in feature space F, can be computed as follows

$$m_v = \phi(v) - 2 f(v) d\left(\phi(v), H_{\omega,b}\right) \frac{\omega}{\|\omega\|} \tag{9}$$

Given a pair of mirror points $(\phi(v), m_v)$ in the feature space, for any $z \in X$, the MPP classification rule [12] is defined as follows

$$g_{\phi(v),m_v}(z) = \begin{cases} f(v) & if\ d(\phi(z),\phi(v)) \le d(\phi(z),m_v) \\ -f(v) & otherwise \end{cases} \tag{10}$$

Since the map from input space to the feature space is implicit, computation of a mirror classifier's decision result is obtained in terms of the dot product, i.e. the kernel function of points in input space. For the given sample $v \in X$, if the pre-image of its mirror point m_v in feature space exists (i.e., there's a point $q \in X$ such that $m_v = \phi(q)$), the SVM classifier can be replaced by an equivalent MPP classifier with no loss in generality. However, in practice, the pre-image of m_v usually does not exist except for trivial cases. So the pre-image of m_v can only be approximated by a point $q \in X$ with $m_v \approx \phi(q)$, which means the SVM can't be approximated precisely by a single MPP classifier. To improve the generalization ability and tolerate the failure of single MPP classifier, we take the strategy of integrating the decisions made by multiple MPP classifiers to approximate the classification rule of the original SVM.

Returning to Equation (7), we note that the decision function $f(x)$ of a SVM classifier is just a linear combination of kernel functions located around the SVs. So we take these set of SVs $\Omega_{sv} = \Omega_{sv}^{+1} \cup \Omega_{sv}^{-1}$ to construct the approximate MPP classifiers, which lie on the position significant to the classification boundary. For any SV $v_k \in \Omega_{sv}^i$ ($i = \pm 1$), we approximate the mirror, m_k^v, of its image $\phi(v_k)$ in feature space by $\beta_k^* \phi(q_k^*)$, where $q_k^* \in \Omega_{sv}^j$ is another SV belonging to the opposite class. The β_k^* and q_k^* can be computed as follows

$$(\beta_k^*, q_k^*) = \arg \min_{\beta \in R, q \in \Omega_{sv}^j} d(\beta\phi(q), m_k^v) \tag{11}$$

where Ω_{sv}^j ($j \ne i$) denotes the set of SVs from the opposite class to v_k. If the Gaussian kernel is adopted, the MPP classifier's decision function (10) can be reformulated as

$$MPP_k(x) = f(v_k)\,\mathrm{sgn}(k(x, v_k) - \beta_k^* k(x, x_k^*) - b_k) \tag{12}$$

where b_k is a bias term. By varying v_k ($k = 1, \ldots, N_s$), N_s approximate MPP classifiers can be generated, where N_s is the number of SVs. Note some of the support vectors may be selected repeatedly, so at last a classifier pool composed of L ($L \le N_s$) MPP classifiers is remained by removing the repetitions.

As discussed before, since each MPP classifier is just an approximation of the original SVM, they must be assembled together to achieve the comparative generalization ability as SVM. Meanwhile, in order to ensure fast classification, we must focus on a small set of critical classifiers which satisfies the accuracy needs of the current level. The problems of classifier selection and combination can be properly solved by the boost learning method [1], which takes into account both classification accuracy and efficiency.

4 Experimental Results

4.1 On Single SVM and the Boosted MPP Classifiers

To show the classification effectiveness and speed improvement of the boosted MPP classifiers as a single detector, we do the comparison experiment on a data set contains 5000 frontal face and 5000 non-face images, which are divided into a training set of 4000 examples and a test set of 1000 examples respectively. We measure the classification performance by false positive rate, given that the detection rate is fixed at 99% on the training set. Initially SVM was trained on the training set images. The kernel used was Gaussian RBF with a standard deviation σ of 3.5 and C set to 1. The SVM with 761 SVs misclassifies 1084 and 297 non-face images as faces respectively on training set and test set. We take these as the baseline for comparison. The approach, introduced in Section 3, is used to generate the boosted MPP classifiers to approximate the SVM's decision rule. The false positive rate curves for the training and test sets by the boosted MPP classifiers with different number of MPP classifiers are shown in Fig. 2(a). As is shown, when the number of MPP classifiers in use is 19, the obtained false positive rates for training and test data sets are respectively 33.9% and 36.1%, and at the same time, the speedup ratio is nearly 20. The best correct rates achieved by the boosted MPP classifiers for training and test set are 92.6% and 90.3% respectively, where the number of vectors involved in classification is 240(i.e., the speedup ratio is 3.17). By adjusting the threshold, it achieves the corresponding false positives of 1143 and 308 by fixing the detection rate at 99% on training set.

As a node of the cascaded multi-layer detector, the boosted MPP classifiers needn't achieve a very low false positive rate. For example, for a 20-layer detector, to anticipate an overall false positive rate at 10^{-6}, the false positive rate in each single layer only needs to be about 50% ($0.5^{20} \approx 10^{-6}$). Fig. 2(b) shows the ROCs, computed on the test data set, of the original SVM composed of 761 SVs, MPP-1 (the boosted MPP classifiers composed of 38 SVs) and MPP-2 (the boosted MPP classifiers composed of 2 SVs). It can be seen that the approximating performances of the boosted MPP classifiers at different points on ROCs are steady, and the MPP classifier with only two support vectors can be qualified to replace the original SVM for the first layer of our multi-resolution tree detector.

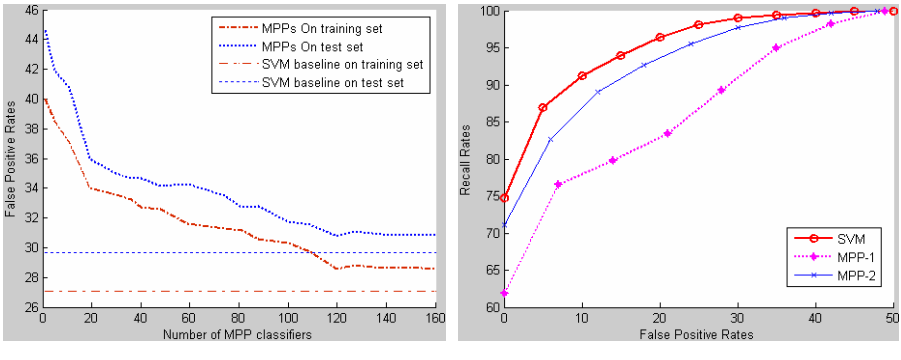


Fig. 2. (a) False positive rates of boosted MPP classifiers with different number of SVs. (b) ROCs of original SVM classifier and the boosted MPP classifiers composed of 2 and 38 SVs.

4.2 On Multi-view Face Detection

For the training of multi-resolution tree face detector, nearly 15000 face samples of size 20x20 are adopted, covering all kinds of views. The non-face samples are generated from about 1000 background images. The distinguishing characteristic of our face detector is that there's no need to label each face sample manually by the angle of its view. With these data sets, we train the multi-view face detector with the multi-resolution face clustering tree structure in Fig. 1. For each layer of the detector, training SVM using each of the face sub-clusters with the non-face images, a series of SVM classifiers are obtained. We then replace all the SVMs by its corresponding boosted MPP classifiers to improve the overall detection speed with no loss in accuracy. Note the branching splitting learning strategy of the cascade structure is interwoven with the training procedure of MPP classifiers. As more MPP classifiers are added, the face and non-face classification boundary becomes more elaborate and meanwhile the lower false positive rate can be achieved by fixing the detection rate. When the false positive rate meets the anticipate needs, the iteration stops, which means the detection resolution level is fine enough to achieve the goal.

Finally, the detector is constructed in the following way: there are 1, 4 and 10 MPP classifiers on the first 3 resolution levels. It rejects about 80% of non-faces, while retaining 99% of training faces. The following 5 levels consist of 141 MPP classifiers and can reject about 93% of non-faces which passed through the first 3 levels, and retain 98% train faces. At the bottom level, the detector reaches a detection rate of about 95% and a false positive rate of about $3x10^{-6}$.

Table 1. Detection rates for various numbers of false positives on the MIT + CMU test set

Detector	False detections				
	10	31	65	95	167
Our(20)	79.4%	88.7%	91.1%	93.2%	94.0%
AB(24)	76.1%	88.4%	92.0%	92.9%	93.9%
FB(20)	83.6%	90.2%	92.5%	93.6%	94.2%

The MIT+CMU frontal-view test set [13], which is composed of 125 images containing 481 frontal faces, is used to test the frontal face detection performance. Our KBCL-based self-created tree detector (OUR (20)) is compared with the cascade of AdaBoost classifiers (AB(24)) with training examples of size 24x24 [1] and the cascade of FloatBoost classifiers (FB(20)) with training examples of size 20x20 [2]. The results are shown in Table 1. From the experimental results, it can be seen that our system outperforms the AdaBoost detector constructed by Viola, and is competitive to Li's FloatBoost detector for the task of the frontal face detection, although our detector is designed to cover all views.

The CMU Schneiderman's profile test set [14] is used to test the performance for multi-view face detection. This data set consists of 208 images with 347 profile faces. The detector's complexity in structure of Viola's decision tree method AB(20) (as implemented by ourselves using training examples of size 20x20) and our KBCL-based self-created tree detector method are compared in Table 2. It can be seen that our method achieves better accuracy while with much fewer MPP classifiers, which

Table 2. Comparison results of the structure of the detector for multi-view face detection

Detector	Detection Rate (%)	# of False Positives	# of Haar-like or MPP Classifiers
Decision tree based	70.4	98	10478
	83.1	700	8704
Self-created tree based	74.6	15	5978
	80.5	101	4069
	91.4	700	2835

means that the MPP classifiers in our detector are able to capture the discriminative characteristics of faces more accurately than Viola's weak classifiers composed of simple haar-like features, thus allowing the development of simpler and more compact structure. The ROCs for both frontal and profile face detection are shown in Fig. 3.

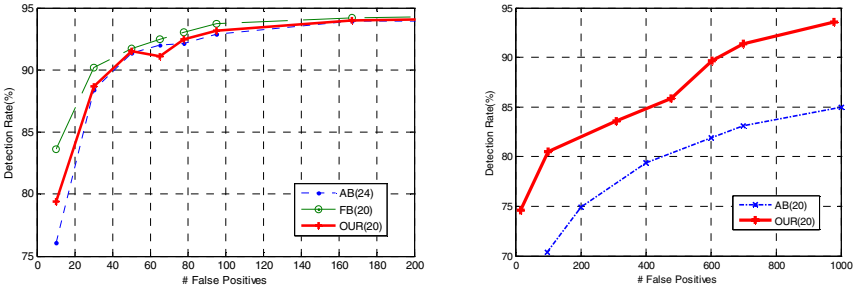


Fig. 3. ROCs for comparison on standard test sets. (a) is on CMU+MIT frontal face test set. (b) is on CMU profile test set.

5 Conclusions

In this paper, the KBCL-based multi-resolution tree structure was developed to address the problem of detecting multi-view faces with high detection rate and low false positive rate. Instead of using the predefined view partition of face samples based on *a priori*, our scheme divides the sample space automatically by using the branching competitive learning method to cluster the multi-view faces in the reduced empirical feature space. The tree node classifier is composed of a series of boosted MPP classifiers which have the approximate generalization capability as SVM, but need less computational burden. According to the visual selection strategy, which allows background regions of the image to be quickly rejected by the first few classifiers while spending more computation on the promising 'face-like' regions, both the clustering of multi-view face samples and the constructing of classifiers for each tree node are fulfilled in a coarse-to-fine manner. Since the structure created is data dependent and don't need any priori knowledge, it can be easily generalized to other object detection problems. The experimental results show the efficiency of our method.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No.60775008), and the National High Technology Research and Development Program (863 Program) of China (Grant No.2007AA01Z196).

References

1. Viola, P., Jones, M.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154 (2004)
2. Li, S.Z., Zhang, Z.: Floatboost Learning and Statistical Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1112–1123 (2004)
3. Wu, B., Ai, H., Huang, C., Lao, S.: Fast Rotation Invariant Multi-View Face Detection Based on Real AdaBoost. In: Proceeding of FGR 2004, Seoul, May 2004, pp. 79–84 (2004)
4. Fleuret, F., Geman, D.: Coarse-to-Fine Face Detection. International Journal of Computer Vision 41(1), 85–107 (2001)
5. Jones, M., Viola, P.: Fast Multi-View Face Detection, MERL Technical Report, vol. 96 (July 2003)
6. Huang, C., Ai, H.Z., Li, Y., Lao, S.H.: High Performance Rotation Invariant Multi-view Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(4), 671–686 (2007)
7. Xiong, H., Swamy, M.N.S., Ahmad, M.O., King, I.: Branching Competitive Learning Network: A Novel Self-Creating Model. IEEE Transactions on Neural Networks 15(2), 417–429 (2004)
8. Xu, R., Wunsch, D.: II: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
9. Girolami, M.: Mercer Kernel Based Clustering in Feature Space. IEEE Transactions on Neural Networks 13(3), 780–784 (2002)
10. Camastra, F., Verri, A.: A Novel Kernel Method for Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(5), 801–805 (2005)
11. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.-R., Rätsch, G., Smola, A.J.: Input Space Versus Feature Space in Kernel-Based Methods. IEEE Transactions on Neural Networks 10(5), 1000–1017 (1999)
12. Chen, J.-H., Chen, C.-S.: Reducing SVM Classification Time Using Multiple Mirror Classifiers. IEEE Transactions on Systems, Man and Cybernetics 34(2) (2004)
13. Rowley, H., Baluja, S., Kanade, T.: Neural Network-Based Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(1), 22–38 (1998)
14. Schneiderman, H., Kanade, T.: A Statistical Method for 3D Object Detection Applied to Faces and Cars. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 746–751 (2000)

Multilinear Nonparametric Feature Analysis

Xu Zhang[1], Xiangqun Zhang[2], Jian Cao[1], and Yushu Liu[1]

[1] Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100080
[2] School of Computer Science and Technology, Xuchang University, Xuchang 461000
zhangxu_01@hotmail.com

Abstract. A novel method with general tensor representation for face recognition based on multilinear nonparametric discriminant analysis is proposed. Traditional LDA-based methods suffer some disadvantages such as small sample size problem (SSS), curse of dimensionality, as well as a fundamental limitation resulting from the parametric nature of scatter matrices, which are based on the Gaussian distribution assumption. In addition, traditional LDA-based methods and their variants don't consider the class boundary of samples and interior structure of each sample class. To address the problems, a new multilinear nonparametric discriminant analysis is proposed, and new formulations of scatter matrices are given. Experimental results indicate the robustness and accuracy of the proposed method.

Keywords: Multilinear nonparametric discriminant analysis, Multilinear algebra, Face recognition.

1 Introduction

Feature extraction is a key issue in the fields of pattern recognition, computer vision, etc. A lot of methods have been proposed, and linear discriminant analysis (LDA) [1] is a popular method among them, which has been widely used in face recognition and image retrieval areas. The aim of LDA is to determine a set of optimal projection vectors maximizing the between-class scatter matrix while minimizing the within-class scatter matrix in the projective feature space. However, when dealing with the high complex dimensional data, LDA often suffers from the following problems: (1) The small sample size problem (SSS). When there are not enough training samples, the within-class scatter matrix may become singular, and it is difficult to compute the LDA vectors. (2) The traditional LDA and its variants are based on the assumption that all classes share the Gaussian distribution with the same covariance matrix. It is not always the case for most of the samples from the real world. So it can not perform well in most cases. (3) The number of the final LDA features has an upper limit $C-1$ (C is the number of class in the samples) since the rank of the between class matrix is at most $C-1$. However, it is often insufficient to separate the classes well with only $C-1$ features, especially in the high-dimensional space. (4) When computing between-class scatter matrix, only the centers of classes are taken into

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 576–585, 2010.

account, it can not capture the boundary structure of classes effectively, which has been shown to be essential in classification. There also exists the similar problem when computing within-class scatter matrix. (5) The traditional LDA converts the samples into the large size vectors will create problem to the LDA implementation for dealing with huge scatter matrices, which also known as "the curse of dimensionality".

To deal with the above problems, many schemes have been proposed. Fisherfaces [2] and NLDA [3] could be used to solve the SSS problems. Fisherfaces first projects the samples to their PCA subspace such that within-class scatter matrix of the projected samples is not singular. Then LDA is applied on the PCA subspace. NLDA suggest that the null space spanned by the eigenvectors of within-class scatter matrix with zero eigenvalues contains the most discriminative information, and it select the project vectors maximizing between-class scatter matrix with the constraint that within-class scatter matrix is zero. Ye *et al.* [4] proposed 2DLDA which operated the original samples directly without vectorizing the samples, and avoided the SSS problems and "the curse of dimensionality" effectively. For a two-class problem, a nonparametric technique called NDA [5] was proposed to solve the aforementioned (1)(2)(3)(4) problems. Recently, Z. Li *et al.* [6] have generalized NDA to multiclass problems successfully, and two complementary methods that are based on the principal space and the null space of the within-class scatter matrix are proposed, respectively.

However, most of the algorithms proposed as above can not solve all the aforementioned problems simultaneously. Even some methods suffer intrinsic limitations [7,8]. In the paper, inspired by the idea of [6,4], we propose a tensor-based nonparametric discriminant analysis which can deal with the above problems effectively.

2 Related Work

2.1 Nonparametric Discriminant Analysis

LDA has been widely used in pattern recognition for feature extraction. It constructs the scatter matrices based on the Gaussian distribution assumption. Suppose there are C known sample classes. According to LDA, the reduced dimension by classical LDA is at most $C-1$, and it is usually insufficient by using only $C-1$ features to separate the classes very well. Also only the centers of classes are taken into account when computing between class scatter matrix, so it fails to capture the boundary structure of classes, which has been proven to be important in classification [5]. These underlying problems will lead to instability and low accuracy of classical LDA.

To address the disadvantage of the classical LDA and its variants, Nonparametric Discriminant Analysis (NDA) is proposed. For a two-class problem, NDA

is similar with LDA, the difference between them lies in the definition of between class scatter matrix. The between class scatter matrix of NDA is defined as

$$\begin{aligned}\mathbf{S}_b^{NDA} &= \sum_{i=1}^{N_1} \mathbf{w}(1,i)(\mathbf{x}_i^1 - \mathbf{m}_2(\mathbf{x}_i^1))(\mathbf{x}_i^1 - \mathbf{m}_2(\mathbf{x}_i^1))^T \\ &+ \sum_{i=1}^{N_2} \mathbf{w}(2,i)(\mathbf{x}_i^2 - \mathbf{m}_1(\mathbf{x}_i^2))(\mathbf{x}_i^2 - \mathbf{m}_1(\mathbf{x}_i^2))^T \end{aligned} \tag{1}$$

Where $\mathbf{w}(t,i)$ is the value of weight function and $\mathbf{m}_j(\mathbf{x}_i^t)$ is the local K-NN mean. $\mathbf{m}_j(\mathbf{x}_i^t)$ is defined by

$$\mathbf{m}_j(\mathbf{x}_i^t) = \frac{1}{k}\sum_{l=1}^{k} \widehat{\mathbf{nn}}_l(\mathbf{x}_i^t, j) \tag{2}$$

Where $\widehat{\mathbf{nn}}_l(\mathbf{x}_i^t, j)$ denote is the lth nearest neighbor from class j to the sample vector $\mathbf{x}_i^t$.

From above, NDA uses local mean to estimate the between-class scatter matrix. Local mean is only the simple generalization of class mean. Also NDA is only used for two-class problem, fortunately, a new method has been proposed to overcome the drawback and limitation of NDA as the following section.

2.2 Nonparametric Feature Analysis

Z.Li et al. proposed nonparametric feature analysis method (NFA) which generalized NDA to multiclass classification problem. In NFA, within class ($\mathbf{S}_w^{NFA}$) and between class ($\mathbf{S}_b^{NFA}$) scatter matrices are redefined as following,

$$\mathbf{S}_w^{NFA} = \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i}(\mathbf{x}_p^i - \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, i))(\mathbf{x}_p^i - \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, i))^T \tag{3}$$

$$\mathbf{S}_b^{NFA} = \sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,l,p)(\mathbf{x}_p^i - \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, j))(\mathbf{x}_p^i - \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, j))^T \tag{4}$$

Where $w(i,j,l,p)$ is defined by

$$w(i,j,l,p) = \frac{\min\{d^\alpha(\mathbf{x}_p^i, \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, i)), d^\alpha(\mathbf{x}_p^i, \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, j))\}}{d^\alpha(\mathbf{x}_p^i, \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, i)) + d^\alpha(\mathbf{x}_p^i, \widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, j))} \tag{5}$$

where $d(\mathbf{x}_1, \mathbf{x}_2)$ denote the Euclidean distance between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, $\alpha \in [0, \infty)$ controls the changing speed of $w(i,j,l,p)$ with respect to the distance ratio.

From Eq. (3)-(4), NFA makes full use of the contribution of the K-NN points for the calculation of the scatter matrices. In addition, the within-class scatter matrix of NFA has the nonparametric form. This will lead to a more flexible and accurate estimation of the within class and between class scatter matrix.

However, NFA represents the samples as vectors, thus the size of the vectors could be very large. NFA will suffer from the problem of the computation of eigen-decomposition of certain large matrices, which not only degrades the efficiency but also makes it hard to scale it to large datasets. In order to solve the problem, inspired by the idea of [6,4], we propose a novel method in the next section, which is the generalization of NFA and overcomes the drawbacks of NFA.

3 Multilinear Nonparametric Feature Analysis (MNFA)

Objects of interests in the many applications of computer vision and pattern recognition, such as two-dimensional images and video sequences are naturally described as tensors or multilinear arrays. However, LDA, NDA and NFA are based on vectors, and the samples should be reshaped into vectors when using the methods, which obviously result in high processing cost in terms of increased computational and memory demands. Beyond the implementing issues, it is obvious that reshaping breaks the natural structure and correlation in the original data. Vectorization ignores the fact that tensor objects are naturally multidimensional objects, e.g., 2D images are 2D objects, instead of 1D objects.

In order to address these problems, we further develop a multilinear nonparametric feature analysis algorithm called MNFA as follows.

3.1 Multilinear Algebra and Notations

This section only briefly introduces some useful notations and concepts of multilinear algebra [9].The notational conventions in [10] are used in this paper except some notations have been specified, such as $\widehat{\mathbf{nn}}_l(\mathbf{x}_p^i, i)$. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index, e.g., $m = 1, 2, ..., M$. Vectors are denoted by lowercase boldface letters, e.g., $\mathbf{x}$, and matrices by uppercase boldface, e.g., $\mathbf{U}$; and tensors by calligraphic letters, e.g., $\mathcal{A}$.Tensors are generalizations of scalars which have no indices, vectors which have exactly one index, and matrices which have exactly two indices to an arbitrary number of indices. Zeroth order, first order and two order tensors are called scalars, vectors and matrices, respectively. Those that transform like first-rank tensors are called vectors, and those that transform like second-rank tensors are called matrices

An Mth order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_M}$. The m-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in \mathbb{R}^{I_m \times J_m}$, denoted by $\mathcal{A} \times_m \mathbf{U}$, is defined by an $I_1 \times ... \times I_{m-1} \times J_m \times I_{m+1} \times ... \times I_M$ tensor with entries: $(\mathcal{A} \times_m \mathbf{U})_{i_1 ... i_{m-1} j_m i_{m+1} ... i_M} \stackrel{\text{def}}{=} \sum_{i_m} a_{i_1 ... i_M} u_{i_m j_m}$.

The m-mode product satisfies the following properties.

Property 1. *Given the tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_M}$ *and the matrices* $\mathbf{U} \in \mathbb{R}^{I_n \times J_n}$, $\mathbf{V} \in \mathbb{R}^{I_m \times J_m} (m \neq n)$, *then*

$$(\mathcal{A} \times_n \mathbf{U}) \times_m \mathbf{V} = (\mathcal{A} \times_m \mathbf{V}) \times_n \mathbf{U} = \mathcal{A} \times_n \mathbf{U} \times_m \mathbf{V}.$$

Unfolding $\mathcal{A}$ along the m-mode is denoted $\mathbf{A}_{(m)} \in \mathbb{R}^{I_m (I_1 \times \ldots \times I_{m-1} \times I_{m+1} \times \ldots \times I_M)}$, and the column vectors of $\mathbf{A}_{(m)}$ are the m-mode vectors of $\mathcal{A}$. The scalar product of two tensors of the same dimensions is defined as: $\langle \mathcal{A}, \mathcal{B} \rangle \stackrel{\text{def}}{=} \sum_{i_1} \sum_{i_2} \ldots \sum_{i_M} a_{i_1 i_2 \ldots i_M} b_{i_1 i_2 \ldots i_M}$. Further more, the Frobenius norm of a tensor $\mathcal{A}$ is defined as $\|\mathcal{A}\|_F \stackrel{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$, the distance between tensor $\mathcal{A}$ and $\mathcal{B}$ is defined as $d(\mathcal{A} - \mathcal{B}) \stackrel{\text{def}}{=} \|\mathcal{A} - \mathcal{B}\|_F$.

3.2 Optimal Criterion of MNFA

Assume that there are N training samples represented as the Mth-order tensors, i.e., $\mathcal{X}_i^j \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ denotes the ith tensor object sample of class j.

In MNFA, the new nonparametric within-class scatter and between-class scatter are defined as

$$s_w = \sum_{i=1}^{C} \sum_{l=1}^{k_1} \sum_{p=1}^{N_i} \|\mathcal{X}_p^i - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i)\|_F^2 \tag{6}$$

$$s_b = \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \sum_{l=1}^{k_2} \sum_{p=1}^{N_i} w(i,j,p,l) \|\mathcal{X}_p^i - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j)\|_F^2 \tag{7}$$

Where $w(i,j,l,p)$ is defined by

$$w(i,j,l,p) = \frac{\min\{d^\alpha(\mathcal{X}_p^i, \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i)), d^\alpha(\mathcal{X}_p^i, \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j))\}}{d^\alpha(\mathcal{X}_p^i, \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i)) + d^\alpha(\mathcal{X}_p^i, \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j))} \tag{8}$$

The aim of MNFA is to pursue multiple interrelated projection matrices, i.e., subspaces, which maximize the between class scatter and simultaneously minimize the within class scatter in the low dimensional tensor space in tensor metric described above. In the low dimensional tensor space, within-class scatter and between class scatter become

$$\tilde{s}_w = \sum_{i=1}^{C} \sum_{l=1}^{k_1} \sum_{p=1}^{N_i} \|\mathcal{X}_p^i \prod_{o=1}^{M} \times_o \mathbf{U}_o - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \prod_{o=1}^{M} \times_o \mathbf{U}_o\|_F^2 \tag{9}$$

$$\tilde{s}_b = \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \sum_{l=1}^{k_2} \sum_{p=1}^{N_i} w(i,j,p,l) \|\mathcal{X}_p^i \prod_{o=1}^{M} \times_o \mathbf{U}_o - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \prod_{o=1}^{M} \times_o \mathbf{U}_o\|_F^2 \tag{10}$$

The optimal projection matrices would maximize $\tilde{s}_b$ and minimize $\tilde{s}_w$, that is

$$\left(\mathbf{U}_k' |_{k=1}^{M}\right) = \arg \max_{\mathbf{U}_k |_{k=1}^{M}} \frac{\tilde{s}_b}{\tilde{s}_w} \tag{11}$$

Eq. (11) is equivalent to a higher order nonlinear optimization problem with a higher order nonlinear constraint; thus, it is difficult to find a closed-form solution. Alternatively, we derive an iterative optimization approach similar with [11] to solve the interrelated discriminative subspaces.

3.3 Optimization of k-Mode

This section we consider optimizing the objective function from only one mode of the tensor, i.e.,

$$\mathbf{U}_k' = \arg\max_{\mathbf{U}_k} \frac{\sum_{i=1}^{C}\sum_{\substack{j=1\\j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,p,l)\|\mathcal{X}_p^i \times_k \mathbf{U}_k - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \times_k \mathbf{U}_k\|_F^2}{\sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \|\mathcal{X}_p^i \times_k \mathbf{U}_k - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \times_k \mathbf{U}_k\|_F^2} \tag{12}$$

In the below, we will present the process of solving Eq. (12).

First, we define

$$\tilde{s}_b^k = \sum_{i=1}^{C}\sum_{\substack{j=1\\j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,p,l)\|\mathcal{X}_p^i \times_k \mathbf{U}_k - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \times_k \mathbf{U}_k\|_F^2$$

$$\tilde{s}_w^k = \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \|\mathcal{X}_p^i \times_k \mathbf{U}_k - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \times_k \mathbf{U}_k\|_F^2$$

With simple algebraic computation, we can easily obtain $\|\mathcal{X}_p^i \times_k \mathbf{U}_k\| = \|(\mathbf{X}_p^i)_{(k)}^T \mathbf{U}_k\|$, where $(\mathbf{X}_p^i)_{(k)}^T$ is the k-mode unfolding of the tensor $\mathcal{X}_p^i$; then, we have

$$\begin{aligned}
\tilde{s}_w^k &= \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \|\mathcal{X}_p^i \times_k \mathbf{U}_k - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \times_k \mathbf{U}_k\|_F^2 \\
&= \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \| \left(\mathbf{X}_p^i\right)_{(k)}^T \mathbf{U}_k - \left(\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i)\right)_{(k)}^T \mathbf{U}_k\|_F^2 \\
&= \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \mathrm{tr}\Big\{ \mathbf{U}_k^T \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right) \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right)^T \mathbf{U}_k \Big\} \\
&= \mathrm{tr}\left\{ \mathbf{U}_k^T \left[\sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right) \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right)^T \right] \mathbf{U}_k \right\} \\
&= \mathrm{tr}(\mathbf{U}_k^T \mathbf{S}_w \mathbf{U}_k)
\end{aligned}$$

Where $\mathbf{S}_w = \sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right) \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, i))_{(k)}\right)^T$.

Similarly,

$$\tilde{s}_b^k = \mathrm{tr}(\mathbf{U}_k^T \mathbf{S}_b \mathbf{U}_k)$$

Where

$$\mathbf{S}_b = \sum_{i=1}^{C}\sum_{\substack{j=1\\j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,p,l) \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, j))_{(k)}\right) \left((\mathbf{X}_p^i)_{(k)} - (\widehat{\mathbf{nn}}_l(\mathbf{X}_p^i, j))_{(k)}\right)^T.$$

So, Eq. (12) can be reformulated as

$$\mathbf{U}'_k = \arg\max_{\mathbf{U}_k} \frac{\tilde{s}_b^k}{\tilde{s}_w^k} = \arg\max_{\mathbf{U}_k} \frac{\mathrm{tr}(\mathbf{U}_k^T \mathbf{S}_b \mathbf{U}_k)}{\mathrm{tr}(\mathbf{U}_k^T \mathbf{S}_w \mathbf{U}_k)} \tag{13}$$

Hence, the optimization problem in Eq. (12) derives a special discriminant analysis problem, which can be solved in the same way for the traditional LDA algorithm.

3.4 Optimal Solution of MNFA

As described above, Eq. (11) has no closed-form solution. we derive an iterative algorithm. In each iteration, $\mathbf{U}_1, ..., \mathbf{U}_{k-1}, \mathbf{U}_{k+1}, ..., \mathbf{U}_M$ are assumed known, then we can compute the optimal $\mathbf{U}_k$ as the follows:

$$\mathbf{U}'_k = \arg\max_{\mathbf{U}_k} \frac{\tilde{s}_b}{\tilde{s}_w} = \arg\max_{\mathbf{U}_k} \frac{\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,p,l) \| \mathcal{X}_p^i \prod_{o=1}^{M} \times_o \mathbf{U}_o - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \prod_{o=1}^{M} \times_o \mathbf{U}_o \|_F^2}{\sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \| \mathcal{X}_p^i \prod_{o=1}^{M} \times_o \mathbf{U}_o - \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \prod_{o=1}^{M} \times_o \mathbf{U}_o \|_F^2}$$

Define $\mathcal{Y}_p^i = \mathcal{X}_p^i \prod_{\substack{o=1\\ o\neq k}}^{M} \times_o \mathbf{U}_o$, $\mathcal{Z}_p^j = \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \prod_{\substack{o=1\\ o\neq k}}^{M} \times_o \mathbf{U}_o$, $\mathcal{Z}_p^i = \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, i) \prod_{\substack{o=1\\ o\neq k}}^{M} \times_o \mathbf{U}_o$, then according to **property** 1, we can get

$$\mathbf{U}'_k = \arg\max_{\mathbf{U}_k} \frac{\sum_{i=1}^{C}\sum_{\substack{j=1\\ j\neq i}}^{C}\sum_{l=1}^{k_2}\sum_{p=1}^{N_i} w(i,j,p,l) \| \mathcal{Y}_p^i \times_k \mathbf{U}_k - \mathcal{Z}_p^j \times_k \mathbf{U}_k \|_F^2}{\sum_{i=1}^{C}\sum_{l=1}^{k_1}\sum_{p=1}^{N_i} \| \mathcal{Y}_p^i \times_k \mathbf{U}_k - \mathcal{Z}_p^i \times_k \mathbf{U}_k \|_F^2} \tag{14}$$

It has the similar formulation as Eq. (12). Obviously it can be solved using the above described optimization of k-mode approach. Therefore, the projection matrices can be iteratively optimized, and the entire procedure to optimize the projection matrices is listed in Algorithm. 1.

Algorithm 1. Multilinear Nonparametric Feature Analysis

Input: Given N training samples are represented as the Mth-order tensors, $\mathcal{X}_i^j \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_M}$ denotes the ith tensor object sample of class j. There are N_i samples in class i; T is the iterative times. The final lower dimensions $\ell_1 \times \ell_2 \times ... \times \ell_M$

Output:Projection matrices $\mathbf{U}_k, k = 1, ..., M$

1. Initialize $\mathbf{U}_1^0 = \mathbf{I}_{I_1}, U_2^0 = \mathbf{I}_{I_2}, ..., U_M^0 = \mathbf{I}_{I_M}$

2. For t=1:T do

For k=1:M do

$$\mathcal{Y}_p^i = \mathcal{X}_p^i \prod_{\substack{o=1\\ o\neq k}}^{M} \times_o \mathbf{U}_o \Rightarrow (\mathbf{Y}^i{}_p)_{(k)}, \; \mathcal{Z}_p^j = \widehat{\mathbf{nn}}_l(\mathcal{X}_p^i, j) \prod_{\substack{o=1\\ o\neq k}}^{M} \times_o \mathbf{U}_o \Rightarrow (\mathbf{Z}_p^j)_{(k)},$$

$\mathcal{Z}_p^i = \widehat{\mathbf{nm}}_l(\mathcal{X}_p^i, i) \prod_{\substack{o=1 \\ o \neq k}}^{M} \times_o \mathbf{U}_o \Rightarrow (\mathbf{Z}_p^i)_{(k)}$,

$\mathbf{S}_b = \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \sum_{l=1}^{k_2} \sum_{p=1}^{N_i} w(i,j,p,l) \left((\mathbf{Y}_p^i)_{(k)} - (\mathbf{Z}_p^j)_{(k)}\right) \left((\mathbf{Y}_p^i)_{(k)} - (\mathbf{Z}_p^j)_{(k)}\right)^T$.

$\mathbf{S}_w = \sum_{i=1}^{C} \sum_{l=1}^{k_1} \sum_{p=1}^{N_i} \left((\mathbf{Y}_p^i)_{(k)} - (\mathbf{Z}_p^i)_{(k)}\right) \left((\mathbf{Y}_p^i)_{(k)} - (\mathbf{Z}_p^i)_{(k)}\right)^T$.

Set the matrix $\mathbf{U}_k^t$ to consist of the ℓ_k eigenvectors of the matrix $(\mathbf{S}_w)^{-1}\mathbf{S}_b$, corresponding to the largest ℓ_k eigenvalues.

End for

if $\|\mathbf{U}_k^t - \mathbf{U}_k^{t-1}\| < \varepsilon$, for $k = 1, ..., M$, break and goto **3**.

End for

3. Output the projection matrices $\mathbf{U}_k = \mathbf{U}_k^t \in \mathbb{R}^{I_k \times \ell_k}, k = 1, ..., M$

3.5 Classification with MNFA

With the learned projection matrices, the low-dimensional representation of the training samples $\mathcal{X}_i^j, i = 1, ..., N_i; j = 1, ..., C$, can be computed as $\mathcal{B}_i^j = \mathcal{X}_i^j \times_i \mathbf{U}_1 \times_2 \mathbf{U}_2 \times ... \times_M \mathbf{U}_M$. For a new test data $\mathcal{X}$, we can compute its low-dimensional representation as:

$$\mathcal{B} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times ... \times_M \mathbf{U}_M$$

Then its class label l is predicted to be that of the sample whose low-dimensional representation is nearest to $\mathcal{B}$, that is

$$l = \arg\min_i \|\mathcal{B}_i^j - \mathcal{B}\|, \ j = 1, ..., C$$

4 Experiments

In the Experiments, ORL face database[12] is used to evaluate the effectiveness of our proposed algorithm—MNFA. The database contains 400 images, 10 different images per person for 40 individuals. All images are grey with 256 levels and of size of 112×92 pixels. To simplify the computation of the experiments and to improve the recognition performance, the facial areas were cropped into the final images with same size for matching, such that the two eyes were aligned at the same position. The size of each cropped image in all the experiments is 32×32 pixels.

We randomly select some images per person for training and the rest for testing. We extract the 40 Gabor features with five different scales and eight different directions and each image is encoded as a third-order tensor of size $32 \times 32 \times 40$.

To demonstrate the efficiency of MNFA, we compare it with some algorithms, such as popular 2DLDA [4] and MDA [11]. In order to compare with them fairly, in the algorithms, the projected dimension of the first two modes of MNFA and MDA are both set 5×5, the projected dimension of 2DLDA is set 5×5. In

MNFA, the number of KNN of with class and that of between class are both set 6, and we select several typical experimental results which are listed in table 1 and table 2.

Table 1. Recognition accuracy(%) comparison of MNFA, 2DLDA, MDA

Num of Experiments	MNFA	2DLDA	MDA
1	**90.00**	78.75	**90.00**
2	**91.25**	80.00	90.00
3	**91.25**	85.00	88.75
4	**92.50**	85.00	85.00
5	**95.00**	87.50	90.00
6	85.00	83.75	**90.00**
7	**92.50**	82.50	86.25
8	92.50	77.50	**95.00**
9	**91.25**	81.25	81.25
10	**86.25**	73.75	81.25

The number of training set is 8×40=320.

Table 2. Recognition accuracy(%) comparison of MNFA, 2DLDA, MDA

Num of Experiments	MNFA	2DLDA	MDA
1	**83.33**	73.33	**83.33**
2	**89.17**	81.67	78.33
3	**93.33**	80.83	75.00
4	**90.00**	73.33	74.17
5	**83.33**	80.00	73.33
6	**87.50**	74.17	79.17
7	**85.83**	76.67	71.67
8	**89.17**	85.83	80.00
9	**89.17**	77.50	75.83
10	**90.83**	83.33	71.67

The number of training set is 7×40=280.

From table 1 and tables 2, we can see that MNFA performs the best. It is more robust and stable than other two methods. The reason is that in MNFA, the boundary between sample classes and interior structure of each class is considered by redefining two scatter matrices. So MNFA can acquire the best results of all the three methods.

5 Conclusions

In the paper, a novel algorithm, called MNFA, has been proposed for supervised dimensionality reduction with the tensor representation. In the algorithm, the

sample objects are encoded as an mth-order tensor. To obtain the optimal solution of MNFA, we introduce a k-mode optimization method which iteratively learn multiple interrelated discriminative subspaces for dimensionality reduction of the higher order tensor. Compared with traditional algorithms, such as 2DLDA and MDA, MNFA effectively avoids the drawbacks, such as Gaussian distribution assumptions of the samples, and "the curses of dimensionality", also it redefines two scatter matrices which consider the class boundary of samples and interior structure of samples. Experimental results show that MNFA is more robust and can acquire the high performance in classification problem.

References

1. Fisher, R.: The Use of Multiple Measures in Taxonomic Problems. Annals of Eugenics 7, 179–188 (1936)
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 711–720 (1997)
3. Ye, J., Xiong, T.: Null Space versus Orthogonal Linear Discriminant Analysis. In: 23rd ACM Conf. Machine Learning, pp. 1073–1080 (2006)
4. Ye, J., Janardan, R., Li, Q.: Two-dimensional linear discriminant analysis. In: Advances in Neural Information Processing Systems (2004)
5. Fukunaga, K.: Statistical Pattern Recognition. Academic Press, London (1990)
6. Li, Z., Lin, D., Tang, X.: Nonparametric discriminant analysis for face recognition. IEEE Trans. Patten Anal. Mach. Intell. 31(4), 755–761 (2009)
7. Wang, X., Tang, X.: Random Sampling LDA for Face Recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2004)
8. Chen, L., Liao, H., Ko, M., Liin, J., Yu, G.: A New LDA-based Face Recognition System Which can Solve the Samll Sample Size Problem. Pattern Recognition 33(10), 1713–1726 (2000)
9. Lathauwer, L., Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21(4), 1253–1278 (2000)
10. Bader, B.W., Kolda, T.G.: Matlab tensor classes for fast algorithm prototyping. Technical Report SAND 2004-5187 (2004)
11. Yan, S., Xu, D., Yang, Q., Zhang, L., et al.: Multilinear Discriminant Analysis for Face Recognition. IEEE Trans. Image Proc. 16(1), 212–220 (2007)
12. The ORL Face Database, http://www.uk.research.att.com/facedatabase.html

A Harris-Like Scale Invariant Feature Detector

Yinan Yu*, Kaiqi Huang, and Tieniu Tan

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{ynyu,kqhuang,tnt}@nlpr.ia.ac.cn

Abstract. Image feature detection is a fundamental issue in computer vision. SIFT[1] and SURF[2] are very effective in scale-space feature detection, but their stabilities are not good enough because unstable features such as edges are often detected even if they use edge suppression as a post-treatment. Inspired by Harris function[3], we extend Harris to scale-space and propose a novel method - Harris-like Scale Invariant Feature Detector (HLSIFD). Different to Harris-Laplace which is a hybrid method of Harris and Laplace, HLSIFD uses Hessian Matrix which is proved to be more stable in scale-space than Harris matrix. Unlike other methods suppressing edges in a sudden way(SIFT) or ignoring it(SURF), HLSIFD suppresses edges smoothly and uniformly, so fewer fake points are detected by HLSIFD. The approach is evaluated on public databases and in real scenes. Compared to the state of arts feature detectors: SIFT and SURF, HLSIFD shows high performance of HLSIFD.

Keywords: Feature detector, image matching, scale invariant, harris.

1 Introduction

Feature detection is a hot topic in computer vision which is widely used in many areas, such as tracking[4], image stitching[5], 3D reconstruction[6,7], camera calibration[8], SLAM system[9], object classification and recognition[10]. In recent years, a lot of work has been done on effective feature detection [1,2,3,4,11,12,13,14,15,16,17]. CSS[11], proposed by Mokhtarian, considers edge with high curvature as corner; Susan[12] and Fast[13] compare the intensity of each pixel with others in its neighborhood to find the corner-like points. Harris[3] constructes a corner model and proposes the Harris Cornerness Function. Similar to Harris, a method proposed by Shi & Tomasis[4] considers the minimum eigenvalue of the Harris Matrix as the cornerness. Apart from the corner detectors mentioned above, SIFT[1], as a region detector, is an approximation of Laplace of Gaussian(LoG), which is proved to be stable and effective in scale-space[14]. From another point of view, LoG is the sum of the eigenvalues of the

* This work is supported by National Basic Research Program of China (Grant No. 2004CB318100), National Natural Science Foundation of China (Grant No. 60736018, 60723005), NLPR 2008NLPRZY-2, National Hi-Tech Research and Development Program of China (2009AA01Z318), National Science Founding (60605014, 60875021).

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 586–595, 2010.

image second derivation matrix(Hessian Matrix). Another region detector which is also based on Hessian Matrix is Determinant of Hessian(DoH). This method calculates the product of the two eigenvalues of the Hessian Matrix, and also performs well in scale feature detection. SURF [2], which is proposed by Herbert Bay et al. and proven to be efficient, is an approximation of DoH.

It can be seen from above that SIFT(DoG/LoG) and SURF(DoH) are both trying to describe the Hessian Matrix with its eigenvalues. These two methods are good at representation of Hessian Matrix, but neither of them describes the matrix very well, for both of them lose much important information about the ratio of the eigenvalues. This value reflects the edge-likelihood of areas. Experiments show that edge areas are unstable in localization and not discriminative in feature description. Edges increase false matches and reduce the accuracy. To cope with this problem, SIFT uses an edge suppression step with a threshold to get rid of edge-like features. The sudden cut process degrades the stability of performance and can not drive out edge-like areas uniformly. SURF, which is calculated very fast, detects a lot of key points full of edges and meaningless features, for it ignores this problem. In order to solve this problem, we propose a new algorithm for feature detection(HLSIFD). To prove the effectiveness of the proposed method, we compare it to the state of arts feature detectors: SIFT and SURF. Experimental results show that HLSIFD outperforms these two methods.

The rest of this paper is organized as follows. Section 2 presents a detailed analysis of feature model and eigenvalue description for feature detection. We then show the proposed detector in Section 3. Finally, experimental results are shown in Section 4 and Section 5 concludes this paper.

2 Feature Model and Eigenvalue Description

Generally, image smoothed by Gaussian filter can be modeled as a Multi-Gaussian-Mix matrix. So feature can also be modeled as Local Gaussian ellipse hill, and the key point is the hilltop, as shown in Fig 1 for an intuitive illustration. The function of this model can be described as follows:

$$M = \kappa \exp\left(-\frac{1}{2}(x,y)\,\Sigma^{-1}\,(x,y)^T\right) + T \tag{1}$$

where κ ,T are parameters and Σ is the Gaussian Covariance Matrix. Note that the second derivative matrix(Hessian Matrix) of the model at point (0, 0) is:

$$H = \begin{bmatrix} M_{xx}(0,0) & M_{xy}(0,0) \\ M_{xy}(0,0) & M_{yy}(0,0) \end{bmatrix} = -\kappa\Sigma^{-1} \tag{2}$$

The Hessian Matrix is composed of the amplitude and the Covariance Matrix which contain most of the model information. The ratio of the eigenvalues of the Hessian Matrix indicates the eccentricity of the Gaussian model, and the eigenvectors depict the orientation. It is important to point out that, the eigenvalues of the Hessian Matrix also have high response in corner areas and edge areas.

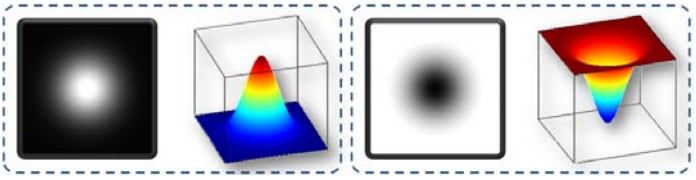


Fig. 1. An intuitive illustration for Gaussian Feature Model in 2D and 3D views. Models with positive κ are shown in the left block, while models with negative κ are shown in the right block.

So one of the advantages of Gaussian model based method is that it can detect both Gaussian-like areas and corner-like areas in scale-space. This property enriches the feature abundance of Hessian based methods. The Hessian Matrix of smoothed image $L = g * I$ is:

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \Rightarrow \begin{cases} \det(H) = \lambda_1\lambda_2 = L_{xx}L_{yy} - L_{xy}^2 = DoH(I) \\ tr(H) = \lambda_1 + \lambda_2 = L_{xx} + L_{yy} = LoG(I) \approx DoG(I) \end{cases} \quad (3)$$

where λ_1 and λ_2 are the eigenvalues of Matrix H. In SIFT, the response of the function(DoG) increases linearly according to the eigenvalues, as shown in Fig 2-(a). We can find that the function has high response even in the edge area where $max(\lambda_1/\lambda_2, \lambda_2/\lambda_1)$ is large. So in SIFT, an edge suppression[1] is added to get rid of edges, leading to unstableness of features on the boundary of the edges area.

SURF uses the multiplication of the eigenvalues, as shown in Fig 2-(b). The response is similar with SIFT(LoG/DoG) in the Salient Feature Area where both λ_1 and λ_2 are high. The problem of SURF is almost the same with SIFT. The product will be large when either of the λ_1 or λ_2 is high. However, SURF does not tackle this problem, resulting in a lot of edges detected, even with a high threshold. This property reduces directly the Repeatability Score. Experiments show that using a step like edge suppression can not tackle this problem well, which has been explained in the last paragraph. Some examples are shown in Fig 3-(a,b).

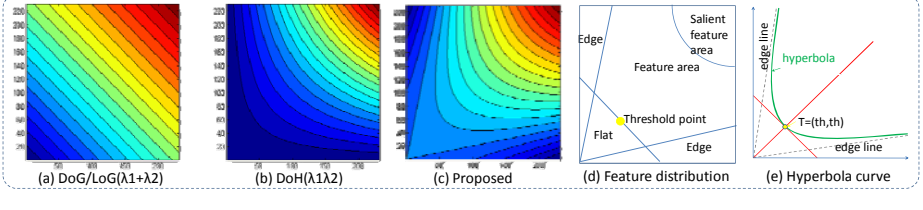


Fig. 2. From left to right: (a)response of $\lambda_1 + \lambda_2$,(b)response of $\lambda_1\lambda_2$, (c)Proposed method, (d)Feature distribution in eigenvalues plane,(e)Hyperbola curve. The angle from edge line to coordinate is α and from edge line to 45^0 line is θ. T is the threshold point. The hyperbola is decided by θ(or α) and T.

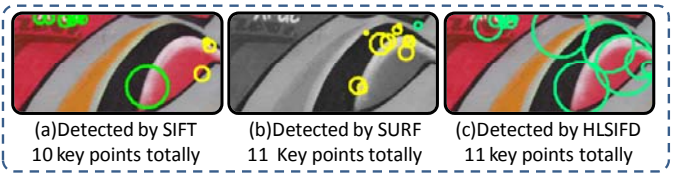


Fig. 3. Key points detected by SIFT, SURF and our method. Our method is proposed in Section 3. Green circles are good feature areas; Yellow circles are edge areas which are not stable.

Actually, two small eigenvalues represent the flat area in image, while a small value and a large one represent edges. The area is stable key point when the two eigenvalues are large, as shown in Fig 2-(d). Thus, using a function to describe the three point mentioned above is the standard mission of feature detectors. Supposing the feature-like point has high response, the detector function should have these four properties:

1. Low response in flat area.
2. Low response in edge-like area.
3. High response in feature-like areas.
4. Smooth function surface.

Our extensive experimental results show that the last property is very important for the stability of detector. In the following section, a novel algorithm for feature detection is described in detail based on the four properties.

3 Our Method: HLSIFD

3.1 Detector Procedure

Denote an image as I. Since the image is usually noisy, a Gaussian filter is used with $\sigma = \sigma_s$ to smooth the original image: $I_{\sigma_s} = g(\sigma_s) * I$. In order to detect points in scale-space, image pyramid is constructed by smoothing the image with a group of Gaussian Filters. To be more efficient, we down sample the pyramid every s layers, and form several octaves, which are shown in the left of Fig 4. The scale interval of two consecutive octaves is 2. Different from SIFT, we must construct s+2 layers per octave and the Gaussian Filters are sampled uniformly in scale-space with a factor of k, where $k = 2^{1/s}$. For each octave, The Gaussian Filters are: $g_i = g(\sigma_i)$, and $\sigma_i = k^i \sigma_0, i \leq s$ and the Hessian Matrix of the image is:

$$H(\sigma_i) = \begin{bmatrix} L_{xx}(\sigma_i, \sigma_s) & L_{xy}(\sigma_i, \sigma_s) \\ L_{xy}(\sigma_i, \sigma_s) & L_{yy}(\sigma_i, \sigma_s) \end{bmatrix} \tag{4}$$

To satisfy the four properties mentioned in last section, we construct a hyperbola function with a rotation of 45^o anticlockwise. The vertex point of the function stands on the threshold line to filter noises, and the asymptotes are set to filter edges uniformly, as shown in Fig 2-(e). Therefore, the cornerness function is:

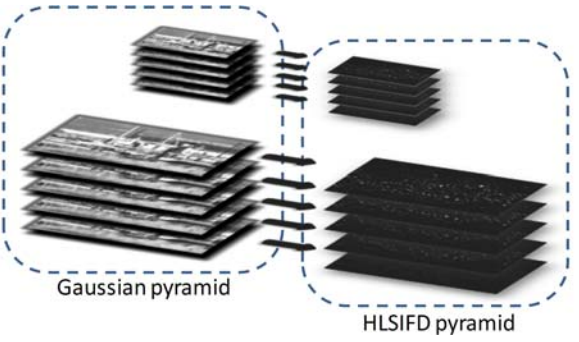


Fig. 4. Gaussian pyramid is shown in left and HLSIFD pyramid is shown in right. Number of octave is decided by the image size, here we only draw up 2 bottom octaves and each octave 5 layers. For each layer, calculate the corner response using Equation (9) from Gaussian pyramid to HLSIFD pyramid.

$$\frac{(\lambda_1+\lambda_2)^2}{2}-\frac{(\lambda_1-\lambda_2)^2}{2tg^2(\theta)}-\frac{(2th)^2}{2} \tag{5}$$

where λ_1 and λ_2 are the two eigenvalues of the Hessian Matrix. Note that $\alpha+\theta=\pi/4$, and the function can be rewritten as:

$$4\left(\frac{1+tg(\alpha)}{1-tg(\alpha)}\right)^2\left(\lambda_1\lambda_2-\frac{1}{4}\left(1-\left(\frac{1-tg(\alpha)}{1+tg(\alpha)}\right)^2\right)(\lambda_1+\lambda_2)^2\right)-2th^2 \tag{6}$$

Let γ represents $4\left(\frac{1+tg(\alpha)}{1-tg(\alpha)}\right)^2$, κ represents $\frac{1}{4}\left(1-\left(\frac{1-tg(\alpha)}{1+tg(\alpha)}\right)^2\right)$, the function can be written briefly as:

$$\gamma\left(\lambda_1\lambda_2-\kappa\left(\lambda_1+\lambda_2\right)^2\right)-2th^2 \tag{7}$$

After normalizing the coefficient of $\lambda_1\lambda_2$ and letting th' equals $\frac{2th^2}{\gamma}$, the final cornerness is:

$$\lambda_1\lambda_2-\kappa\left(\lambda_1+\lambda_2\right)^2-th' \tag{8}$$

Function (8) has the same form as Harris Cornerness Function. The difference is Harris corner is based on Harris Matrix which represents image edge curvature, while Function (8) is based on Hessian Matrix which represents the Gaussian Ellipse Model. In order to detect feature in scale-space, we use the scale normalized Hessian Matrix to express cornerness:

$$D(x,y,\sigma)=\sigma^4(\lambda_1\lambda_2-\kappa\left(\lambda_1+\lambda_2\right)^2-th') \tag{9}$$

where σ^4 is a normalization coefficient in scale-space. We call this method the Harris-like Scale Invariant Feature Detector(HLSIFD). The Gaussian pyramid

is filtered by Equation (9) to get the HLSIFD pyramid, as shown in the right part of Fig 4. Our HLSIFD is:

$$\begin{aligned} D(\sigma_l) &= \sigma_l^4(\lambda_1\lambda_2 - \kappa(\lambda_1+\lambda_2)^2 - th') \\ &= \sigma_l^4(det(H(\sigma_l)) - \kappa trace^2(H(\sigma_l)) - th') \\ &= \sigma_l^4(L_{xx}(\sigma_l,\sigma_i)L_{yy}(\sigma_l,\sigma_i) - L_{xy}(\sigma_l,\sigma_i)^2 \\ &\quad -\kappa\left(L_{xx}(\sigma_l,\sigma_i) + L_{yy}(\sigma_l,\sigma_i)\right)^2 - th') \end{aligned} \tag{10}$$

$$\kappa = \frac{1}{4}\left(1 - \left(\frac{1-tg(\alpha)}{1+tg(\alpha)}\right)^2\right) \Leftrightarrow tg(\alpha) = \frac{1-\sqrt{1-4\kappa}}{1+\sqrt{1-4\kappa}} \tag{11}$$

where $\kappa \in [0, 0.25]$. On one hand Function (10) degenerates into the Determinant of Hessian(DoH) when $\kappa = 0$. On the other hand, when $\kappa \to 0.25 \Rightarrow \alpha \to 45^0$, only the areas with approximately equivalent Hessian eigenvalues would be selected. An un-max-suppression step is used to get the local peak which is considered as a key-point in $3 \times 3 \times 3$ neighborhood in the HLSIFD pyramid. Experimentally, a larger neighborhood is not helpful for increasing the performance. Negative minimums should also be discarded, since they may be edges, noise or even worse saddle points with opposite eigenvalues.

3.2 Matching and Description Procedure

We use Repeatability Score(RS)[18] to evaluate the performance of detector. This score is the ratio of the number of correct matches and key points totally detected on reference image. In order to get this score, we detect key points in reference image and test image first. Then all key points are described by SIFT descriptor for easy comparison. For each key point A in the reference image, we calculate the feature distance(Euclid Distance) from every point detected in the test image to A. Next, the first and second nearest point B and C are found in the test image. Supposing the feature distance between A and B is d_1, and that between A and C is d_2, if $d_1/d_2 < t$, A matchs B. Otherwise A does not match any point in the test image. Finally, RANSAC [19] algorithm is used to eliminate fake matchings and selects the correct matching pairs from all matching pairs.

4 Experimental Results

To evaluate the performance of the proposed detector, we do experiments on the database provided by Mikolajczyk [1] in comparison to the state of arts: SIFT[2] and SURF[3]. This database contains 8 groups images with challenging transformations. Parts of them are shown in Fig 5.

We test our method with the first two lightly transformed images at each group first. The total number of features detected, true positive(Repeatability/correct

[1] http://www.robots.ox.ac.uk/$\tilde{\text{v}}$gg/research/affine/index.html
[2] Provided by Rob Hess:http://web.engr.oregonstate.edu/ hess/index.html
[3] Provided by OpenCV 1.1:http://sourceforge.net/projects/opencvlibrary/

Fig. 5. Database with 8 groups images provided by Mikolajczyk. Each group contains one or two transformations with 6 images and parts of them are shown.

Table 1. Experimental result on the low transformed images

Detector	SIFT			SURF			Ours(HLSIFD)		
type	total	tp[a](%)	p[b](%)	total	tp(%)	p(%)	total	tp(%)	p(%)
Bark(s[c]+r)	4162	14.4	20.5	3481	6.29	11.8	3588	24.6	31.0
Bikes(b)	3202	23.7	30.3	4019	33.5	44.0	4363	43.8	50.9
Boats(s+r)	7986	20.0	27.1	5056	12.2	18.1	4677	31.5	38.2
Graf(s+a)	2837	33.1	41.3	3342	14.8	22.1	2493	35.4	40.9
Leuven(i)	2131	40.5	49.5	3245	39.4	49.0	2841	55.0	61.8
Trees(b)	11279	9.19	14.0	7684	11.1	19.4	7442	17.2	22.5
Ubc(j)	4511	47.3	58.3	4286	68.8	77.3	4025	74.8	80.3
Walls(s+a)	8218	36.2	47.5	6792	8.84	15.7	8107	42.9	51.2

[a] "tp" denotes true positive rate.
[b] "p" denotes precision rate.
[c] "s","r","b","i","a","j" respectively denote scale, rotation, blur, illumination change, affine transformation, and Jepg compression.

match) and precision(ratio between number of correct match and total match) were calculated for comparison, as shown in Table 1. Our method is better in the true positive, not that our method increases the total matching rate, but our method increases the precision. The precision of our method is always higher than others. Experimentally, detector with a higher precision denotes that the key points detected are more accurate and stable, and there are fewer fake, invalid or meaningless key points. t and κ are set to 0.95 and 0.1 in this experiment. For more experiments about our method, we tested HLSIFD in all the images of the database we mentioned above. The results are shown in Fig 6 (a)-(h). The RS of HLSIFD outperforms others in scale, rotation, and affine transformation, since Harris-like function is smooth in eigenvalues space and intensities of edges can be restricted by κ. HLSIFD is better than SIFT and SURF in most groups, but

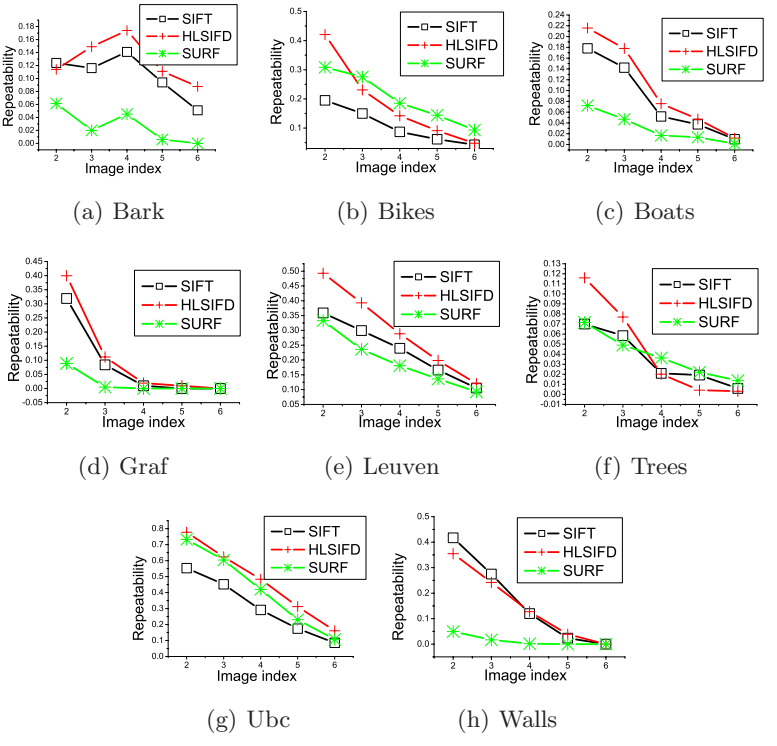


Fig. 6. 8 groups experimental results are shown. Use the first image of each group as reference image and others as test image. From left wo right, top to down:(a) Bark: scale and rotation change. (b)Bikes: blur. (c)Boats: scale and rotation change. (d)Graf: scale and view change image. (e)Leuven: illumination change. (f)Trees: blur. (g)Ubc: Jpeg compression. (h)Walls: scale and view change.

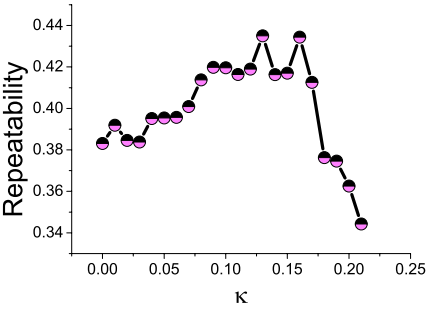


Fig. 7. This figure is one test of influence from κ to the repeatability. The repeatability is increasing with κ from 0 to 0.16, and suddenly drops after 0.16

a little weaker than SURF in seriously blurred image, shown in Fig 6 (b). The computation of our method is nearly same with SIFT, because the most time consuming step is pyramid constructing.

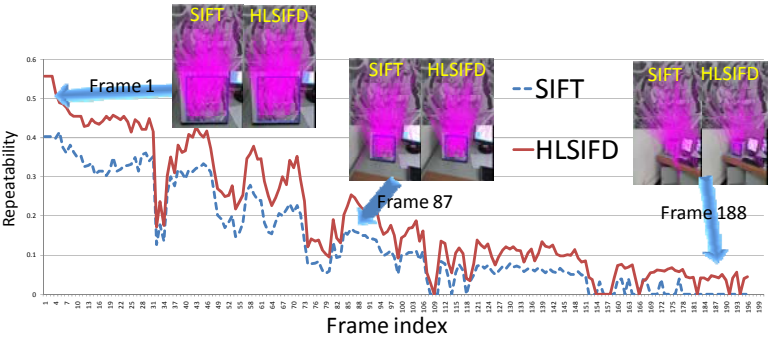


Fig. 8. Experiments on video with scale and view transformation. The reference image was captured from a camera. The results of frame 1, 87 and 188 are shown with SIFT in the left and HLSIFD in the right.

κ is an important parameter in our method. When $\kappa = 0$, it degenerates into Determinant of Hessian(DoH) [17]. We have done many experiments to test the influence of κ to the performance, one of them is shown in Fig 7. We choose κ between 0.04 and 0.15 experimentally and the performance of repeatability would be increased by 10% approximately.

Using feature detection and image matching in video processing is an important application. We compare our method with SIFT in video sequences matching. We study the reference image and matched all the video frames with it. Then the Affine Transform Matrix is calculated by matching points. The video we used contains a large scale change and a gradual view change from 0^0 to 45^0. These transformations are common and have certain representativeness in real scene. Experimental results are shown in Fig 8. The Repeatability Scores of SIFT and HLSIFD reduce continuously with scale and view change. Some motion blurs are presented because of the shaking of our hand-hold camera, and the performance suddenly drops in this frames. The Repeatability Score of HLSIFD is better in most of the time, since it is more stable and detects less fake points by the smooth edge suppression of HLSIFD function.

5 Conclusions

In this paper, we propose a novel scale invariant feature detector: Harris-like Scale Invariant Feature Detector(HLSIFD). The advantage of this detector is the high precision, since fewer fake points could be detected by the proposed method. Unlike SIFT, our method does not need a post-treatment step to cut edge-like points suddenly which would affect the stability. The proposed method can suppress the unstable fake feature points in a uniform way and increase the feature repeatability. Thus, with fewer meaningless key points, features are more significant. Experimental results show the effectiveness of our method.

In the future, we will use our method in real-time image stitching and SLAM system, so a fast algorithm of the proposed detector will be investigated.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Comput. Vis. Image Underst. 110(3), 346–359 (2008)
3. Harris, C., Stephens, M.: A combined corner and edge detection, pp. 147–151 (1988)
4. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE Computer Society Conference on CVPR 1994, June 1994, pp. 593–600 (1994)
5. Li, Y., Wang, Y., Huang, W., Zhang, Z.: Automatic image stitching using sift. In: International Conference on Audio, Language and Image Processing. ICALIP 2008, July 2008, pp. 568–571 (2008)
6. 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(3), 207–232 (2004)
7. Brown, M., Lowe, D.: Unsupervised 3d object recognition and reconstruction in unordered datasets. In: Fifth International Conference on 3-D Digital Imaging and Modeling. 3DIM 2005, June 2005, pp. 56–63 (2005)
8. Telle, B., Aldon, M.J., Ramdani, N.: Camera calibration and 3d reconstruction using interval analysis. In: Proceedings of 12th International Conference on Image Analysis and Processing, pp. 374–379 (2003)
9. Davison, A., Mayol, W., Murray, D.: Real-time localization and mapping with wearable active vision. In: Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, October 2003, pp. 18–27 (2003)
10. Lisin, D., Mattar, M., Blaschko, M., Learned-Miller, E., Benfield, M.: Combining local and global image features for object class recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. CVPR Workshops, June 2005, p. 47 (2005)
11. Mokhtarian, F., Suomela, R.: Robust image corner detection through curvature scale space. IEEE Trans. Pattern Anal. Mach. Intell. 20(12), 1376–1381 (1998)
12. Smith, S.M., Brady, J.M.: Susan - a new approach to low level image processing. International Journal of Computer Vision 23, 45–78 (1997)
13. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection 1, 430–443 (May 2006)
14. Lindeberg, T.: Scale-space theory in computer vision (1994)
15. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
16. Grabner, M., Grabner, H., Bischof, H.: Fast approximated SIFT. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 918–927. Springer, Heidelberg (2006)
17. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vision 60(1), 63–86 (2004)
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vision 65(1-2), 43–72 (2005)
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)

Probabilistic Cascade Random Fields for Man-Made Structure Detection

Songfeng Zheng

Department of Mathematics
Missouri State University, Springfield, MO 65897, USA
SongfengZheng@MissouriState.edu

Abstract. This paper develops the probabilistic version of cascade algorithm, specifically, Probabilistic AdaBoost Cascade (PABC). The proposed PABC algorithm is further employed to learn the association potential in the Discriminative Random Fields (DRF) model, resulting the Probabilistic Cascade Random Fields (PCRF) model. PCRF model enjoys the advantage of incorporating far more informative features than the conventional DRF model. Moreover, compared to the original DRF model, PCRF is less sensitive to the class imbalance problem. The proposed PABC and PCRF were applied to the task of man-made structure detection. We compared the performance of PABC with different settings, the performance of the original DRF model and that of PCRF. Detailed numerical analysis demonstrated that PABC improves the performance with more AdaBoost nodes, and the interaction potential in PCRF further improves the performance significantly.

1 Introduction

Traditional pattern classification methods assume that the class labels are independent to each other. However, in real life data (e.g. sequences, images, videos), the labels of the adjacent data points are correlated. This suggests us take account of the label dependencies in designing classifiers for real life data. For example, Markov Random Fields (MRF) [6], Conditional Random Fields (CRF) [4], and Discriminative Random Fields (DRF) [9], improve the performance of an i.i.d. classification technique by taking into account the spatial dependencies.

In this paper, we are primarily interested in classifying elements (pixels or regions) of a two-dimensional image. Let $\mathbf{X}$ be the observed data from an input image, where $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ with $\mathbf{x}_i$ being the data from the i^{th} image site, and S is the set of all the image sites. Let the corresponding labels for the image be $\mathbf{Y} = \{y_i\}_{i \in S}$, where y_i is the label for image site i.

MRF is usually used in the generative model framework which models the joint distribution of the observed data and the labels. The posterior of the labels given the data can be expressed by Bayes' rule as

$$P(\mathbf{Y}|\mathbf{X}) \propto P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y}). \tag{1}$$

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 596–607, 2010.

The prior distribution of the labels, $P(\mathbf{Y})$, is modelled as MRF. However, the likelihood term, $P(\mathbf{X}|\mathbf{Y})$, is usually very complicated, and it is a distribution in a high-dimensional space (since the image data $\mathbf{X}$ is of high dimension). Thus, it is usually very difficult, if not impossible, to find a good model for $P(\mathbf{X}|\mathbf{Y})$.

On the other hand, CRF and DRF are employed in the discriminative model framework, in which we directly model the posterior distribution of the labels given the data, $P(\mathbf{Y}|\mathbf{X})$. CRF was proposed in the context of segmentation and labelling of 1D sequences, and DRF is generalized version of CRF for 2D image data.

There are two components in DRF model, namely, the association potential and the interaction potential (see Section 2 for details about DRF model). The association potential models the local evidence which ignores the neighborhood information. In [9], the association potential was modelled by a logistic regression classifier, which can only incorporate a limited number of features, leading to restricted classification capability.

AdaBoost [2] is a classification framework which has appealing theoretical properties, and has shown impressive empirical results in a wide variety of tasks, for example, face detection [15,16,17]. This paper takes the advantage of the power of AdaBoost to incorporate more informative features for learning the association potential in DRF, thus overcoming the limitations of logistic regression model in [9]. In the learning stage, we face the problem of unbalanced training set, i.e. far less positive examples than negative examples. AdaBoost cascade [15,16,17] and WaldBoost [13] are usually used to solve this problem. However, the aforementioned methods give a results in $\{-1, 1\}$, while we need a real number for the association potential, which is the logarithm of a probability value as in [9]. To achieve this purpose, we develop Probabilistic version of AdaBoost Cascade (PABC), which calculates the posterior probability of class label when a testing example is presented. PABC is employed to learn the association potential in DRF model, and the interaction potential is learned in the same way as in the original DRF model [9]. The resulting model, Probabilistic Cascade Random Fields (PCRF), enjoys the capability of incorporating far more informative features and a more powerful association potential than the conventional DRF model.

The proposed PCRF was applied to man-made structure detection problem. We compared the performance of PABC with different settings, the performance of the original DRF model and the performance of PCRF. Detailed quantitative measures demonstrate that with more AdaBoost nodes, the overall performance of PABC improves, and with the information from interaction potential, PCRF further removes some false positives and fills in some missing parts of the object.

2 Review of Discriminative Random Fields

Discriminative Random Fields (DRF) model [9] avoids the independence assumption and seek to model the conditional joint distribution of the labels if the data is given, i.e., $P(\mathbf{Y}|\mathbf{X})$. DRF model defines the conditional probability of the labels $\mathbf{Y}$ as:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_{i\in S} A_i(y_i, \mathbf{X}) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i} I_{ij}(y_i, y_j, \mathbf{X})\right), \tag{2}$$

where A_i is the association potential that models the dependencies between the observations and the class labels, and I_{ij} is the interaction potential which models the dependencies between the labels of the adjacent elements (and the observations), and $\mathcal{N}_i$ is a neighborhood of image site i. In this paper, we assume the random field is homogeneous and isotropic, i.e., the functional forms of A_i and I_{ij} are independent of the locations i and j, hence we can simplify the notations as A and I, respectively. This model alleviates the need to model the observation data $P(\mathbf{X}|\mathbf{Y})$ (a necessary step in Bayesian statistics), and it also allows the use of arbitrary attributes of the observations without explicitly modelling them.

The association potential $A(y_i, \mathbf{X})$ reflects the local evidence of the label for image site i. For two-class classification, $y_i \in \{1, -1\}$, the association potential is modelled as [9]:

$$A(y_i, \mathbf{X}) = \log\left(\sigma(y_i \mathbf{w}^T \mathbf{h}_i(\mathbf{X}))\right), \tag{3}$$

where $\sigma(\cdot)$ is logistic regression function:

$$\sigma(y_i \mathbf{w}^T \mathbf{h}_i(\mathbf{X})) = P(y_i|\mathbf{X}) = \frac{1}{1+\exp\left(-y_i \mathbf{w}^T \mathbf{h}_i(\mathbf{X})\right)}. \tag{4}$$

In Eqn. (3) and (4), $\mathbf{h}_i(\mathbf{X})$ is the feature vector extracted from the image data for site i, and $\mathbf{w}$ is the weight vector. In principle, the feature vector $\mathbf{h}_i(\mathbf{X})$ can be any transformation of the image data. The association potential defined in Eqn. (3) makes DRF equivalent to a logistic regression classifier if the interaction potential is set to zero.

To model the interaction potential, let $\boldsymbol{\mu}_{ij}(\mathbf{X})$ be the pairwise feature vector extracted from the image data $\mathbf{X}$ which reflects the property of the image site pair (i, j). Similar to Eqn. (3), the pairwise discriminative term (a probability) is defined as

$$P(y_i, y_j|\mathbf{X}) = \sigma\left(y_i y_j \mathbf{v}^T \boldsymbol{\mu}_{ij}(\mathbf{X})\right), \tag{5}$$

where $\mathbf{v}$ is the parameter vector. The interaction potential is modelled as a convex combination of two terms, i.e.:

$$I(y_i, y_j, \mathbf{X}) = \beta\left\{K y_i y_j + (1-K)\left[2\sigma\left(y_i y_j \mathbf{v}^T \boldsymbol{\mu}_{ij}(\mathbf{X})\right) - 1\right]\right\}, \tag{6}$$

where $0 \le K \le 1$. When $K = 1$, the interaction potential boils down to the Ising model, therefore the interaction potential can be thought of as a generalization of the Ising model.

Note that both the association potential $A(y_i, \mathbf{X})$ and the interaction potential $I(y_i, y_j, \mathbf{X})$ depend on the whole image $\mathbf{X}$, not only on the image data at site i or site j. This is different from the traditional classification setting.

The parameters $\boldsymbol{\theta} = \{\mathbf{w}, \mathbf{v}, \beta, K\}$ can be obtained by maximizing the pseudo-likelihood function:

$$\hat{\boldsymbol{\theta}} \approx \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} \prod_{i \in S} P(y_i^m | y_{\mathcal{N}_i}^m, \mathbf{X}, \boldsymbol{\theta}), \tag{7}$$

where m indexes the training images and M is the total number of training images, and

$$P(y_i | y_{\mathcal{N}_i}, \mathbf{X}, \boldsymbol{\theta}) = \frac{1}{Z_i} \exp\left\{ A(y_i, \mathbf{X}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{X}) \right\}, \tag{8}$$

with Z_i as the normalization factor. The pseudo-likelihood function given in Eqn. (7) can be maximized by linear search method [9], Newton's method, or stochastic gradient method.

3 Probabilistic AdaBoost Cascade

As a classification algorithm, AdaBoost [2] combines a set of weak classifiers (features) to form a strong classifier, and the obtained strong classifier is

$$H(\mathbf{x}) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\} \in \{-1, 1\}, \tag{9}$$

where $\mathbf{x}$ is the input example, $h_t(\mathbf{x}) \in \{-1, 1\}$ is the weak classifier (feature) selected at the t^{th} iteration with weight α_t, and T is the total number of iterations. It is well-known that there is a deep relation between AdaBoost and the additive logistic regression model [3], i.e.,

$$p(y|\mathbf{x}) = \frac{\exp\left\{ y \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\}}{\exp\left\{ \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\} + \exp\left\{ -\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\}}, \quad \text{with} \quad y \in \{-1, 1\}. \tag{10}$$

In applications, we usually have a limited number of positive examples, but abundant negative examples, that is, the training set is highly unbalanced. AdaBoost cascade [15,16], and its variant, Boosting Chain [17], are successfully used for object detection with unbalanced training set. WaldBoost [13], using sequential likelihood ratio test in decision making, implicitly builds cascade structure after every weak classifier is added. In AdaBoost cascade, before training each AdaBoost node, we can bootstrap negative examples in case there are not enough negative examples, as shown in Fig. 1. A testing example will be classified as positive if it can pass all the AdaBoost nodes; otherwise, it will be classified as a negative example.

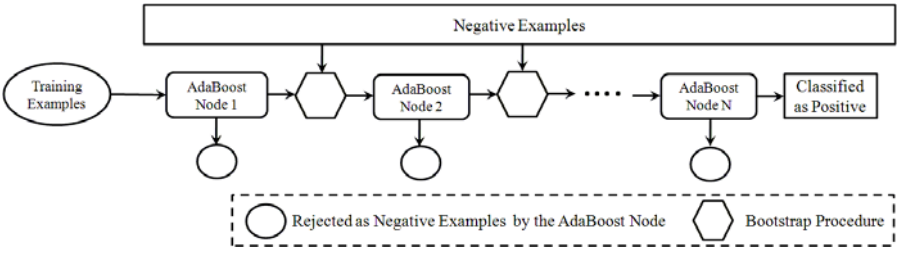


Fig. 1. AdaBoost Cascade which can bootstrap negative examples at each stage

AdaBoost cascade, Boosting chain, and WaldBoost output a value in $\{-1, 1\}$, while in certain application scenarios, we prefer a probability value $P(y = 1|\mathbf{x})$. As such, we develop Probabilistic AdaBoost Cascade (PABC) which has the same structure as the non-probabilistic version, see Fig. 1. The difference is that for each testing example $\mathbf{x}$, PABC outputs the probability $P(y = 1|\mathbf{x})$ instead of ± 1.

The training process of PABC is very similar to that of AdaBoost cascade as shown in Fig. 1, except that we use Eqn. (10) to calculate the probability value when we split the training set. Ideally, we would like to keep all the positive examples in training the AdaBoost nodes, but inevitably we will make some mistakes when splitting the training set. To keep as many positive examples as possible, we put restriction on the false negative rate for each split by the AdaBoost nodes. For each split, we also calculate the proportion of the positive examples among the rejected examples. The detailed training process is given in Fig. 2.

Let $\mathbf{x}$ be a testing example, and y be the corresponding label. We regard each of the subset rejected by the AdaBoost node $\mathcal{S}_{\text{RJ}}$ as a classifier as well, and it outputs the probability

$$P(y = 1|\mathbf{x}) = P(y = 1|\mathcal{S}_{\text{RJ}}, \mathbf{x}) = p_n, \tag{11}$$

that is, the proportion of positive examples in the subset $\mathcal{S}_{\text{RJ}}$. From Fig. 3, it is easy to write out the posterior probability of y given the testing example $\mathbf{x}$ as:

$$P(y = 1|\mathbf{x}) = \sum_{y_1 \in \{-1,1\}} P(y = 1|y_1, \mathbf{x}) P(y_1|\mathbf{x}), \tag{12}$$

and similarly, we have the following recursive formula:

$$P(y = 1|y_{n-1} = \cdots = y_1 = 1, \mathbf{x}) = \sum_{y_n \in \{-1,1\}} P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x}) \times P(y_n|y_{n-1} = \cdots = y_1 = 1, \mathbf{x}), \tag{13}$$

where $y_i \in \{-1, 1\}$, $P(y_n|y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is the probability calculated by the n^{th} AdaBoost node using Eqn. (10); at a terminal node, $P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is the output by the terminal node using Eqn. (11); if the

- **Input:** An initial training set $\mathcal{S} = \{(\mathbf{x}_i, y_i) : i \in \Lambda\}$, where $\mathbf{x}_i$'s are feature vectors, $y_i \in \{-1, 1\}$ is the label, and Λ indexes all the training examples. We also have a set $\mathcal{S}_{\text{neg}}$ which contains a large number of negative examples. The desired false negative rate for each cascade split fn is also given.
- For $n = 1, \cdots, N$
  1. Train a strong classifier by AdaBoost for node n using the current training set $\mathcal{S}$.
  2. For each $\mathbf{x}_i \in \mathcal{S}$, calculate the probability $P(y_i = 1|\mathbf{x}_i)$ using Eqn. (10).
  3. Split the training set $\mathcal{S}$ into $\mathcal{S}_{\text{RJ}}$ and $\mathcal{S}_{\text{go}}$, where $\mathcal{S}_{\text{RJ}}$ and $\mathcal{S}_{\text{go}}$ are the subset classified by the current AdaBoost node as negative and positive examples, respectively; when making this split, we adjust the threshold such that the false negative rate is at least as small as the given value fn; for $\mathcal{S}_{\text{RJ}}$, calculate the proportion of positive examples p_n.
  4. If there are not enough negative examples in $\mathcal{S}_{\text{go}}$, bootstrap negative examples from the given set $\mathcal{S}_{\text{neg}}$; let $\mathcal{S} = \mathcal{S}_{\text{go}}$.
- End For

Fig. 2. Training process of PABC with N AdaBoost nodes

current node is not a terminal node, then $P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is calculated recursively by Eqn. (13). Using Eqn. (12) and the recursive relation (13), we can calculate the output probability by PABC. Thus in the testing stage, PABC integrates information from every node to make decision.

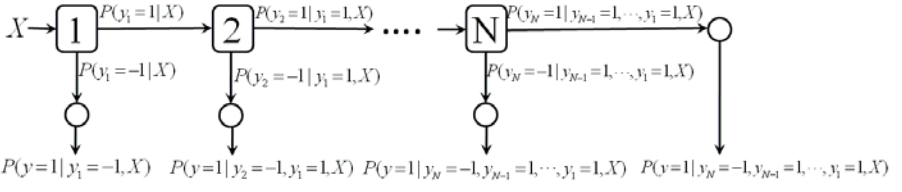


Fig. 3. Testing procedure of PABC. The boxes are the classifiers learned by AdaBoost, and the circles are the terminal nodes of the cascade. y_i is the decision result of the i^{th} AdaBoost node, and y is the output result. The probabilities are calculated according to each AdaBoost node (Eqn. 10) or from the proportions at the terminal nodes (Eqn. 11).

Tu [14] proposed a Probabilistic Boosting Tree (PBT) algorithm, of which PABC is a special case, since the chain structure in PABC is a special case of the tree structure in PBT. However, as the depth of the tree increases, the number of nodes in PBT increases exponentially, which will need much more time in the training stage than PABC. Furthermore, a tree is much more complicated than a chain, thus PBT is more likely to over-fit the data than PABC. In [18], a learning procedure called Probabilistic Boosting Network (PBN) is presented, which is implemented by means of an efficient graph structure. In [18], PBN

was used to classify object and estimate pose parameters at the same time, while in this paper, we are only focus on classification. In PBN, if there is no pose parameter, the graph structure of PBN will boils down to the structure of AdaBoost cascade.

4 Probabilistic Cascade Random Fields

The original DRF model [9] learns the association potential by a logistic regression model. However, the logistic regression model can only incorporate a small number of features, and the classification capability of logistic regression model is not strong. Moreover, logistic regression often does not estimate appropriate parameters, and this is especially true for image data where feature vectors may have a high number of dimensions and possibly there are high degree of correlations among features.

Fortunately, the DRF framework allows a flexible choice of the association potential. By making use of the strong classification ability of Support Vector Machines (SVM), Lee et al. [5] proposed to use probabilistic version of SVM [11] for learning the association potential. Although SVM has good classification performance, it needs a large amount of training time when the feature number and training set are large. More over, SVM does not have an explicit solution to the problem of imbalanced training set which is common in applications.

This motivates us to apply the introduced PABC algorithm to learn the association potential since PABC can deal with a large number of features and a large number of training examples. Due to the powerful feature selection mechanism of AdaBoost, PABC will not select correlated features. Furthermore, PABC is designed for imbalanced data, thus it is less sensitive to imbalanced training set compared to SVM and AdaBoost.

The learned association potential by PABC algorithm is expressed as

$$A(y_i, \mathbf{X}) = \log P(y_i = 1|\mathbf{X}), \tag{14}$$

where $P(y_i = 1|\mathbf{X})$ is fitted by the procedure described in Fig. 2, and calculated for a given sample by Eqn. (12) and Eqn. (13).

This work still adopts the interaction potential as in Eqn. (6), also see [9]. We maximize the pseudo-likelihood function to estimate the parameters $\boldsymbol{\theta} = (\mathbf{v}, \beta, K)$ in the interaction potential, i.e.

$$(\hat{\mathbf{v}}, \hat{\beta}, \hat{K}) \approx \arg\max_{(\mathbf{v},\beta,K)} \prod_{m=1}^{M} \prod_{i \in S} P(y_i^m | y_{\mathcal{N}_i}^m, \mathbf{X}, \boldsymbol{\theta}). \tag{15}$$

To ensure that the log-likelihood is convex and prevent over-smoothing due to the pseudo-likelihood approximation, we assume a Gaussian prior on $\mathbf{v}$ and use the penalized log pseudo-likelihood function [10]

$$l(\mathbf{v}, \beta, K) = \sum_{m=1}^{M} \sum_{i \in S} \left\{ A(y_i, \mathbf{X}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{X}) - \log Z_i \right\} - \frac{1}{2} \mathbf{v}^T \mathbf{v}, \tag{16}$$

where $I(y_i, y_j, \mathbf{X})$ depends on the parameters $(\mathbf{v}, \beta, K)$ as defined in Eqn. (6), and Z_i is a normalization constant which also depends on the parameters $(\mathbf{v}, \beta, K)$. Note that $A(y_i, \mathbf{X})$ is learned by PABC and calculated according to Eqn. (14), therefore in the optimization procedure $A(y_i, \mathbf{X})$ can be ignored since they are constants to $(\mathbf{v}, \beta, K)$. We use gradient ascent to maximize the penalized log pseudo-likelihood function in Eqn. (16).

Given a testing image $\mathbf{X}$, our goal is to find the most probable label configuration $\mathbf{Y}^*$ for $\mathbf{X}$, i.e., solve a Maximum A Posteriori (MAP) problem:

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}), \tag{17}$$

where the probability $P(\mathbf{Y}|\mathbf{X})$ is evaluated according to Eqn. (2) with the learned parameters. Since the probability distribution only contains unary and binary terms, the MAP can be solved by max-flow/min-cut type of algorithms [7]. As in [9], we choose to use iterated conditional modes (ICM) [1] for inference due to its simplicity, and it yields a local maximum of the posterior probability. Given an initial labelling, ICM iteratively maximizes the local conditional probability, that is, for each image site, we update the label by

$$y_i \leftarrow \arg\max_{y_i \in \{-1,1\}} P(y_i|\mathbf{Y}_{\mathcal{N}_i}, \mathbf{X}). \tag{18}$$

5 Experiment

We test the proposed model on the task of man-made structure detection from natural images. The training and testing sets contain 108 and 129 images, respectively, each of size 256×384 pixels. Each image is divided into non-overlapping 16×16 image blocks, and each image block is an image site in our model. The ground truth was generated by manually labelling every image site as *building* and *non-building*. There are 5,203 building blocks and 36,269 non-building blocks in the training set, and 6,372 building blocks and 43,164 non-building blocks in the testing set[1].

5.1 Features

For the man-made structure detection problem, we use the features described in [8,9] as our first set of features, which are based on the weighted histogram of the gradient orientation. Please refer to [8] for more details. We also use different combinations (sum, difference, etc.) of features from [8].

We apply different filters (e.g. Gabor filters, Gaussian filters, Canny Edge detectors) to the original image, and other features are extracted from the filter responses. We notice that most building regions are relatively smooth with small variance while most background regions have cluttered pattern with large variation. This observation inspires us to use mean and variance values of different filter responses (include the original image) inside sub-windows as features.

[1] The original image data and the labels are provided by [9].

For each sub-window, we can also calculate the histograms from each filter response, and use each bin of the histogram as a feature, and the entropy of the histogram can be used as a feature as well to evaluate the regularity of the sub-window. We also notice that man-made structures are primarily characterized by straight lines with horizontal or vertical direction, and this motivates us to extract features from the edge map. In canny edge map, we count the numbers of horizontal and vertical edge points inside each sub-window, and use these numbers as features. The regularity of the building region and the irregularity of the background also make the orientation of the gradient a good discriminator, therefore, we calculate the mean value of the orientation of the gradient inside a sub-window and use it as a feature.

The largest sub-window has size 48×48, and the smallest is of size 6×6. We design the sub-windows such that they must have at least 6×6 intersection with the current image site (a 16×16 window). By doing this, each feature contains neighborhood information to classify the current image site. This feature design strategy is consistent with our notation $P(y_i|\mathbf{X})$, i.e. the class label for image site i depends on the whole image, not only $\mathbf{x}_i$ itself. For each sub-window in the image, the mean, variance, and histogram can be calculated efficiently using integral image [15] and integral histogram [12]. Altogether, we have around 10,000 features for learning the association potential.

In learning the association potential by PABC, the first 4 features selected by the first AdaBoost node are: Variance of the Gabor filter response inside the sub-window at the relative location (-9, -16, 26, 26) to the top-left corner of the current image site, with error rate 0.189; the sum of the first and 21st features from [8], with error rate 0.298; the difference of the second and 17th features from [8], with error rate 0.349; the average number of vertical edge points in the sub-window at the relative location (-16, -9, 19, 26) to the top-left corner of the current image site, with error rate 0.386.

To learn the interaction potential, we use features $\boldsymbol{\mu}_{ij}(\mathbf{X})$ as those used in [9], that is, the difference of two vectors from [8] at image sites i and j, such that the feature vector $\boldsymbol{\mu}_{ij}(\mathbf{X})$ encodes the difference between image sites i and j.

5.2 Results

When applying PABC[2] to learn the association potential for PCRF, we use 5 AdaBoost nodes, and for each of them, we select 120 features. We feed all the positive examples to the first AdaBoost node, and each AdaBoost is learned with 10,000 negative examples. When splitting the training set, the false negative rate is set to be 0.015, and we bootstrap negative examples if necessary. Learning the association potential by PABC needs about 2 hours, and learning the interaction potential by maximizing pseudo-likelihood needs about 5 minutes with 40 iterations to converge. In the testing stage, for each input image, the computer needs about 20 seconds to output the detection result. The computer has a 2GHz CPU and 3.25G Bytes memory.

[2] The PABC is implemented based on the source code provided by [16].

From Eqn. (2), when the interaction potential is set to zero, the PCRF model is reduced to a classification model learned by PABC. In this case, given the observed image data $\mathbf{X}$, the optimal label configuration $\mathbf{Y}^*$ is found by maximizing the class posterior. In another word, the optimal label for the i^{th} site is

$$y_i^* = \arg\max_{y_i \in \{-1,1\}} P(y_i|\mathbf{X}). \tag{19}$$

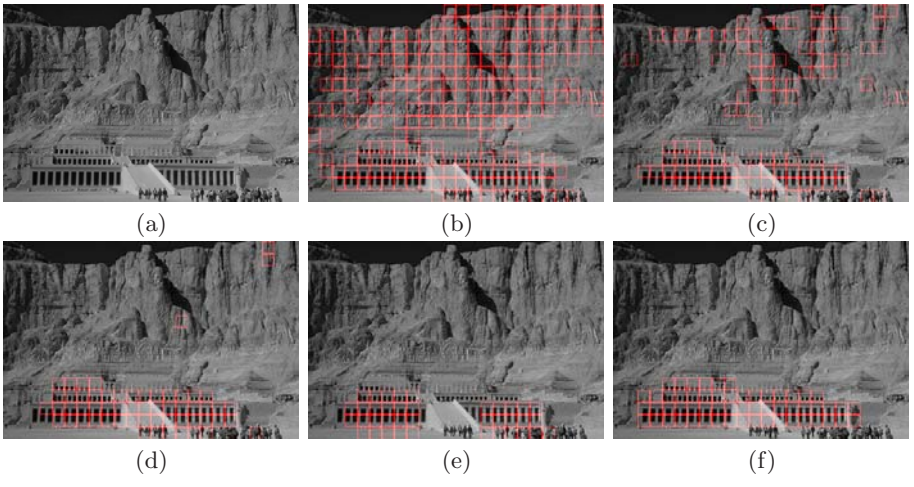


Fig. 4. The experimental result on man-made structure detection, the detected building blocks are marked in red boundary: (a) shows the input image in gray scale; (b) and (c) are the detection results from PABC with 1 and 5 AdaBoost nodes, respectively; (d) and (e) are the detection results from PCRF and the original DRF, respectively; (f) is the the manually labelled result. Please view in color for better visual effect.

Fig. 4 shows the detection result on a testing image. As can be seen from (b), initially, with one AdaBoost node, PABC can detect almost all the building blocks, i.e., it has high detection rate, but it also has high false positive rate. With more AdaBoost nodes, PABC can remove some false positives, as seen from (c). (d) is the result obtained by PCRF, which shows that the interaction potential further removes the false positives, although there are still false positives compared to the manually labelled result in (f). (e) shows the result obtained by the original DRF model[3], from which we can see that the original DRF model has fewer false positives, but it has more false negatives.

Table 1 presents the performance measures of the model with different settings. As we can see, with more AdaBoost nodes, the detection rate decreases, but the false positive rate also decreases, as a result, the site-wise classification error rate decreases monotonically. This is expected because PABC aims

[3] The MATLAB toolbox of DRF model for man-made structure detection was downloaded from http://www.cs.ubc.ca/∼murphyk/Software/CRF/crf.html

at minimizing the error rate. The table shows that with the information from the interaction potential, the PCRF improves the detection rate slightly, but the false positive rate drops significantly.

Table 1. The numerical evaluation result on 129 testing images: PABC n stands for PABC with n AdaBoost nodes

Performance Measures	PABC 1	PABC 2	PABC 3	PABC 4	PABC 5	PCRF
Detection Rate	94.27%	89.01%	83.33%	77.56%	72.18%	72.64%
False Positive Rate	25.37%	16.18%	11.58%	8.37%	6.23%	3.94%
Site-wise Error Rate	22.84%	15.52%	12.24%	10.18%	9.01%	6.95%

Our final result has better detection rate than that reported in [9], but slightly worse false positive rate. The reason is that in [9], the parameters for the association potential and for the interaction potential are estimated simultaneously, while the PCRF model learns the model parameters separately, which might be a suboptimal strategy. Pursuing learning methods which can estimate the parameters simultaneously needs more investigation.

6 Conclusions and Future Works

This paper develops the probabilistic version of AdaBoost cascade (PABC), which outputs a probability value instead of -1/1 value. We use PABC to learn the association potential in the DRF model, resulting the Probabilistic Cascade Random Fields (PCRF) model. We applied the proposed model to the task of man-made structure detection, and compared the performance of PABC with different settings, the performance of the original DRF model, and the performance of PCRF. Detailed qualitative and quantitative analysis showed that PABC improves the overall performance with more AdaBoost nodes. With the information from interaction potential, PCRF further removes some false positives and fills in some missing parts of the object of interest. Our final result is comparable to that reported literature.

In this paper, only the association potential is learned by PABC, while the interaction potential is learned by a simple logistic regression model. Therefore, the current PCRF model still has limited ability to combine more informative features in the interaction potential. Our next step is using PABC to learn the interaction potential. Also, it is desirable to test the proposed approach to other applications and compare to state-of-the-art results, e.g. face detection [15].

Acknowledgement

The author would like to thank the anonymous reviewers for their comments which improve the quality of this paper. Part of this work was supported by CNAS Research Fellowship of Missouri State University.

References

1. Besag, J.: On the Statistical Analysis of Dirty Pictures. J. of Royal Statisticial Society B-48, 259–302 (1986)
2. Freund, Y., Schapire, R.E.: A Decision-theoretic Generalization of on-line Learning and an Application to Boosting. J. of Computer and Sys. Sci. 55 (1997)
3. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Annals of Statistics 28 (2000)
4. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. of ICML (2001)
5. Lee, C., Greiner, R., Schmidt, M.: Support Vector Random Fields for Spatial Classification. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 121–132. Springer, Heidelberg (2005)
6. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, Tokyo (2001)
7. Kolmogorov, V., Zabih, R.: What Energy Functions can be Minimized via Graph Cuts? IEEE Trans. on PAMI 26(2), 147–159 (2004)
8. Kumar, S., Hebert, M.: Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field. In: Proc. of CVPR (2003)
9. Kumar, S., Hebert, M.: Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In: Proc. of ICCV (2003)
10. Kumar, S., Hebert, M.: Discriminative Fields for Modeling Spatial Dependencies in Natural Images. In: Proc. of NIPS (2004)
11. Platt, J.C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: Advances in Large Margin Classifiers, pp. 61–74 (1999)
12. Porikli, F.: Integral Histogram: a Fast Way to Extract Histograms in Cartesian Spaces. In: Proc. of CVPR (2005)
13. Šochman, J., Matas, J.: WaldBoost – Learning for Time Constrained Sequential Detection. In: Proc. of CVPR (2005)
14. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: Proc. of ICCV (October 2005)
15. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. of CVPR (December 2001)
16. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast Asymmetric Learning for Cascade Face Detection. IEEE Trans. on PAMI 30(3), 369–382 (2008)
17. Xiao, R., Zhu, L., Zhang, H.: Boosting Chain Learning for Object Detection. In: Proc. of ICCV (October 2003)
18. Zhang, J., Zhou, S.K., McMillan, L., Comaniciu, D.: Joint Real-time Object Detection and Pose Estimation Using Probabilistic Boosting Network. In: Proc. of CVPR (2007)

A Novel System for Robust Text Location and Recognition of Book Covers

Zhiyuan Zhang[1], Kaiyue Qi[1], Kai Chen[1], Chenxuan Li[1], Jianbo Chen[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
zzy_3@sjtu.edu.cn, kchen@sjtu.edu.cn

Abstract. Text location and recognition is a vital and fundamental problem of processing images. In this paper we propose a novel system for text location and recognition focused on book covers. Our work consists of two main parts, learning-based text location and adaptive binarization guided recognition. First we extract three types of robust features from the training data provided on ICDAR2005 and utilize Ada-boost to combine these features into a powerful classifier for text regions detection and location. Second we apply the proposed adaptive binarization to process the located regions for recognition. Compared with previous works, our algorithm is robust in size, font and color of text, and insensitive for languages. In experiments, our system proved to have attractive performance.

1 Introduction

Applications of image text location and recognition are useful and capital in many situations, including identifying products by reading text (such as books, movie posters, CD covers), indexing images in digital databases for retrieval and so on. However, Optical Character Recognizer (OCR) can only recognize texts with simple background. Text areas in images must be located and binarized in advance. Usually, systems for text location and recognition include three parts, text location, binarization and OCR.

Approaches for text location can be divided into two categories based on feature utilized: region-based and texture-based methods. [1]

Region-based methods consider that text areas have distinct intensity or color compared with its background. Jain and Yu [2] used a set of geometry features to classify generated connected-components. Hasan and Karam [3] utilized morphological operations to extract text regions. S. Messelodi and C.M. Modena [4] proposed a cover oriented method which can estimate the skew of text lines. However, these methods need a lot of experimental thresholds and parameters, so they cannot support robust location.

Texture-based methods assume that text areas have distinct texture apart from background. These methods often involve techniques like Gabor filter, wavelet, FFT, spatial variance and so on. For example, Wu [5] used a multi-scale texture segmentation scheme which includes 9 second-order Gaussian derivatives. As this kind of methods is very sensitive to text font and size, it is hard to select a threshold

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 608–617, 2010.

manually. Accordingly, many approaches manipulate machine learning algorithm to do this. Kim [6] used SVM to learn texture feature of text. They all achieved a much better result by using machine learning algorithm.

Binarization methods can also be classified into two types: global methods and local adaptive methods. The simplest binarization technique is to use a global fixed threshold. These techniques are generally based on histogram analysis. The most famous global binarization is Otsu's method [7]. However, global method only works well for images with well separated foreground and background intensities. In practice, most images do not meet this condition. On the contrary, local methods use a dynamic threshold to binary every pixel. For example, Niblack's method [8] computed threshold for each pixel according to mean and standard deviation value in a local window. But the method caused a lot of noise in background area. Sauvola's method [9], an improved version of Niblack's method, minimized the background noise. However, this method was based on the hypothesis that the gray values of the text are close to 0. Chang's method [10] made an adaptive decision between thresholds calculated at different spatial scales. This method works well at most situations. But it can't solve the color opposite problem.

This paper puts forward a novel system of image text location and recognition focused on book covers. See Fig. 1. The text location part combines both region-based feature and texture-based feature. Ada-boost is used to select and combine these features into a powerful classifier and locate text regions in images by classifying subregions of images as text or non-text. The followed binarization part, which improves Karsar's method [16] greatly, not only minimizes the noise, but also adapts to different font, color and language.

In our experiments, two data sets are used. The training data set is downloaded from ICDAR2005 text location competition with labeled samples. The testing data set is constructed by book covers, which includes scanned covers and camera-shot covers. See Fig. 2.

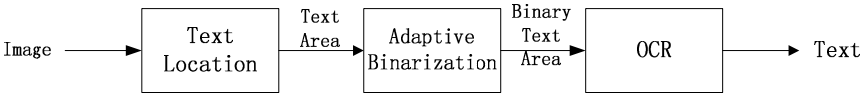


Fig. 1. Structure of proposed system

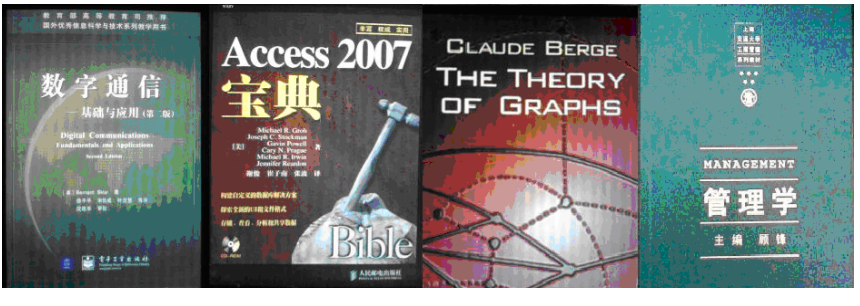


Fig. 2. Example images of testing data sets. As these images are taken by phone cameras, their qualities are rather low

The rest of the paper is organized as followed. In section 2, three sets of robust features will be introduced. In section 3, the location method will be described. Adaptive binarization is mentioned in section 4. Experiment results are shown in section 5. Section 6 is the conclusion of the paper.

2 Robust Feature Extraction

In fact, image text location is a binary classification problem, classifying windows or areas as text or non-text. In classification problems, features are of vital importance for accuracy of results. So we specified three sets of robust features based on statistical analysis of datasets.

Histogram Features. Histogram features are based on this observation. In an image which contains text, there must be a large number of horizontal, vertical or diagonal lines. And in those don't, number of these lines is usually small. [12]

So, we can define patterns like these, which represent vertical, horizontal, vertical, horizontal, and diagonal lines respectively. See Fig. 3.

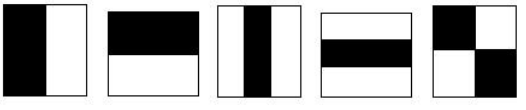


Fig. 3. Five patterns

-1	+2	-1
-1	+2	-1
-1	+2	-1
-1	+2	-1
-1	+2	-1

Fig. 4. Feature used in test

A histogram feature contains these parameters:

First: Type. Which type of pattern this feature belongs to. Second: Height and width. Third: Value interval. An interval includes an upper threshold and a lower threshold. These thresholds refer to pixel value threshold of filtered image.

Given an input image or an image window {xij} and a predefined feature fk, a feature value is generated after the following steps:

1. Filter image {xij} with pattern defined in fk and get another image {yij}.
2. Get percentage of pixels of {yij} whose values are between certain intervals. These intervals are also defined in fk. See Fig. 5.

Edge Features. The second class of features is based on the following observation. It's certain that long edge lines mostly appear in a small window containing text. So, based on edge detection, using CANNY, we can get edge image from original image. And this type of feature is to count the number of long edges.

Block-based Features. The third feature set is from Chen [13], but has been improved. Tested on training dataset, the average response of x and y derivatives have obvious patterns shown in Fig. 6. The x derivatives tend to be large in central of text area while y derivatives are large at the top and bottom. And the variance of x derivatives is large while y derivatives are small.

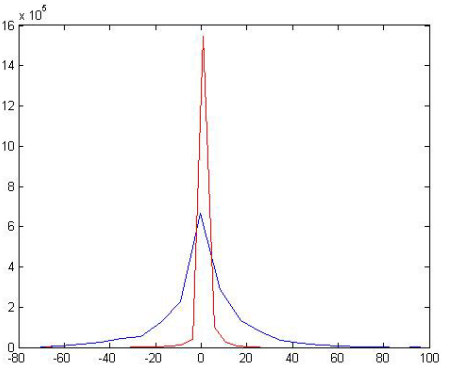


Fig. 5. Test results. For non-text images, their responses to this feature are mostly around zero and have low entropy. For text images, their responses are much more scattered.

Different from Chen [13], who designed symmetric block patterns suitable for English words, we designed block patterns regardless of number of words in a window. See Fig. 7. Without this limitation, training samples are much easier to get.

In summary, there are: (1)5200 first class features based on histogram, (2)30 second class features based on long edge counting, and (3)384 third class features based on x and y derivatives. Therefore, a large amount of features have been specified and Ada-boost can be utilized to generate a powerful classifier.

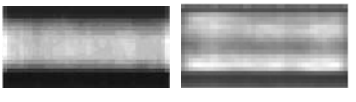


Fig. 6. Response of x and y derivatives. X derivatives are small at top and bottom while y derivatives are large at the top and bottom.

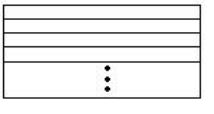


Fig. 7. Block pattern. In this block pattern, the height of each sub-line varies from 1/6 to 1/2 of window height.

3 Text Location

Given a training set of positive and negative samples and a set of features, any machine learning algorithm could be used to train a strong classifier. But Ada-boost's performance on detecting faces [14] has proved that it's the most effective algorithm for detecting target object in images.

First, Ada-boost learning requires a set of training data labeled manually as text or non-text. We use the data set provided on ICDAR2005 text locating competition. From this data set, we divide each text window into several overlapping samples with fixed aspect ratio of 2:1 and get 2522 positive samples. The negative samples are extracted randomly from the non-text area of this data set and we get 8846 negative samples. See Fig. 8.

Second, we transform features described in previous section into weak classifiers. A weak classifier $w_i(x)$ usually consists of a feature $f_i(x)$, a threshold t_i and a parity p_i which indicates the direction of the inequality sign:

$$w_i(x) = \begin{cases} 1, p_i f_i(x) < t_i \\ 0, otherwise \end{cases} . \quad (1)$$

Here x is a 40×20 pixel sub-window of an image. We selected these weak classifiers with standard Ada-boost learning procedure combined with an attentional cascade [14]. A cascade could drop those sub-windows which are apparently non-text in early stages. This brings a significant boost in processing speed compared with standard Ada-boost algorithm [15]. Our algorithm had 3 cascade layers. The first layer consists of only 1 block-based weak classifier. The second and third layer includes histogram and edge based classifiers, which are much more computational exhausted. By applying generated powerful classifier on image, we can get its text area.

Fig. 8. Positive samples used in Ada-boost training. These samples include various type of text which appear in book covers.

4 Adaptive Binarization

Binarization is a necessary part in recognition system. When using OCR module to process a book cover directly, lots of messy codes would appear. Our binary module adaptively exports the foreground text as black and background as white.

4.1 Kasar's Method [16]

The Kasar's Method can be described briefly as following.

Firstly, canny edge detection is performed individually on each channel of the image and an edge map E can be obtained: $E = E_R \mid E_G \mid E_B$. Where E_R E_G E_B separately represents the edges detected in the RGB channel. Here E stands for all possible boundaries detected by canny operator.

Secondly, an eight-connected component labeling follows. In Kasar's method, each component obtained is called an edge-box (EB).

Thirdly, as every EB may have its own inner and outer boundary, so an EB may enclose several EBs inside. For English character, study shows that every EB has no more than two EBs inside. The rule is that, if N_{int} is less than 3, reject EB_{int} and accept EB_{out}. If N_{int} is more than 3, reject EB_{out} and accept EB_{int}. Where EB_{int} means EB that completely lies inside another EB and N_{int} means the number of EB_{int}.

Fourthly, get a threshold by judging the foreground and background intensities.

$$F_{EB} = \frac{1}{N_E} \sum_{(x,y)\in E} I(x,y) \cdot \tag{2}$$

Where E represents the pixels in the edge, N_E means the number and I(x,y) means the grayscale intensity. F_{EB} represents the foreground, which is the average grayscale of all the pixels in the boundary. B_{EB} which stands for background is computed by the grayscale intensities of twelve points, which are located around the four corners of the bounding box. See Fig. 10. $BW_{EB}(x,y)$ represents for the finally processed binary value. According to these two arguments, we can binary every pixel according to criteria as below:

$$\begin{aligned} &\text{If } F_{EB} < B_{EB}\text{: } BW_{EB}(x,y) = \begin{cases} 1, I(x,y) \geq F_{EB} \\ 0, I(x,y) < F_{EB} \end{cases} \cdot \\ &\text{If } F_{EB} > B_{EB}\text{: } BW_{EB}(x,y) = \begin{cases} 0, I(x,y) \geq F_{EB} \\ 1, I(x,y) < F_{EB} \end{cases} \cdot \end{aligned} \tag{3}$$

4.2 Evaluation of the Original Algorithm

In actual applications, Kasar's Method has some drawbacks. First of all, edge detection in the passage utilized canny operator. As a fact of the canny detection, Gaussian filter must go first. The variance of the associated is fixed to 1. Its disadvantage is that one solid parameter can not adapt to every situation.

Second, the former algorithm meets some problem when dealing with the diagonally aligned text. These EBs detected may interfere with other adjacent EBs. Though using median value can solve the problem to some extent, there are still lots of color-opposite phenomena during our experiments.

Last and most important, the algorithm can't deal with Chinese character. For English text, it's true that every EB has no more than two inner EBs. But if the algorithm is used directly to process the Chinese characters, many texts will be wiped off.

4.3 Improved Kasar's Method

To solve these problems, we contrapuntally proposed three new techniques. They are multi-scale filtering, precise background computation, and EB selecting amendment. Particularly, the new EB selecting method immediately widens the scope of the algorithm, making it adaptive to characters in various languages.

Multi-scale filtering. The purpose of using in multi-scale is to find the detail characteristics in different scales. Based on the particularity of canny detection, the edges detected at different Gaussian variance vary a lot. Single scale filtering has two main shortcomings. One is that it's hard to decide the appropriate parameter to supply the edge information. The other one is that edges detected by a single scale are fragmented with each other.

We have used the thresholds 0.2 and 0.3 for the hysteresis threshold step of canny edge detection. The variance of the Gaussian function steps from 0.2 to 1.8 by 0.2 per

operation. The final edge map E is obtained by combining all the edge images at different scales. $E = E_1 \mid E_2 \mid \ldots \mid E_n$. E_i stands for the edge image detected by the ith scale. Edge map E obtained in this way can effectively solve the two problems mentioned above.

Improved Background Intensity Computation. The original Kasar's method uses twelve points' intensities, which are located around the four corners of the bounding box, to compute the background intensity. The method works well when the text lies horizontally. If not, the adjacent bounding boxes would have some overlaps with each other, especially the points around two of four corners, when the text lies diagonally. The intensity of the point picked up around the corner would not represent the real background intensity. As displayed in Fig. 9. Therefore, besides these points, we expand the scope of the points choosing. As it is observed in Fig. 10, these twenty four points not only increase the sample set, but also reduce the interfere rate. In the end, the local background intensity can be estimated by considering the median intensity of the 24 pixels.

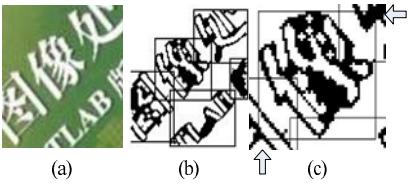


Fig. 9. (a) The original image which text lies diagonally (b) Output of EB-box (c) Partial enlargement. The c drawing shows that the intensities of the points which are located around the two corners cannot represent the background intensity.

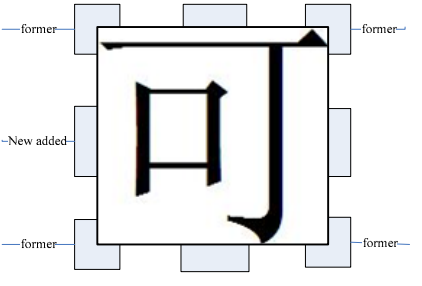


Fig. 10. Former ones are the twelve pixels around the four corners. New added twelve pixels are the located in the middle of each line of the bounding box.

EB layer is a label for every EB. Refer to Fig. 11. An EB which doesn't completely enclose other EBs is called layer-1 EB. While an EB completely encloses layer-1 EB is called layer-2 EB. Similarly, in an image, there may be layer-3, layer-4 EB… And for the convenience of the study, those EBs whose layer number is more than 2 are all called layer-3 EBs.

Characters always obey the rule that there are only layer-1 and layer-2 EBs. Layer 3 EB often appears at the complex frame and the texture region which are all non-text regions. So, we can reject all these layer-3 EBs. After the process, there are only layer-2 and layer-1 EBs left. And, layer-1 EBs which are located completely inside a layer-2 EB are also rejected. The purpose of the step is that the outer EB can already represent the text information. There is no need to keep too many details inside. The final EBs left are all used to binary and give a very good effect.

At last, compute foreground and background intensity for every EB left. Use the criteria to binary every EB. Though a Chinese character may have some bounding boxes overlapped with each other, it's not a big problem anyway. Merging these binary results by using or operation can show all the text information.

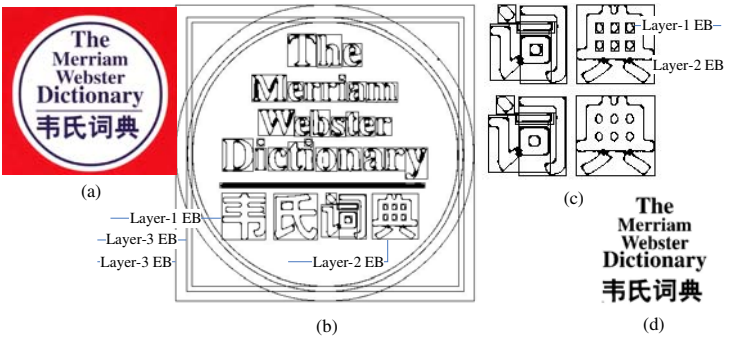


Fig. 11. (a) Original image (b) output of EB-box (c) the upper one is B drawing's partial enlargement; the below one is the EB left (d) the binary result of original image. As can be seen from the figure, the circular frame and the line are all wiped out in the final image.

Fig. 12. Results of text location on test images. You can notice that the texts are bounded with a relative big rectangle for robustness of location. But these big rectangles would be reduced at adaptive binarization process.

Fig. 13. (a) Original image (b) Niblack's Method (c) Sauvola's Method (d) Chang's Method (e) Kasar's Method (f) Proposed Method

Fig. 14. Examples of binarization results using the proposed method. Every character is inverted to black and background to white.

5 Experiments

In text location stage, we applied trained powerful classifier on sub-windows of input images. The window size scales from 40×20 to 303×152, with scaling factor 1.5 and shifting factor 0.2. We get a binary image indicating text areas. By finding connected components of this image, text area could be retained. See Fig. 12.

While in binarization stage, In Fig. 13, we can see great differences between binarization methods clearly. Niblack's method causes lots of noise in the background. Sauvola's method only solves the problem to some extent. Chang's method changes the background to black which may create lots of messy code in the OCR module. Kasar's method can't deal with every Chinese character. On the other hand, our method can deal with characters of any size, color and language.

We have performed the experiments on more than 100 pictures. All the pictures are all camera shooting ones. About 1200 characters are detected. Statistically, our location rate is more than 90% and the recognition rate is more than 80%. Other four methods' recognition rates are all lower than the proposed one's. While the recognition rate by sauvola's method, is better than the Niblack's and Chang's. Kasar's method is more suitable for the English characters. Its recognition rate decrease when pictures are integrated with the Chinese characters. So, our result is able to withstand the test. Some results of binarization are shown in Fig. 14.

6 Conclusion

This paper presents a novel system of image text location and recognition in book covers. This system is robust for various text font, size, color, and image skew, rotation, view point change, and noise. Compared with other researchers' work, such as Chen[4], this system can locate and recognize text with different language rather than only English.

Our future work is to find out other useful statistical feature of text, which could be very helpful in locating text. Since the system presented only acts well for book covers, we shall also improve its performance in other circumstances, such as natural scenes, movie poster and so on.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No.60773093) and the Key Program for Basic Research of Shanghai (Grant No.08JC1411800), the Ministry of Education, and Intel joint research foundation (Grant No.MOE-INTEL-08-11).

References

[1] Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and videos: A survey. Pattern Recognition 37, 977–997 (2004)
[2] Jain, A.K., Yu, B.: Automatic Text Location in Images and Video Frames. Pattern Recognition 31(12), 2055–2076 (1998)
[3] Hasan, Y.M.Y., Karam, L.J.: Morphological text extraction from images. IEEE Trans. Image Process. 9(11), 1978–1983 (2000)
[4] Messelodi, S., Modena, C.M.: Automatic identification and skew estimation of text lines in real scene images. Pattern Recognition 32, 791–810 (1999)
[5] Wu, V., Manmatha, R., Riseman, E.M.: TextFinder: an automatic system to detect and recognize text in images. IEEE Trans. Pattern Anal. Mach. Intell. 21(11), 1224–1229 (1999)
[6] Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support vector machine-based text detection in digital video. Pattern Recognition 34(2), 527–529 (2001)
[7] Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Systems Man Cybernetics 9(1), 62–66 (1979)
[8] Niblack, W.: An introduction to digital image processing, pp. 115–116. Prentice Hall, Englewood Cliffs (1986)
[9] Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recognition 33, 225–236 (2000)
[10] Chang, F.: Retrieving information from document images: problems and solutions. Int. J. Doc. Anal. Recognition 4, 46–55 (2001)
[11] Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic Regression: a statistical view of boosting. The Annals of Statistics 28(2), 337–374 (2000)
[12] Li, C., Ding, X.G., Wu, Y.S.: An Algorithm for Text Location in Images Based on Histogram Features and Ada-boost. Journal of Image and Graphics (2006)
[13] Chen, X.R., Yuille, A.L.: Detecting and reading text in natural scene. In: Proceeding of CVPR 2004 (2004)
[14] Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR 2001 (2001)
[15] Viola, P., Jones, M.: Fast and Robust Classification using Asymmetric Ada-boost and a detector cascade. In: Advances in Neural Information Processing Systems (2002)
[16] Kasar, T.: Font and Background Color Independent Text Binarization. In: ICDAR 2005 (2005)

A Multi-scale Bilateral Structure Tensor Based Corner Detector

Lin Zhang, Lei Zhang*, and David Zhang

Biometrics Research Center, Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
Tel.: 852-27667355
{cslinzhang,cslzhang,csdzhang}@comp.polyu.edu.hk

Abstract. In this paper, a novel multi-scale nonlinear structure tensor based corner detection algorithm is proposed to improve effectively the classical Harris corner detector. By considering both the spatial and gradient distances of neighboring pixels, a nonlinear bilateral structure tensor is constructed to examine the image local pattern. It can be seen that the linear structure tensor used in the original Harris corner detector is a special case of the proposed bilateral one by considering only the spatial distance. Moreover, a multi-scale filtering scheme is developed to tell the trivial structures from true corners based on their different characteristics in multiple scales. The comparison between the proposed approach and four representative and state-of-the-art corner detectors shows that our method has much better performance in terms of both detection rate and localization accuracy.

Keywords: Harris, corner detector, bilateral structure tensor.

1 Introduction

Corner detection is a critical task in various machine vision and image processing systems because corners play an important role in describing object unique features for recognition and identification. Applications that rely on corners include motion tracking, object recognition, 3D object modeling, and stereo matching, etc.

Considerable research has been carried out on corner detection. One of the earliest successful corner detectors can be Harris corner detector [1]. Harris et al. [1] calculated the first-order derivatives of the image along horizontal and vertical directions, with which a 2×2 structure tensor was formed. The corner detection was accomplished by analyzing the eigenvalues of the structure tensor at each pixel. However, computing derivatives is sensitive to noise, and the Harris corner detector has poor localization performance because it needs to smooth the derivatives for noise reduction. Thus, several methods [2-3] have been proposed to improve its performance.

Apart from Harris corner detector and its variants, many other corner detectors have also been proposed by researchers. Kitchen and Rosenfeld [4] proposed a cornerness measure based on the change of gradient direction along an edge contour

* Corresponding author.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 618–627, 2010.

multiplied by the local gradient magnitude. Smith and Brady [5] proposed the SUSAN scheme. In SUSAN, a circular mask is taken around the examined pixel and this pixel is considered as the nucleus of the mask. Then "USAN" (Univalue Segment Assimilating Nucleus) is defined as an area of the mask which has the similar brightness as the nucleus. Smith et al. [5] assumed that the USAN would reach a minimum when the nucleus lies on a corner point. Wang and Brady [6] proposed a corner detection algorithm based on the measurement of surface curvature. In [7] and [8], Mokhtarian et al. proposed two CSS (Curvature Scale Space) based corner detectors. In these two algorithms, edge contours are first extracted and then corners are detected as the positions with high curvatures on edge contours. Zheng et al.'s [9] cornerness measure was simply the gradient module of the image gradient direction.

This paper presents a novel effective evolution of the classical Harris corner detector. In the original Harris corner detector, an isotropic Gaussian kernel is used to smooth each of the four elements in the 2×2 structure tensor over a local window before calculating the eigenvalues. Such a smoothing operation will have two disadvantages. First, some weak corners will be smoothed out. Second, the localization accuracy is much degraded. Inspired by the success of bilateral filters [10] in image denoising, which consider both the spatial and the intensity similarities in averaging neighboring pixels for noise removal, in this paper we construct a nonlinear bilateral structure tensor and use it to detect corner points.

The basic idea of the proposed method lies in that both the spatial and gradient distances should be involved in smoothing the structure tensor elements. The neighboring pixels that have shorter spatial and gradient distances to the given one should have higher weights in the averaging. In this way, a nonlinear structure tensor, which is adaptive to image local structures, could be constructed and hence the image local pattern could be better distinguished. It can be seen that the classical Harris corner detector is a special case of the proposed method by exploiting only the spatial distance in the structure tensor smoothing. However, the proposed nonlinear structure tensor has much higher sensitivity to corner-like fine structures than the linear structure tensor. Therefore, it may respond strongly to some trivial feature points in the image. In order to get rid of the possible false corners detected at fine image scales, we propose a multi-scale filtering scheme based on the different characteristics of true corners and trivial structures in multiple scales.

The rest of the paper is organized as follows. Section 2 briefly reviews the Harris corner detector. Section 3 presents the new corner detector in detail. Experimental results are presented in section 4 and the conclusion is made in section 5.

2 Harris Corner Detector

Harris corner detector [1] has been very widely used in machine vision applications. Consider a 2D gray-scale image I. Denote by $W \in I$ an image patch centered on (x_0, y_0). The sum of square differences between W and a shifted window $W_{(\Delta x, \Delta y)}$ is calculated as

$$S = \sum_{(x_i, y_i) \in W} (I(x_i, y_i) - I(x_i - \Delta x, y_i - \Delta y))^2 \tag{1}$$

By approximating the shifted patch using a Taylor expansion truncated to the first order terms, we have:

$$S = [\Delta x, \Delta y] A \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2}$$

where $A = \begin{bmatrix} \sum_{(x_i, y_i) \in W} (\nabla_i^h)^2 & \sum_{(x_i, y_i) \in W} \nabla_i^h \nabla_i^v \\ \sum_{(x_i, y_i) \in W} \nabla_i^v \nabla_i^h & \sum_{(x_i, y_i) \in W} (\nabla_i^v)^2 \end{bmatrix}$ and ∇_i^h and ∇_i^v represent the first order partial derivatives of image I along horizontal and vertical directions at pixel (x_i, y_i).

In practice matrix A is computed by averaging the tensor product $\nabla I \cdot \nabla I^{\mathrm{T}}$ (∇I denotes the gradient image of I) over the window W with a weighting function K_ρ, i.e.

$$A_\rho = \begin{bmatrix} \sum_{(x_i, y_i) \in W} K_\rho(i)(\nabla_i^h)^2 & \sum_{(x_i, y_i) \in W} K_\rho(i) \nabla_i^h \nabla_i^v \\ \sum_{(x_i, y_i) \in W} K_\rho(i) \nabla_i^v \nabla_i^h & \sum_{(x_i, y_i) \in W} K_\rho(i)(\nabla_i^v)^2 \end{bmatrix} \tag{3}$$

Usually K_ρ is set as a Gaussian function $K_\rho(i) = \frac{1}{\sqrt{2\pi}\rho} \exp\left(-\frac{d_i^2}{2\rho^2}\right)$, where $d_i^2 = (x_i - x_0)^2 + (y_i - y_0)^2$ and ρ is the standard deviation of the Gaussian kernel.

A_ρ is symmetric and positive semi-definite. Its main modes of variation correspond to the partial derivatives in orthogonal directions and they are reflected by the eigenvalues λ_1 and λ_2 of A_ρ. The two eigenvalues can form a rotation-invariant description of the local pattern. Under the situation of corner detection, three distinct cases are considered. 1) Both the eigenvalues are small. This means that the local area is flat around the examined pixel. 2) One eigenvalue is large and the other one is small. The local neighborhood is ridge-shaped. 3) Both the eigenvalues are rather large. This indicates that a small shift in any direction can cause significant change of the image at the examined pixel. Thus a corner is detected at this pixel.

Harris suggested that the exact eigenvalue computation can be avoided by calculating the response function

$$R(A_\rho) = det(A_\rho) - k \cdot trace^2(A_\rho) \tag{4}$$

where $det(A_\rho)$ is the determinant of A_ρ, $trace(A_\rho)$ is the trace of A_ρ, and k is a tunable parameter.

3 Bilateral Structure Tensor Based Corner Detection

This section presents the proposed multi-scale nonlinear bilateral structure tensor based corner detector in detail. Our algorithm differs from the original Harris corner detector mainly in two aspects. First, a nonlinear structure tensor is constructed to substitute for the linear one used in the Harris corner detector; second, a multi-scale filtering scheme is proposed to filter out the false and trivial corners detected at small scales.

3.1 Construction of the Bilateral Structure Tensor

The structure tensor for a gray level image I is a 2×2 symmetric matrix that contains in each element the orientation and intensity information in a local area. Denote by ∇I the gradient image of I. The initial matrix field can be computed as the tensor product $J_0 = \nabla I \cdot \nabla I^{\mathrm{T}}$. To incorporate the neighboring structural information into the given position, an averaging kernel could be used to smooth each element of J_0. Usually a Gaussian kernel K_ρ with standard deviation ρ is employed for this purpose:

$$J_\rho = K_\rho * J_0 \tag{5}$$

where symbol "*" means convolution. Since convolution is a linear operator, the structure tensor J_ρ is referred to as linear structure tensor [11]. It is a symmetric, positive semi-definite matrix. Comparing Eq. (3) with Eq. (5), we see that the matrix A_ρ in Harris corner detector is actually the linear structure tensor J_ρ at pixel (x_0, y_0).

In Harris corner detector [1], the "cornerness" of a pixel (x,y) is totally determined by its local structure tensor $J_\rho(x,y)$. However, the smoothing kernel K_ρ has two problems. First, the isotropic smoothing operation will smooth some weak corner features out so that the detection capability is decreased. Second, the localization accuracy of detected corner points will be reduced, which is a well-known problem of the Harris corner detector. Intuitively, if the local structure tensor can better preserve the local structural information at (x,y), the cornerness measured from it should be more reliable and accurate.

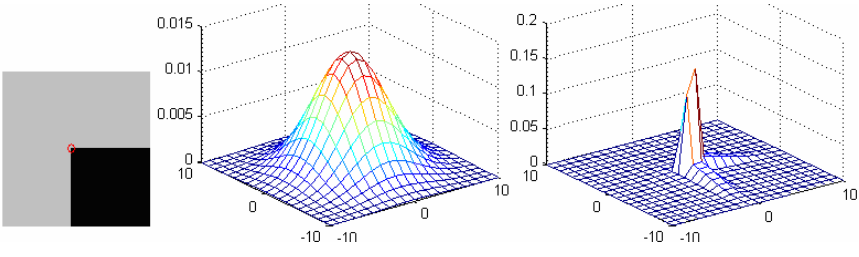


Fig. 1. Weight distributions in a neighborhood of a corner pixel. (a) An artificial image with an ideal corner (red circle); (b) weights distribution by using the Gaussian kernel K_ρ; (c) weights distribution by using the proposed bilateral weighting function $N_{\rho,\sigma}$.

As an early denoising technique, Gaussian smoothing is simple but it will over-blur the image details. The Gaussian weighting kernel only uses the notation of spatial location in the weights assignment. The greater the spatial distance from a neighboring pixel to the central pixel, the smaller the averaging weight will be assigned. The intensity similarity between the pixels is not exploited in Gaussian smoothing. In [10], the bilateral filter was proposed, which employs both the spatial and intensity similarities between pixels in averaging weight design. It has been shown that bilateral filtering could significantly improve the edge structure preservation while removing noise [10].

Inspired by the success of bilateral filters in image denoising, in this paper we construct a bilateral structure tensor for better corner detection performance. There are two basic factors in the formation of a local pattern: the relative positions between neighboring pixels and the intensity variations between them. Therefore, in the smoothing of J_0, we should consider both the spatial distance and the gradient distance in the averaging weight assignment. In the original Harris corner detector, only the spatial distance is considered by applying a Gaussian smoothing kernel K_ρ to $\nabla I \nabla I^{\mathrm{T}}$. In this paper, we will also involve the gradient distance in the smoothing of $\nabla I \nabla I^{\mathrm{T}}$.

Here, the gradient distance from the position (x_i, y_i) to the central position (x_0, y_0) is defined as:

$$d_i^g = \sqrt{\left(\nabla_i^h - \nabla_0^h\right)^2 + \left(\nabla_i^v - \nabla_0^v\right)^2} \tag{6}$$

The spatial distance from (x_i, y_i) to (x_0, y_0) is the same as in the original Harris corner detector:

$$d_i^s = \sqrt{\left(x_i - x_0\right)^2 + \left(y_i - y_0\right)^2} \tag{7}$$

By considering both the spatial and gradient distances into the assignment of averaging weight, we define the following bilateral weighting function for each pixel $(x_i, y_i) \in W$:

$$N_{\rho,\sigma}(i) = \frac{1}{C_{\rho,\sigma}} \exp\left(-\frac{(d_i^s)^2}{2\rho^2}\right) \cdot \exp\left(-\frac{(d_i^g)^2}{2\sigma^2}\right) \tag{8}$$

where ρ and σ are the parameters to control the decaying speeds over spatial and gradient distances, and

$$C_{\rho,\sigma} = \sum_W \exp\left(-\frac{(d_i^s)^2}{2\rho^2}\right) \cdot \exp\left(-\frac{(d_i^g)^2}{2\sigma^2}\right) \tag{9}$$

is the normalization factor.

Fig. 1 shows an example to illustrate the weight distributions by using the Gaussian kernel K_ρ and the proposed function $N_{\rho,\sigma}$. Fig. 1-a is an artificial image with an ideal corner in the center, which is marked by a red circle. The size of local window W for smoothing is set as 21×21. Figs. 1-b and 1-c illustrate the weight distributions for the pixels within W by using the Gaussian kernel K_ρ and the proposed bilateral weighting function $N_{\rho,\sigma}$, respectively. It is clearly seen that K_ρ is isotropic and is independent of the image local structure, while $N_{\rho,\sigma}$ is anisotropic and is adaptive to the image local pattern. In this example, the edge pixels have higher weights than the non-edge pixels because they are more similar to the examined corner pixel in terms of gradient. Meanwhile, for the pixels lying on the same edge, the ones near to the corner pixel have higher weights than the others because they have shorter spatial distances to the corner point.

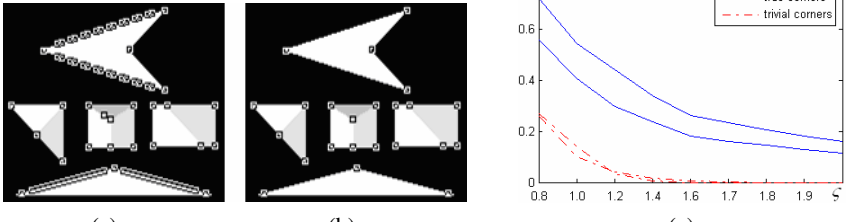
With the nonlinear bilateral weighting function $N_{\rho,\sigma}$, the nonlinear bilateral structure tensor is defined as:

$$A_{\rho,\sigma} = \begin{bmatrix} \sum_{(x_i,y_i)\in W} N_{\rho,\sigma}(i)(\nabla_i^h)^2 & \sum_{(x_i,y_i)\in W} N_{\rho,\sigma}\nabla_i^h\nabla_i^v \\ \sum_{(x_i,y_i)\in W} N_{\rho,\sigma}\nabla_i^v\nabla_i^h & \sum_{(x_i,y_i)\in W} N_{\rho,\sigma}(\nabla_i^v)^2 \end{bmatrix} \tag{10}$$

The corner detection is based on the analysis of the above defined nonlinear bilateral structure tensor $A_{\rho,\sigma}$. Similar to the original Harris corner detector, we calculate the response function $R(A_{\rho,\sigma}) = det(A_{\rho,\sigma}) - k \cdot trace^2(A_{\rho,\sigma})$ to determine if a corner point exists in the current position.

3.2 Multi-scale Filtering

Because the proposed nonlinear bilateral structure tensor $A_{\rho,\sigma}$ incorporates the local gradient information in the structure tensor construction, it could achieve much higher true detection and localization accuracies than the linear structure tensor used in the original Harris corner detector. However, it is also sensitive to some trivial structures. Due to digitization in the square grid, in discrete images often the ramp edges will show corner-like trivial structures in a fine scale. Those trivial structures will be enhanced by the proposed nonlinear structure tensor $A_{\rho,\sigma}$ and they may be falsely detected as true corners. Fig. 2-a shows an example. We can see many false detections along the ramp edge by using $A_{\rho,\sigma}$. To solve this problem, we propose a multi-scale filtering scheme to filter out those small scale trivial structures.



(a) (b) (c)

Fig. 2. (a) Corner candidates before multi-scale filtering; (b) final corner detection result after multi-scale filtering; (c) Relative cornerness ratio (RCR) curves of two true corners (blue curves) and two trivial corners (red curves);

Suppose that we have obtained some corner candidates with the proposed nonlinear structure tensor. We will distinguish the trivial corner-like structures from the true corners by their different cornerness characteristics at multiple image scales. The images at different scales can be obtained by smoothing the original image I with a series of Gaussian kernels K_ς with different standard deviations ς. By increasing the values of ς, a fine to coarse scale space can be formed. The underlying principle for our multi-scale filtering scheme is as follows. If a trivial structure is detected as a corner at a fine scale, the cornerness of this point should decrease rapidly with the increase of scale ς because it will be smoothed out by K_ς. On the contrary, if a true

corner point is detected at a fine scale, the cornerness of it will decrease smoothly with the increase of ς because it will appear in a wide range of scales.

Denote by R_0 the cornerness of a corner candidate measured by Eq. (4) at the finest scale 0, and by R_ς its cornerness measured at scale ς. We define the relative cornerness ratio (RCR) as

$$r_\varsigma = R_\varsigma / R_0 \tag{11}$$

Fig. 2-c shows the RCR curves of two true corner points (blue curves) and two trivial corner points (red curves). From this figure we can clearly see that the RCR of false corners will decay much faster than the RCR of true corners.

Based on the different behaviors of true corners and trivial corners in the scale space, we are able to tell them to remove false and trivial corners. Suppose we use L scales in the multi-scale filtering. A candidate corner point is recognized as a true corner point if

$$\sum_{l=1}^{L} r_\varsigma(l) \geq T \tag{12}$$

where T is a threshold. Fig. 2-b shows the final corner detection result after multi-scale filtering (L=3). We see that many false corners detected in Fig. 2-a are removed in Fig. 2-b without affecting the true corners.

4 Experimental Results

Experiments were performed on 3 standard test images. The ground truth corner points were manually labeled. For the *artificial* test image (refer to Fig. 3-a_3), it is easy to identify these reference corners and the locations of corners can be accurately located. However, for real test images *blocks* (refer to Fig. 3-a_1) and *house* (refer to Fig. 3-a_2), it is nearly impossible to give absolutely accurate corner locations. Therefore, we only computed the localization accuracy for the artificial test image, while computed the detection accuracy for all the three test images. The code can be found at http://www.comp.polyu.edu.hk/~cslzhang/MBST_CD/.

The proposed corner detector was compared with four representative algorithms: Harris [1], SUSAN [5], Enhanced CSS [8] and the nonlinear structure tensor based method [11]. In [11], the authors proposed two different ways to construct a nonlinear structure tensor: one is by isotropic diffusion and the other is by anisotropic diffusion. In this paper, we compared the result given by the isotropic diffusion because it achieves similar result to that by anisotropic diffusion but has much less computational cost. We refer to it as INLST for short in the following. For the four methods used in comparison, we tuned the parameters so that the best corner detection results were obtained.

The proposed method has several parameters. The parameter ρ (referring to Eq. (8)) is adaptively determined based on the size of window W, i.e. the spatial range, according to the 3-sigma principle of Gaussian function. Similarly, the parameter σ (referring to Eq. (8)) is fixed by the range of d_i^g (referring to Eq. (6)), i.e. the gradient range, according to the 3-sigma principle. In the multi-scale filtering, we empirically

find that it is insensitive to the scale selection and usually 3~5 scales are enough. Thus, in our experiments we used 3 scales and the same threshold for all the test images: ς_1=0.6, ς_2=1.0, ς_3=1.4 and T=1.0 (referring to Eq. (11) and Eq. (12)). Finally, the parameters left to set are the window size W and coefficient k (referring to Eq. (4)). In this paper they were set as follows: for the artificial test image, W=5×5 and k=0.04; for the *blocks* test image, W=21×21 and k=0.02; and for the *house* test image, W=13×13 and k=0.02.

Denote by C_{ref} the set of reference (ground truth) corners and by C_{det} the set of detected corners by a particular detector. Denote by d_{max} the maximal acceptable distance between the reference corner and the detected corner. In this paper, we set d_{max}=4(pixels). For a pair of corner points $C_i \in C_{ref}$ and $C_j \in C_{det}$, if the distance $d_{i,j}$ between C_i and C_j is minimum for $\forall i, j$ and $d_{i,j} \le d_{max}$, then C_j is labeled as a "correct" detection of C_i. Otherwise, C_i is labeled as "missed". The corners labeled as "missed" in C_{ref} are considered as true corners but not detected, and the remaining corners in C_{det} are considered to be the "false" detections. The localization error is the average of all the distances $d_{i,j}$ for the corners detected correctly.

The experimental results are summarized in Table 1 and Fig. 3. The classical Harris corner detector performs moderately well with respect to the true detection rate. However, it loses some weak corners, which can be clearly seen in Fig. 3-b_1 and Fig. 3-b_2. SUSAN performs very well on the *artificial* test image whereas its performance on the natural images is not so good. For the enhanced CSS method, its detection rate and localization accuracy heavily depend on the output of the contour extraction. If an actual connected contour is broken up by the contour extraction step, more false corner points would be detected since the algorithm regards the line endings as corner points. Table 1 shows that INLST has better localization performance than Harris, SUSAN and Enhanced CSS. However, it is sensitive to noise and trivial structures and has much false detection. The proposed method performs the best in terms of both detection rate and localization accuracy.

Table 1. Evaluation results on test images

Method	artificial				blocks			house		
	correct	missed	false	location error	correct	missed	false	correct	missed	false
Harris	78	0	0	1.1347	52	8	3	57	20	46
SUSAN	78	0	0	1.0982	48	12	15	62	15	27
Enhanced CSS	76	2	3	1.6992	55	5	8	50	27	11
INLST	78	1	52	0.6235	55	5	5	57	20	12
Proposed	**78**	**0**	**0**	**0.4187**	**57**	**3**	**0**	**64**	**13**	**4**

Among the tested detectors, SUSAN is the fastest one. The proposed method is slower than the other ones because it needs to compute the weight function $N_{\rho,\sigma}$ for each pixel. In the future we will investigate how to reduce the computational cost without sacrificing much the accuracy.

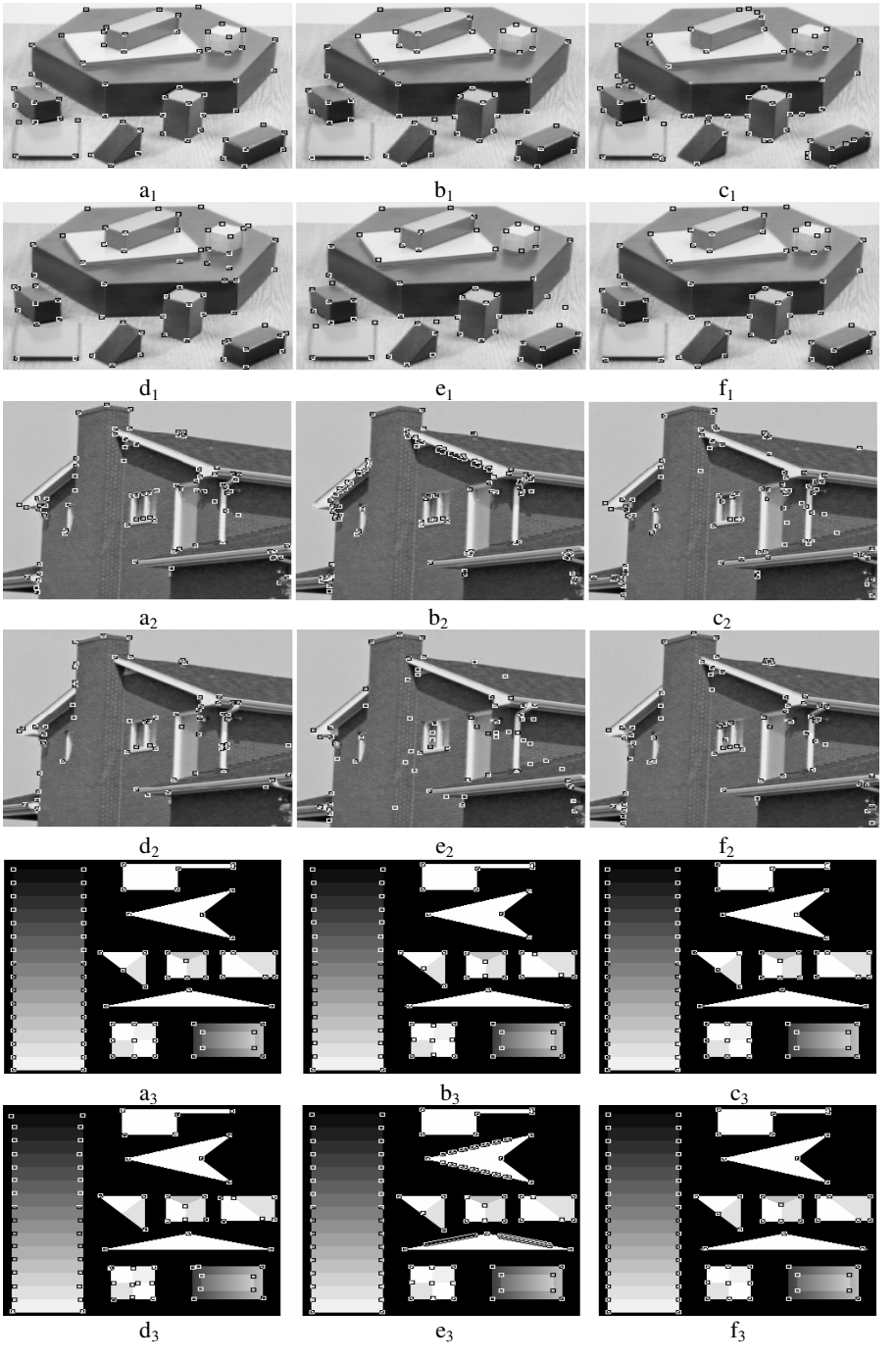


Fig. 3. Experimental results on 3 test images. (a_i) ground truth; (b_i) Harris; (c_i) SUSAN; (d_i) enhanced CSS; (e_i) INLST; (f_i) the proposed method.

5 Conclusions

In this paper, we proposed a corner detection algorithm by constructing a nonlinear bilateral structure tensor, which exploits both the spatial distances and the gradient distances from the neighboring pixels to the central pixel to be examined. Moreover, in order to remove the trivial corner-like structures, a multi-scale filtering scheme was developed. Experimental results on some standard test images show the effectiveness of the proposed corner detector in terms of both detection rate and localization accuracy. However, it should be noted that the computational cost of the proposed algorithm is higher than the other detectors. It can be a choice when the speed of corner detection is not a great concern but the accuracy is of the most importance.

Acknowledgement

The work is supported by the Hong Kong RGC General Research Fund (PolyU 5351/08E), Edward Sai Kim Hotung Fund (5-ZH52), and the HK-PolyU Internal Competitive Research Grant (G-YH54).

References

1. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, pp. 147–151 (1988)
2. Pei, S., Ding, J.: Improved Harris' Algorithm for Corner and Edge Detections. In: Proc. ICIP, pp. 57–60 (2007)
3. Liu, Y., Hou, M., Rao, X., Zhang, Y.: A Steady Corner Detection of Gray Level Images Based on Improved Harris Algorithm. In: Proc. Int. Conf. on Networking, Sensing and Control, pp. 708–713 (2008)
4. Kitchen, L., Rosenfeld, A.: Gray-level corner detection. Pattern Recognition Letters 1(2), 95–102 (1982)
5. Smith, S.M., Brady, J.M.: SUSAN—A New Approach to Low Level Image Processing. International Journal of Computer Vision 23(1), 45–78 (1997)
6. Wang, H., Brady, J.M.: Real-time corner detection algorithm for motion estimation. Image and Vision Computing 13(9), 695–703 (1995)
7. Mokhtarian, F., Suomela, R.: Robust image corner detection through curvature scale space. IEEE Trans. PAMI 20(12), 1376–1381 (1998)
8. Mokhtarian, F., Mohanna, F.: Enhancing the curvature scale space corner detector. In: Proc. Scandinavian Conf. on Image Analysis, pp. 145–152 (2001)
9. Zheng, Z., Wang, H.E.K., Teoh, E.K.: Analysis of gray level corner detection. Pattern Recognition Letters 20(2), 149–162 (1999)
10. Tomasi, C., Manduchi, R.: Bilateral Filtering for Gray and Color Images. In: Proc. ICCV, pp. 839–846 (1998)
11. Brox, T., Weickert, J., Burgeth, B., Mrazek, P.: Nonlinear structure tensors. Image and Vision Computing 24(1), 41–55 (2006)

Pedestrian Recognition Using Second-Order HOG Feature

Hui Cao, Koichiro Yamaguchi, Takashi Naito, and Yoshiki Ninomiya

Road Environment Recognition Lab
Toyota Central R&D LABS.,INC.
Nagakute, 480-1192 Aichi, Japan
{caohui,yamaguchi,naito,ninomiya}@mosk.tytlabs.co.jp

Abstract. Histogram of Oriented Gradients (HOG) is a well-known feature for pedestrian recognition which describes object appearance as local histograms of gradient orientation. However, it is incapable of describing higher-order properties of object appearance. In this paper we present a second-order HOG feature which attempts to capture second-order properties of object appearance by estimating the pairwise relationships among spatially neighbor components of HOG feature. In our preliminary experiments, we found that using harmonic-mean or min function to measure pairwise relationship gives satisfactory results. We demonstrate that the proposed second-order HOG feature can significantly improve the HOG feature on several pedestrian datasets, and it is also competitive to other second-order features including GLAC and CoHOG.

1 Introduction

Pedestrian detection receives a growing interest in the field of computer vision. While it is effortless to implement a real-time pedestrian detection system since Viola and Jones's seminal work on Haar wavelet features and cascading Adaboost [1]. However, using Haar-like features will inevitably result in a significant number of wrong detections. In order to eliminate these wrong detections, the latter stage of classification needs to adopt stronger image features which are more discriminative for recognition, such as HOG feature.

Histograms of Oriented Gradients (HOG) proposed by Dalal and Triggs is one of the most popular features for pedestrian detection [2] which describes object appearance by a combination of local histograms of gradient orientation. The HOG feature enjoys many advantages such as its invariance to photometric transformation and small individual body movement. However, because HOG feature only extracts first-order histogram statistics of local object appearance, images with non-pedestrian content may be wrongly classified when they generate pedestrian-like first-order statistics. In this paper we propose a second-order HOG feature which describes object appearance by a combination of local pairwise relationships among components of HOG feature. We demonstrate that the proposed second-order HOG feature can significantly outperform the original HOG feature on several pedestrian datasets.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 628–634, 2010.

This paper is organized as follows: section 2 reviews some related works on pedestrian detection; section 3 presents the detail of the proposed second-order HOG feature; experimental results are shown in section 4 and finally, section 5 concludes the paper with some final remarks.

2 Related Works

There is an extensive literature on human detection, here we just mention some of the representative works and also some works related to our method.

Viola and Jones proposed a rapid object detection framework based on Haar wavelet features and an Adaboost classification cascade [1]. Later, Edgelet [3], HOG [4], Covariance [5] and mixing of above features [6,7] have been integrated into the Viola-Jones' framework, obtaining better detection rates. Besides, incorporating auxiliary information like spatial context and depth cues has recently shown benefits in reducing false detections [8,9].

Since human body is a combination of parts, many part-based methods have been presented. Papageorgiou and Poggio learned a polynomial SVM classifier using Haar wavelets per part and integrated classification scores using a second-stage SVM [10]. Shashua et al. learned multiple linear classifiers using HOG-like gradient features per part and integrated classification scores using a second-stage Adaboost [11]. More recently, Felzenszwalb et al. proposed a multi-scale and deformable SVM-based part model using HOG features [12].

Gradient Local Auto-Correlations (GLAC) [13] and Co-occurrence Histograms of Oriented Gradients (CoHOG) [14] are two methods related to our work which represent object appearance by a combination of local second-order histograms of gradient orientation. They divide image into small regions, and for each region they calculate joint histograms for various patterns of neighbor pixels, like up-down,left-right, etc.. Our method presented in this paper differs from them as ours extracts second-order statistics at region level rather than at pixel level done by them.

3 Second-Order Histograms of Oriented Gradients

In HOG feature, each component represents the accumulated gradient magnitude in each orientation within a region. Instead of using these components directly, we attempt to estimate pairwise relationships among them.

Figure 1 shows the comparison between the proposed second-order HOG and HOG features. The process of computing second-order HOG feature is as follow: (1) divide the input image into dense grids, called cells; (2) create histogram of gradient orientations for each cell; (3) compute pairwise relationships among histogram components in a (non-)overlapping block[1], (a block means a larger region containing spatially-connected cells); (4) apply block normalization to pairwise relationship vector. The combination of all pairwise relationship vectors

[1] In case of overlapping block, each cell contributes more than once to the final feature.

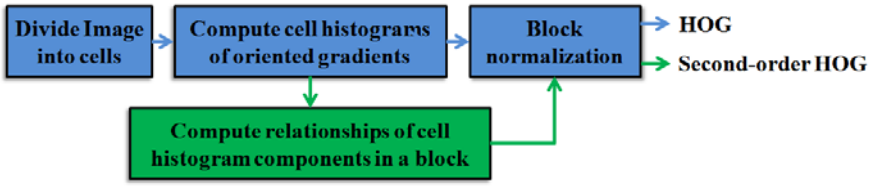


Fig. 1. Comparison between second-order HOG feature and original HOG feature

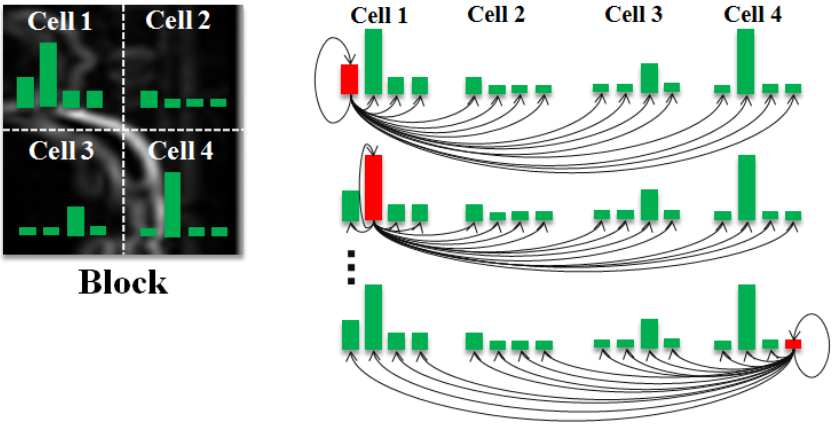


Fig. 2. Pairwise relationships among histogram components

across the image represents the second-order HOG feature. In the following two subsections, we explain the two key steps, step2 and step3, in detail.

3.1 Histogram of Gradient Orientations

Each pixel within the cell casts a weighted vote for corresponding histogram bin based on its gradient magnitude and orientation. The orientation bins of each cell histogram are evenly spaced over $0° \sim 180°$ or $0° \sim 360°$ depending on whether the signs of gradient are informative or not. The gradient is computed by 1-D derivatives($[-1; 1]$, $[-1; 0; 1]$) or 2-D derivatives like sobel masks.

3.2 Pairwise Relationships among Histogram Components

In the previous step, the corresponding histogram of gradient orientations for each cell is calculated. For a given cell i, we denote its cell histogram by $\mathbf{h}_i = [h_{i1}, \ldots, h_{in}]$, in which n indicates the number of orientation bins. Then, the spatially-connected m cells are grouped into a block, so the block histogram is the combination of relevant cell histograms, $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m]$. We enumerate all pairwise combinations of components in block histogram $\mathbf{H}$ (see a illustration

Table 1. The specification of experiment datasets

Dataset Type	Daimler-Chrysler	Near-infrared (day)	Near-infrared (night)
Training data	4800×3 pedestrian	5000 pedestrian	5000 pedestrian
	5000×3 non-pedestrian	5000 non-pedestrian	5000 non-pedestrian
Test data	4800×2 pedestrian	5000 pedestrian	5000 non-pedestrian
	5000×2 non-pedestrian	5000 non-pedestrian	5000 non-pedestrian
Image size	18×36 pixels	30×60 pixels	30×60 pixels

in Fig.2). The pairwise relationships between these combinations are measured in terms of a predefined function $f(\cdot,\cdot)$. This results in a pairwise relationship vector like

$$[f(h_{11},h_{11}),\ldots,f(h_{11},h_{mn}),f(h_{12},h_{12}),\ldots,f(h_{12},h_{mn}),\ldots,f(h_{mn},h_{mn})] \quad (1)$$

The function $f(\cdot,\cdot)$ to measure the degree of relationship could be any metric function. Here, three functions including (1) harmonic mean ($\frac{2h_1h_2}{h_1+h_2}$), (2) min ($min(h_1,h_2)$) and (3) product (h_1h_2) are considered.

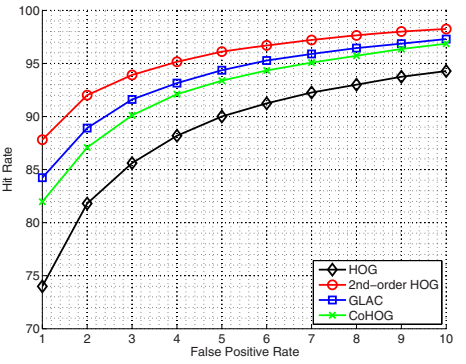
4 Experimental Results

We conduct experiments on three datasets: Daimler-Chrysler dataset [15] and two near-infrared datasets. The Daimler-Chrysler dataset has been widely used as a benchmark dataset in performance comparison. The two near-infrared datasets are collected by us in the daytime and at nighttime respectively. The specification of the three datasets are listed in Fig. 1.

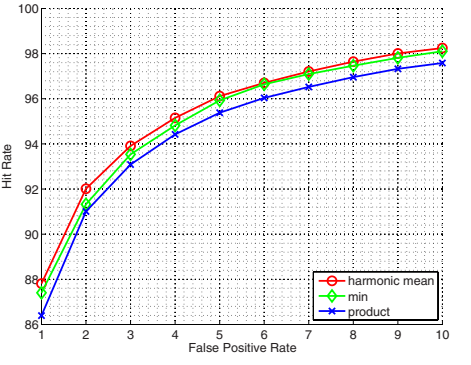
In the experiments, apart from the proposed second-order HOG feature, we also implement and evaluated three image features: HOG, GLAC and CoHOG. The latter two features are two implementations of pixel-based joint gradient histograms that are recently reported state-of-the-art performances on Daimler-Chrysler dataset. In order to make fair comparisons for four classes of features, we use the same parameter setting: (1) roberts gradient filter; (2) 8 orientation bins in $0 \sim 360$ degrees; (3) image division in 3×6 non-overlapping for CoHOG[2] and 5×11 overlapping blocks for others; (4) block-wise normalization based on L2-norm scheme; (5) each block consisting of 2×2 cells for HOG and second-order HOG. We used linear SVM as classification tool for training and testing of these features.

The Daimler-Chrysler Benchmark dataset is divided into five disjoint sets, three for training and two for testing. Each set consists of $4,800$ pedestrian examples and $5,000$ non-pedestrian examples. Two out of the three training sets are used for training and the third training set is used for cross validation. Thus we obtain three SVM classifiers. Applying three classifiers to two test sets yields

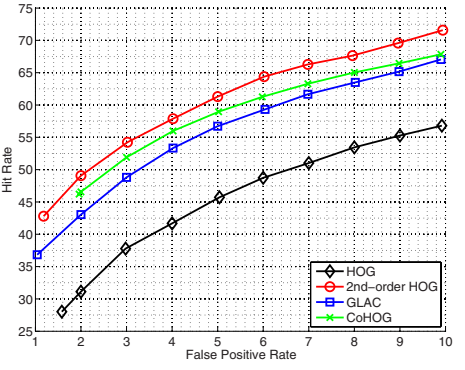
[2] In case of overlapping block, the feature dimensions of CoHOG are too huge to be handled by our computer.



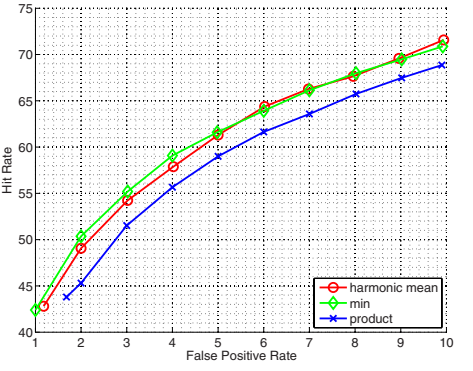
(a) Feature comparison (DC)



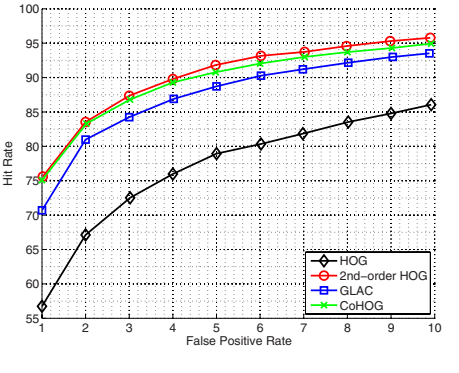
(b) Function comparison (DC)



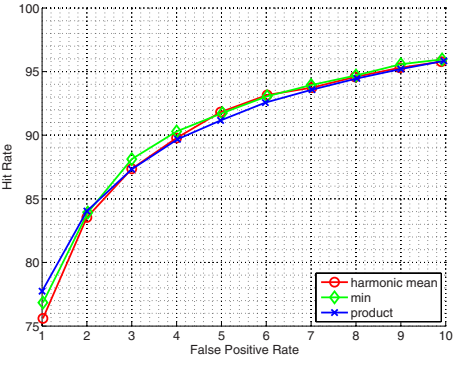
(c) Feature comparison (Daytime)



(d) Function comparison (Daytime)



(e) Feature comparison (Nighttime)



(f) Function comparison (Nighttime)

Fig. 3. The ROC curves on the left columns show the performances achieved by the second-order HOG and other features. The ROC curves on the right columns show performance change over different functions used for measuring pairwise relationship. From top to down, Daimler-Chrysler dataset (DC), daytime near-infrared dataset (Daytime) and nighttime near-infrared dataset (Nighttime).

Table 2. The comparison of features dimensions

Feature Type	HOG	Second-order HOG	GLAC	CoHOG
Dimensions	1,760	29,040	14,080	34,704

6 ROC curves from which we calculate the mean ROC curve. The mean ROC curves by four classes of features are plotted in the Fig.3(a). Our second-order HOG (using harmonic mean function) significantly improve HOG about $6 \sim 15\%$ at different false positive rates, outperforming GLAC and CoHOG. To the best of our knowledge, the performance achieved by second-order HOG is also the best results that have ever been published to date.

The daytime/nighttime near-infrared dataset consists of one training set and one test set. Each set consists of 5000 pedestrian and 5000 non-pedestrian examples. Linear SVM Classifier is learned on daytime/nighttime training set by cross validation. Applying the learned SVM classifier to daytime/nighttime test set generates the classification results and the ROC curves are summarized in Fig.3(c) (for daytime dataset) and Fig.3(e) (for nighttime dataset). The classification performance achieved on the nighttime dataset is similar to that obtained on the Daimler-Chrysler dataset. However, the classification performance achieved on the daytime dataset is not that desirable, probably due to a lot of indistinguishable samples being contained in the test set. For both cases of the daytime and the nighttime near-infrared datasets, the proposed second-order HOG significantly improves HOG and is slightly better than GLAC and CoHOG.

We also investigate the performance difference of different functions for estimating pairwise relationship in the second-order HOG. The ROC curves on three datasets are plotted on Fig.3(b), Fig.3(d) and Fig.3(f), respectively. Overall, the harmonic-mean and min functions achieve similar performances, better than the one by product function. The performance gains by harmonic-mean and min functions are probably due to that: (1) they are nonlinear functions and (2) they are appropriate for histogram comparison.

Table 2 lists the dimensions of four classes of features. The second-order HOG is about 16 times larger than HOG, about twice larger than GLAC and smaller than CoHOG. The region-based second-order HOG feature is supposed to be computationally efficient than other GLAC and CoHOG because they need to compute joint histograms for different arrangements of neighbor pixels.

5 Conclusion

The higher-order properties of object appearance are important cues for object detection. In this paper we present a second-order HOG feature which extends the well-known HOG feature to be able to describe second-order properties of local object appearance. The second-order HOG feature is implemented by estimating the pairwise relationships among spatially neighbor components in HOG feature. Experimental results on several pedestrian datasets show that the new

feature significantly improves the HOG feature and slightly outperforms other second-order features.

Our current work is to do detailed investigation of the performance with regard to different metrics. In addition, we are planing to develop a second-order SIFT.

References

1. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR (2001)
2. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, pp. 886–893 (2005)
3. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In: ICCV (2005)
4. Zhu, Q., Avidan, S., Yeh, M., Cheng, K.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: CVPR (2006)
5. Tuzel, O., Porikli, F., Meer, P.: Human Detection via Classification on Riemannian Manifolds. In: CVPR (2007)
6. Geronimo, D., Lopez, A., Ponsa, D., Sappa, A.D.: Haar Wavelets and Edge Orientation Histograms for On-Board Pedestrian Detection. In: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis (2007)
7. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: CVPR (2008)
8. Heitz, G., Koller, D.: Learning Spatial Context: Using Stuff to Find Things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
9. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded Classification Models: Combining Models for Holistic Scene Understanding. In: NIPS (2008)
10. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38(1), 15–33 (2000)
11. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In: IEEE Intelligent Vehicles Symposium (2004)
12. Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model, In: CVPR (2008)
13. Kobayashi, T., Otsu, N.: Image Feature Extraction using Gradient Local Auto-Correlations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 346–358. Springer, Heidelberg (2008)
14. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection. In: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology, pp. 37–47 (2009)
15. Munder, S., Gavrila, D.M.: An Experimental Study on Pedestrian Classification. IEEE T. PAMI 28(11), 1863–1868 (2006)

Fabric Defect Detection and Classification Using Gabor Filters and Gaussian Mixture Model

Yu Zhang, Zhaoyang Lu, and Jing Li

State Key Laboratory of Integrated Service Networks,
Xidian University, Xi'an, 710071, China
roykimbly@hotmail.com

Abstract. This work investigates the problem of automatic and robust fabric defect detection and classification which are more essential and important in assuring the fabric quality. Two characteristics of this work are: first, a new scheme combining Gabor filters and Gaussian mixture model (GMM) is proposed for fabric defect detection and classification. In detection, the foreground mask and texture features are extracted using Gabor filters. In classification, a GMM based classifier is trained and assigns each foreground pixel to known classes. The second characteristic of this work is the test data is actually collected from Qinfeng textile factory, China, including nine different fabric defects with more than 1000 samples. All the evaluation of our method is based on these actual fabric images and the experimental results show the proposed algorithm achieved satisfied performance.

1 Introduction

Automatic visual inspection (AVI) [1] plays an important role in modern textile industry. It is reported that a sophisticated worker at most could detect 60-70 percent of all defects at a very low speed. Thus, it is desirable for a system to achieve an accuracy up to 90 percent with much faster speed. In AVI, fabric texture analysis has attracted much attention.

A great many methods of texture analysis have been investigated during the past several decades. Generally, defect detection techniques have been classified into statistical, spectral and model-based categories [1, 4, 5]. Among them, spectral methods are the most widely adopted in application. There are Fourier transform, wavelets and Gabor filters [4]. Studies on human vision supported the multi-resolution analysis, which motivated the prevalence of Gabor filters. In addition, Gabor filters have tunable angular and axial frequency bandwidths, tunable center frequencies, and can achieve optimal joint resolution in spatial and frequency domain.

In texture classification, according to Randen [6], Bayes classifiers, nearest neighbor and neural networks are commonly used as effective classifiers for texture classification. When texture image is decomposed through wavelets, Gaussian distribution is found in sub-band statistics, which are related to the structure of texture. Besides, the linear combination of several Gaussian

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 635–644, 2010.

distributions can approximate any distribution very well. Consequently, we prefer Gaussian Mixture model (GMM) [3] to model the defect texture.

In ref. [3], wavelet packet frame and Gaussian mixture model are used for texture classification. Features of texture are extracted by wavelet packet frame and then sent into the GMM based classifier for classification. However, this feature extraction method is not suitable for fabric image with local defect. In contrast, Gabor filters have an excellent ability to locate the local fabric defect precisely. Meanwhile, it can describe the defect in any orientation and scale from spatial or frequency domain, which is more flexible than the wavelet packet frame. In this paper, we propose a new method for fabric defect detection and classification using Gabor filters and Gaussian mixture models. To our best knowledge, it is the first time that GMM is applied in the classification of fabric defects.

The rest of the paper is organized as follows. Section 2 introduces the fabric defect detection using Gabor filters. Section 3 introduces the defect classification based on GMM. Section 4 gives the experiment and analysis. Section 5 presents the conclusion.

2 Defect Detection Using Gabor Filters

2.1 Improved Gabor Filters

A 2-D Gabor function (1) is a complex exponential modulated by a Gaussian function [4], which can form a complete but non-orthogonal basis set.

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp[-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})]\exp(j2\pi fx) \tag{1}$$

where σ_x and σ_y are the envelops along the x and y axes, f is the central frequency of this Gabor function, $j = \sqrt{-1}$.

In most cases, a reasonable design is to select a set of circularly symmetric Gabor filters, i.e., $\sigma_x = \sigma_y = \sigma$. Here we adopt a relation between spatial envelop and central frequency $f = 1/(\alpha * \sigma)$, where α is a constant that controls the ratio between central frequency and bandwidth.

A bank of Gabor filters $g_i(x', y'), i = 1, 2, \ldots, S \times L$, can be obtained by dilation (scale) and rotation (orientation) of $g(x, y)$, where (x', y') are the coordinates rotated by θ. To make the algorithm more robust against brightness, a discrete Gabor filter $g_i(x', y')$ is turned to zero DC [7] using the following formula:

$$\overline{g}_i(x', y') = g_i(x', y') - mean[g_i(x', y')] \tag{2}$$

For a given image $I(x, y)$, its magnitude response is its convolution with $\overline{g}_i(x', y')$.

2.2 Optimal Output Selection

Fabric defect is of strong orientation and different scale so it can react remarkably to certain appropriate Gabor filters. Here, we use an easy and effective method

proposed by Kumar [4] to choose an optimal output to best describe the defect. (1) The initial image $I(x, y)$ is divided into K non-overlapping square regions with the same size; (2) To calculate the mean value of each square region in the ith output and acquire its maximum value D^i_{max} and the minimum value D^i_{min}; (3) A cost function as equation (3) is applied to evaluating the output of each filter:

$$J(i) = (\frac{D^i_{max} - D^i_{min}}{D^i_{min}}) \tag{3}$$

Finally, the channel which gives the maximum output of the cost function is chosen as the optimal one and denoted as $I_{opt}(x, y)$.

2.3 Binarization

The optimal result which is a gray image is to be converted into a binary image. Following ref. [5], we use a Gaussian lowpass filter which is related to the central frequency of the Gabor filter to reduce speckle-like noise. The threshold limits can be determined by filtering a defect-free (reference) image with the optimal Gabor filter and a smoothing filter to obtain a new imageB_{ref}:

$$\begin{cases} \lambda_{max} = \max\limits_{x,y \in W} B_{ref}(x, y) \\ \lambda_{min} = \min\limits_{x,y \in W} B_{ref}(x, y) \end{cases} \tag{4}$$

where W is a window centered in the image to avoid the distortion of convolution caused in the edge. The binary process of $I_{opt}(x, y)$can be conducted as the following step:

$$mask(x, y) = \begin{cases} 1, I_{opt}(x, y) > \lambda_{max} or I_{opt}(x, y) < \lambda_{min} \\ 0, \qquad\qquad \lambda_{min} \leq I_{opt}(x, y) \leq \lambda_{max} \end{cases} \tag{5}$$

In this binary mask, the region of defect is white and the rest is black, which give us the accurate information of defect's location.

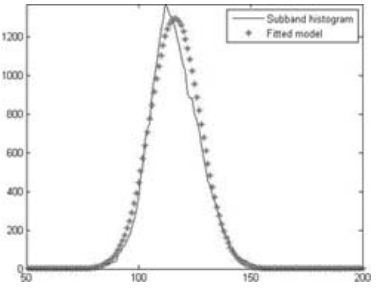


Fig. 1. Approximation of gray-level histogram of fabric defect in one Gabor channel

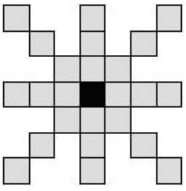


Fig. 2. The configuration of the neighborhood for each pixel [2]

3 Defect Classification Using GMM

In the detection phase, we choose an optimal output from the Gabor filtered results and convert it into a binary mask to acquire its precise location. In classification, we use GMM to describe the gray-level distribution of each pixel in the defect region. Generally, GMM can approximate any distribution using weighed sum of three or more Gaussian distributions. The histogram of the fabric defect in sub-channel shows good conformity by the approximation of GMM in fig.1. In this phase, we first extract a set of features and these features will provide unique characteristic to each defect pattern. Second, these features are used for the training of corresponding GMM of each defect, which is the composition of the classifier for the classification of unknown defects.

3.1 Feature Extraction

In this paper, we employ 5 features: mean, variance, 1-norm, 2-norm and entropy to characterize the neighborhood information of texture pattern.

$$F^i_{mean} = \frac{1}{25} \sum_{x,y \in neighbor} I_i(x,y) \tag{6}$$

$$F^i_{var} = \frac{1}{25} \sum_{x,y \in neighbor} (I_i(x,y) - F^i_{mean})^2 \tag{7}$$

$$F^i_{norm1} = \frac{1}{25} \sum_{x,y \in neighbor} |I_i(x,y)| \tag{8}$$

$$F^i_{norm2} = \frac{1}{25} \sum_{x,y \in neighbor} |I_i(x,y)|^2 \tag{9}$$

$$F^i_{entropy} = \frac{1}{25} \sum_{x,y \in neighbor} |I_i(x,y)|^2 log|I_i(x,y)|^2 \tag{10}$$

Generally, the larger the size of the neighborhood, the more precise the texture feature is. The configuration of an appropriate neighborhood is suggested using 7×7 pixels with 25 effective elements [2] shown in fig.2. The above features are

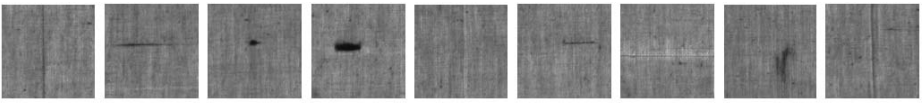


Fig. 3. Samples of nine defects

computed from the Gabor filtered outputs of all channels and the dimension is significant. Thus, PCA is required to reduce the dimension of the feature vector for the input and the training of GMM based classifier. It's efficient to keep the important information such as the dominant orientation of fabric defects. More important is that the transformed feature vector is uncorrelated with each other. We only keep those principle components which contribute dominantly to the total variance.

3.2 GMM Based Classification

In this modeling method, it is supposed that any distribution p can be described through the linear combination of several Gaussian distributions with varying mean μ and covariance matrix Σ .

$$p(x) = \sum_{i=1}^{N} \beta_i g_i(x; \mu_i, \Sigma_i) \tag{11}$$

$$\sum_{i=1}^{N} \beta_i = 1 \tag{12}$$

where N is the number of models, β is the weight to each Gaussian distribution g.

The detailed description for training GMM can be found in ref. [3], the most important of which is the expectation maximum (EM) criterion. Here, we care more about a way proposed by A.Bouman [8] to determine the precise number of the GMM needed for a distribution. First, an initial number of clusters is set, which is big enough to cover the possible situations. Then we follow the EM criterion to train the Gaussian mixture model. Based on the minimum description length (MDL) estimator suggested by Rissanen, some clusters will be combined and an appropriate number can be finally determined.

To each kind of fabric defect, a set of sample images are used to train its corresponding Gaussian mixture model $GMM_i, i = 1, 2, \ldots, L$, where L is the number of the defect class. All these GMMs compose of a classifier. The probability of newly extracted feature vector is estimated to each GMM. The outputs of all feature vectors are summed up to each model. In the decision unit, the model which produces the maximum sum is determined as the class which the defect belongs to.

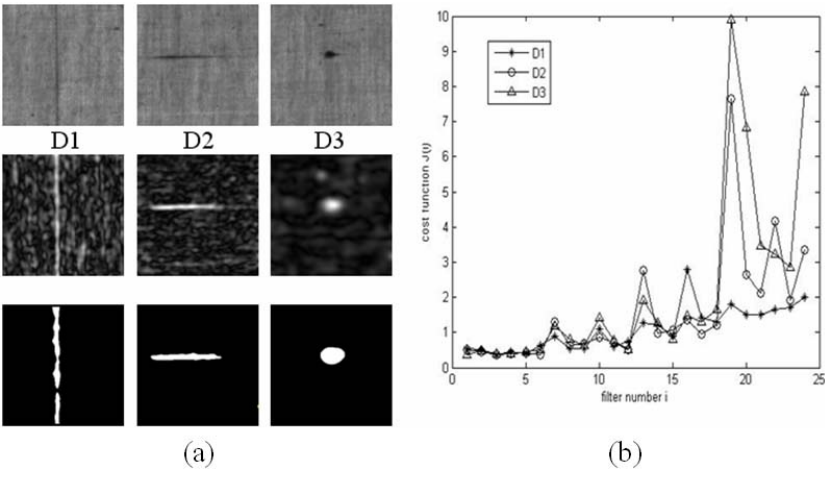


Fig. 4. (a) Fabric sample with defect D1, D2 and D3 in the first row respectively; corresponding optimal result using Gabor filters in the second row; binary masks in the third row; (b) the cost function corresponding to defects in (a)

4 Experiment and Analysis

Our experimental images are acquired using industrial monitors from Qinfeng textile factory, China. These images are in the size of 256×256 pixels with 8 bit resolution. Our database includes 300 normal images and 720 defective images. Normal images are those which contain no defect or defect smaller in size than the requirement. Nine different kinds of defects are D1 (coarse pick), D2 (bamboo), D3 (cotton), D4 (junk), D5 (end missing), D6 (weft missing), D7 (double weft), D8 (slime spots) and D9 (wrinkle) respectively shown in fig.3.

4.1 Defect Detection

Each of the filters in the Gabor filter bank is implemented as an 11×11 convolution mask for each of its real and imaginary components. The central frequencies are chosen as $1/2, 1/4, 1/8$ and $1/16$. The number of orientation is 6, which is equally divided between 0 and 180. $\alpha = 2$. Each of these images is divided into 64 non-overlapping regions. Segmentation results of 3 classes D1, D2 and D3 are shown in fig.4 (a) and the cost function of each image in (b).

In fig.4 (b), the numbering scheme for each Gabor filter on the horizontal axis is $(p-1) \times 6 + q, p = 1, 2, 3, 4; q = 1, 2, \ldots, 6$. Samples D1, D2 and D3 achieve their peaks in the number 16, 19 and 19 filter respectively. Taken D1 as an example, the order of the maximum output is 16, which means this optimal filter is in the 3rd scale and 4th orientation. More segmentation results for defect D4-D9 are shown in fig.5.

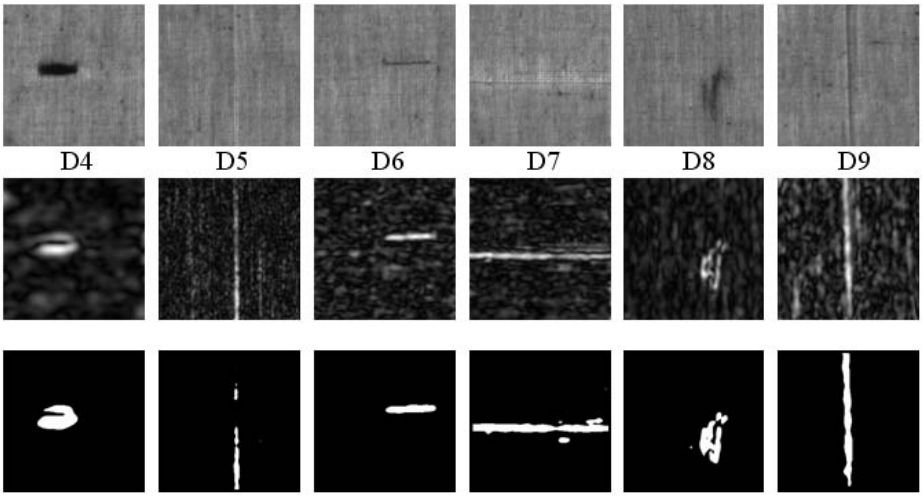


Fig. 5. More samples and their segmentation results, from left to right, they are D4-D9

In this part, several factors may influence the performance of segmentation. The frequency range of the Gabor filter bank is required to cover defects as much as possible. We always have the dilemma between the accuracy and computation. One measure is to choose several frequencies of interest from some prior knowledge. Another issue is the block size for optimal Gabor output selection. Its size must be able to cover defect and defect-free regions separately, so that the discrimination between the two cases can be maximized.

4.2 Defect Classification

We train the Gaussian mixture based classifier using half of the images in each class and the other half are for test. To each defect image, we think that the pixels in the mask belong to some unknown defect. In the outputs from 24 Gabor filter channels, we extract the feature of unknown defect pixel-wisely. We experiment on three combinations of features (6-10): (1) mean, variance; (2) mean, variance, entropy; (3) all features: mean, variance, entropy, 1-norm, 2-norm. So that the dimension of each feature vector in the three situations are 48, 72 and 120 respectively. PCA is applied to them and 98 percent is set to determine the variance of required principal components to the whole variance.

To evaluate the performance of the classifier, first we apply 3 Gaussian mixture models to describe the distribution of each class. The k-means algorithm is used to determine the initial positions of the cluster centers. The diagonal covariance matrixes are initialized by means of sample covariance. In our experiment, we build a classifier with 9 Gaussian mixture models that represent different classes of defects. Defects of each class are tested by this classifier separately and the rate of correct classification is shown in fig.6. (a). Second, we use Bouman' method as a comparison to evaluate the accurate number of the GMM each class needs.

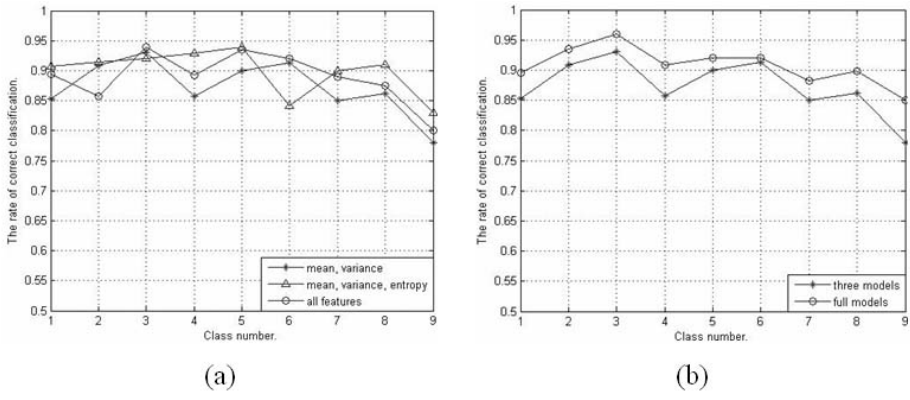


Fig. 6. (a) Comparison of performance using 3 different combinations of features. (b) Comparison of performance using 3 models and full models.

The initial number of GMM is set 20. The comparison of 3 models and Bouman' method (full models) using features mean and variation is shown in fig.6 (b).

From fig.6 (a), we could see the rate doesn't improve much due to the increase of features. With 2 features mean and variance to characterize the defect texture, the rate has reached an average level more than 85 percent. Three features can increase the rate about 5 percent more than 2 features to some defects, because they can give more information than the latter case. While more features will surely bring more computation load. To certain defect, more important is to choose an appropriate feature rather than to increase the unrelated features' number. Fig.6 (b) reveals us some information about the number of GMM. Under the case with the same features, the classifier with accurate model number estimation performs better than the classifier with predetermined model number. It is precise to use Bouman's method to evaluate an appropriate model number but time consuming, which is its greatest weakness.

4.3 Comparison with Other Techniques for Fabric Defect Classification

As we know, AR feature is a very simple and effective operator to capture enough texture information and has been widely and successfully used in fabric defect detection and pattern classification [2, 9, 10]. It only uses the gray-level values of the pixels which can be directly acquired from mask matching technique. Compared to the feature used in this paper, it can be seen as a raw feature somewhat without any process. We apply the AR feature according to the binary mask and use a window the same as fig.2 to move over every pixel in the initial gray-level image. Every pixel in the defective region is represented by a 25-dimensional vector corresponding to the gray-level values of the 25 effective neighborhood pixels. Then it is subject to PCA and fed into GMM for training. The classification result is shown in fig.7 (a). Although AR feature can cover

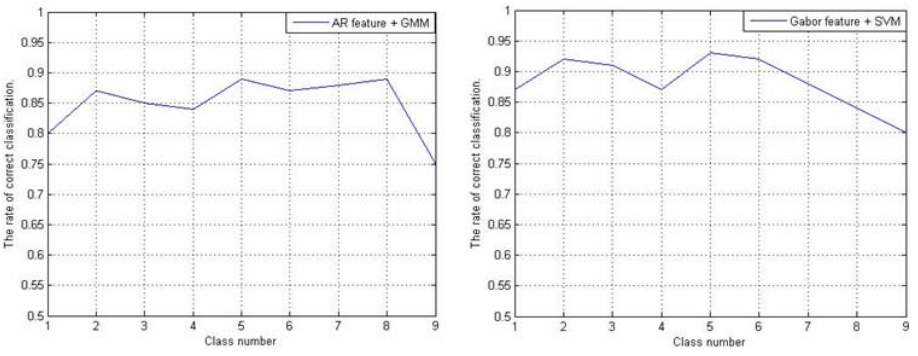


Fig. 7. Classification results using other techniques (a) using AR feature and GMM; (b) using Gabor feature and SVM

enough texture variance, it shows less regularity and contrast of the texture than Gabor feature used in this paper. However, it includes some unrelated information which eliminates the class discrimination. So it only achieves an average rate of 85 percent and no one surpasses 90 percent.

We also compare the classification ability of GMM with SVM [10]. SVM is considered for its linear discrimination ability and its superior performance on small samples. The structure of the classifier using GMM and SVM is implemented almost the same. To the multi-classification problem, for each class, we train a corresponding SVM. The final result is determined by the SVM which produces the maximum output. Gaussian function is selected as the kernel function here and the parameter σ is chosen 0.5 empirically. SVM automatically fix the support vectors of the training data and its number is always great. In this experiment, the average number of support vectors for each class is about 2800, which is much more than the number of GMM for each class. This means in the SVM based classifier, every feature to be estimated will be subject to more than 2800×9 nonlinear transforms, which is much more computationally intensive than GMM. The comparison of the two classifiers using mean and variance is shown in fig.7 (b). 3-model GMM is employed for each class. In this figure, the average rate using SVM is 88 percent, which is a bit higher than 87 percent using GMM. However, the time cost for GMM and SVM is average 45 and 2.5 seconds respectively. On balance, GMM is more suitable for online classification than SVM.

5 Conclusions

In this paper, an algorithm of defect detection and classification using Gabor filters and Gaussian mixture models has been demonstrated. A bank of improved Gabor filters shows its good performance in fabric defect detection. Then we use

a simple but effective cost function to determine the optimal output which is binarized using two thresholds acquired from a normal image. The binary mask of each defect proves the accuracy of our method. In defect classification, with the features based on Gabor filters, we train a GMM based classifier considering different feature combinations. We also find the accurate determination of model number can improve the performance of classification evidently. Our work is still needed to be further developed, for example, the database need to be expanded with more samples of new defects. In order to achieve more accurate classification, other forms of defect feature can be considered such as the geological information of defect. After all, our proposed algorithm can reach an average accuracy of more than 85 percent in classification of 9 different classes, which proves its utility in practice.

Acknowledgments. This work is supported by Chinese National Nature Science Foundation(No.60872141,60802075), State Key Laboratory of Integrated Services Networks Foundation(No.ISN090302), Technology Foundation for University and College of Huawei Co., Ltd(No.YJCB2008052RE).

References

1. Kumar, A.: Computer-Vision-Based Fabric Defect Detection: A Survey. IEEE Transactions on Industrial Electronics 55, 348–363 (2008)
2. Kumar, A.: Neural network based detection of local textile defects. Pattern Recognition 36, 1645–1659 (2003)
3. Kim, S.C., Kang, T.J.: Texture classification and segmentation using wavelet packet frame and Gaussian mixture model. Pattern Recognition 40, 1207–1221 (2007)
4. Kumar, A., Pang, K.H.: Defect Detection in Textured Materials Using Gabor Filters. IEEE Transactions on Industry Applications 38, 425–440 (2002)
5. Mak, K.L., Peng, P.: Detecting Defects in Textile Fabrics with Optimal Gabor Filters. Proceedings of World Academy of Science, Engineering and Technology 13, 75–80 (2006)
6. Randen, T., Husoy, J.H.: Filtering for Texture Classification: A Comparative Study. IEEE Trans Pattern Anal Mach Intell 21, 291–310 (1999)
7. Zhang, D., Kong, W.K., You, J., Wong, M.: Online Palmprint Identification. IEEE Trans. 25(9), 1041–1050 (2003)
8. Bouman, C.A.: Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures, http://www.ece.purdue.edu/~bouman.2001-10
9. Jain, A.K., Karu, K.: Learning Texture Discrimination Masks. IEEE Trans. on Pattern Analysis and Machine Intelligence 18, 195–205 (1996)
10. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support Vector Machines for Texture Classificaion. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(11) (2002)

Moving Object Segmentation in the H.264 Compressed Domain

Changfeng Niu and Yushu Liu

School of Computer Science and Technology
Beijing Institute of Technology, Beijing 100081, P.R. China
niucf@126.com

Abstract. A novel method for moving object segmentation in the H.264 compressed domain is proposed. In contrast to all known methods in which only motion information is used, the proposed method utilizes some characters of H.264 besides motion information with no more decoding required. In the proposed method, motion vector is refined firstly by spatial and temporal correlation of motion and initial segmentation is produced by using the motion vector difference after global motion estimation. Then, the result of segmentation is refined by using intra prediction information in intra-frame. The refined result of segmentation is projected to subsequent frame and expansion and contraction operation is followed. Experimental results for several H.264 compressed video sequences demonstrate the good segmentation quality of the proposed approach.

Keywords: Moving Object Segmentation, Compressed Domain, H.264.

1 Introduction

Moving object segmentation is an interesting and challenging research topic. It has been widely exploited in the various applications such as video surveillance, retrieval tasks and scene analysis. In general, object tracking in the pixel domain is more robust and performs better than compressed domain methods, since more precise information is available. Nevertheless, the motivation for compressed domain analysis remains and is driven by fast processing speed and the fact that videos are primarily stored in compressed form.

For MPEG compressed domain, moving object segmentation algorithms usually rely either on motion vectors (MVs), residual information (DCT coefficients), or both. However, for h.264 video stream, full decoding is necessary to get residual information, so nothing but MVs is used to segment moving object in all known algorithms.

In the paper, we propose a novel moving object segmentation algorithm in which more coding information besides MV in H.264 compressed domain are used with no more decoding required. The main contribution of this paper is twofold. One is that spatial and temporal correlation is used to eliminate MVs noise with an improved MVs similarity measure formula, especially, a spatial filter based on the partition of macroblock is employed. The other one is intra prediction model used to refine object segmentation.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 645–654, 2010.

The paper is organized as follows. First, section 2 reviews some known moving object segmentation method in compressed domain. Section 3 describes our approach to moving object segmentation in H.264 compressed domain. Experimental results are presented in Section 4, and conclusions are given in Section 5.

2 Related Works

A large number of compressed domain object segmentation algorithms appeared over the years. The most of these algorithms are focus on MPEG compression domain. In MPEG compressed video, pictures are encoded in terms of I-frame, P-frame and B-frame. P-frames and B-frames store the motion information and residues of the motion compensation; I-frame has no motion information but stores the DCT transformed signals of the original image. Thus, I-frame can provide texture or color information without decoding. Most of the object segmentation algorithms in compressed domain employ MVs and DCT coefficients to extract moving objects.

Babu et al. [5] proposed an accumulation of motion vectors (MVs) over time, followed by a K-Means clustering to determine the number of objects in the scene and the EM algorithm for object segmentation. Wang et al.[8] transformed Gaussian Mixture background model to compressed domain and used the way similar to pixel domain to segment moving object. Porikli [6] derived some frequency-temporal features from MVs and DCT coefficients for each block, and exploits these features for volume growing from homogenous blocks. MVs are then used to estimate an affine motion model for each volume, and hierarchical clustering is employed to iteratively merge volumes with similar motion into different video objects. Manerba et al. [7] proposed to combine both motion information and region-based color segmentation to extract moving object from MPEG-2 compressed video stream.

A few algorithms which concentrated on the H.264 compressed domain have proposed recently. Zeng et al. [1] employ a block-based Markov Random Field (MRF) model to segment moving objects from the sparse MV field, which is extracted from H.264 compressed streams. The proposed method is limited to static cameras. Liu et al. [2] use accumulated MVs by iteratively backward projection to enhance the salient motion, the residual between the accumulated MVs and global motion vector is used to detect foreground object, and a region growing method is employed to segment foreground object into different moving object. In [3], those blocks whose MVs is not fitted to global motion model are regarded as outlier, and a temporal filter is used to remove the noise in outlier. Then, motion history images are employed to detect moving object from the outlier mask.

3 Moving Object Segmentation in H.264 Compressed Domain

The flowchart of the proposed moving object segmentation in H.264 compressed domain is illustrated in Fig.1. The details are described in the following section.

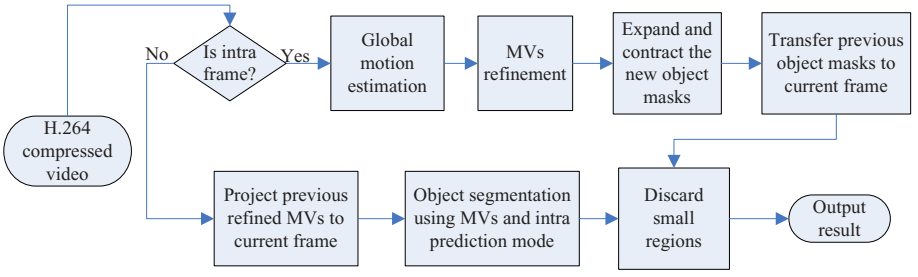


Fig. 1. The flowchart of the proposed moving object segmentation

3.1 Global Motion Estimation

Global motion estimation is performed on the MV field. In order to obtain the MV values in quarter-pel precision, the entropy coding of H.264 has to be reversed. MBs in skip-mode and intra-mode are excluded from the estimation process. We estimate the global motion for each video frame.

A six parameters affine model is adapted as global motion model and is defined as

$$\begin{aligned} d_x &= a_0 x_i + a_1 y_i + a_2 \\ d_y &= a_3 x_i + a_4 y_i + a_5 \end{aligned} \tag{1}$$

where $[a_0, a_1, a_2, a_3, a_4, a_5]^T$ is the affine parameter vector which character the global motion, $(x_i, y_i)^T$ denotes the MB center in pixel coordinate. We estimate the model in least squares method. The process is repeated iteratively and outliers that are 4x4 blocks with large estimation error are discarded after each iteration. It shows that convergence is reached after approximately 3 iterations.

Given the global motion parameter, we can detect the foreground from video by the difference between MVs and the global motion. More details are described in the following section.

3.2 Motion Vector Similarity Measure

To judge the similarity between two motion vectors, an immediate and simplest measure is Euclidean distance, however, the differences in only magnitudes and non direction is considered. An alternate measure is the cosines of angel between motion vectors however, difference in magnitudes is ignored.

In the paper, it is the same as [10] that each component of MV is half-wave rectified into four non-negative channels. By this way, the angle between two MV has been mapped to [0°,90°]. Then, the similarity measure between two MVs is defined as [9] (equation (2)). In equation (2), MV^+ is the rectified motion vector. Equation (2) takes into account the differences in both direction and magnitudes between two motion vectors, and the range of $dist$ is [0, 1] (1, if two MVs is same; 0, if the

direction of two MVs is opposite). Nevertheless, equation (2) is noise sensitive, especially when motion vector have a small value.

$$dist(MV_1, MV_2) = \frac{MV_1^+ \bullet MV_2^+}{\left\|MV_1^+\right\|\left\|MV_2^+\right\|} \min\left(\frac{\left\|MV_1^+\right\|}{\left\|MV_2^+\right\|}, \frac{\left\|MV_2^+\right\|}{\left\|MV_1^+\right\|}\right) = \frac{MV_1^+ \bullet MV_2^+}{\max\left(\left\|MV_1^+\right\|^2, \left\|MV_2^+\right\|^2\right)} \tag{2}$$

To improve the effectiveness of the equation, we add some addition judge condition to the equation. The improved equation is described in equation (3). The effective of equation (3) is illustrated in Fig. 2(b) .

$$dist(MV_1, MV_2) = \begin{cases} 1 & if \begin{pmatrix} abs(MV_1(x) - MV_2(x)) <= 1 and \\ abs(MV_1(y) - MV_2(y)) <= 1 \end{pmatrix} \\ equation(2) & otherwise \end{cases} \tag{3}$$

Equation (3) is used as similarity measure between two MVs and regard two MVs as similar if $dist$ is not less than the predefined threshold (τ), otherwise, no similar. In the paper, τ is set to 0.8.

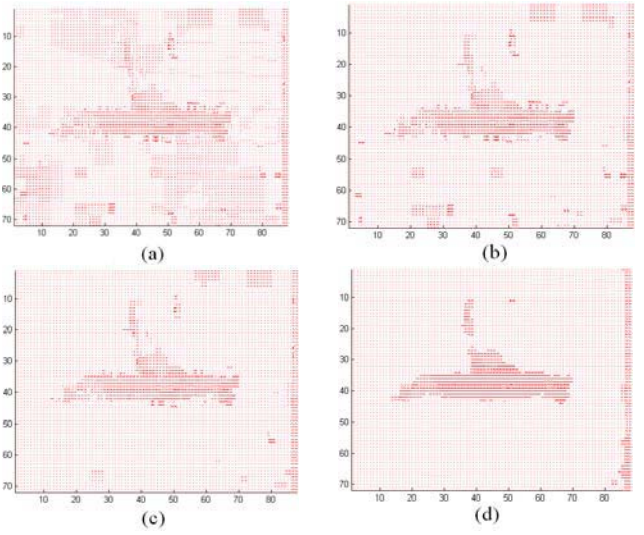


Fig. 2. The difference motion vector field for the 274th of *Coastguard* sequence

Given the similarity measure, the difference MV field is formed as follows: those MVs in the current frame similar to global motions are set to 0 and the others are subtracted by corresponding global motion vector. Then, the initial segmentation is produced by the difference MV field. In Fig.2, the different similarity measure

equations (equation (2) and (3)) are used and the results are shown in Fig.2 (a) and Fig.2 (b) respectively.

3.3 MV Field Spatio-temporal Refinement

In H.264, a macroblock in inter frame can be partitioned into several partitions with different block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4). The raw MV field in H.264 stream is of variant block size. In order to obtain a uniform sampled MV field, the raw MV field is converted into a sparse MV field uniformly sampled at each 4x4 block.

Since MVs are issued from a coding-oriented criterion, the MV field is quantized and noisy. In order to minimize the singularities, some refinement processes are implemented in space and time.

Due to motion continuous in temporal domain, these MVs in the current frame which are similar to the corresponding MVs in the immediately next frame shall be more reliable, especially, for background region, otherwise, the MV is unreliable and likely to be a noise MV. Here, a temporal filter is employed to remove these unreliable MV in background region. The filter is described as follows.

$$MV_{ij} = \begin{cases} GMV_{i,j} & \text{if } dist\left(MV_{ij}, GMV_{ij}\right) < \tau \text{ and } dist\left(MV_{ij}^{c}, GMV_{ij}\right) \geq \tau \\ MV_{i,j} & \text{otherwise} \end{cases} \tag{4}$$

Where MV_{ij} denotes the motion vector of the 4x4 block whose center coordinates is i, j. MV_{ij}^{c} is the corresponding MV of the MV_{ij} in the immediately next frame, the way to compute corresponding MV will be described in the last of the section. The definition of $dist$ and τ are the same as section 3.2. $dist\left(MV_{ij}, GMV_{ij}\right) < \tau$ means the MV_{ij} is not fitted to global motion and the block owning MV_{ij} doesn't belongs to background. The effective of the filter is illustrated in Fig.2 (c).

For the spatial refinement, we make use of the conclusion about the reliability of MV from [11]. In [11], these MVs which significantly different from all of its neighboring motion vectors had been proved to be not reliably. As known that each 4x4 block in one partition has the same motion vector in H.264 stream, we judge if the motion vector in one partition is similar to those of neighboring partitions and refine the motion vector according to the judgment. The refinement process is described as: if no similar, the MV in the partition is replaced with the weighted mean of the neighboring motion vectors, otherwise, it keeps unchanged. The effective of the spatial refinement is illustrated in Fig.2 (d).

The corresponding MV of the MV in the current frame is calculated as illustrated in Fig. 3. After backward projecting the 4x4 block in current frame using the minus MV, we compute the overlapping areas between original and projected block. Then, we can get MV of corresponding block from these original blocks with respect to the ratio of the overlapping area to these block size of these original blocks. In the paper, the MV of corresponding block is called the corresponding MV for short.

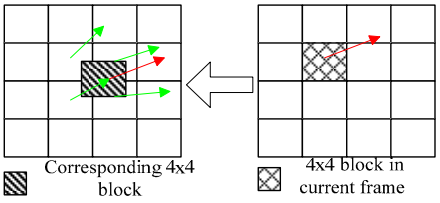


Fig. 3. The method to calculate corresponding motion vector

3.4 Object Detection in Intra Frame

In H.264, Intra frame coding employs spatial prediction to deduce spatial residues, and encoder determines a prediction for each block from neighboring pixels in Intra frame. Block size for luminance component is 4x4 or 16x16. There are nine prediction models for 4x4 blocks and 4 models for 16x16 blocks. Prediction model for each block can be easily gotten with only partial decoding required. The nine prediction modes for each 4x4 luma block are shown in Fig. 4. It can be seen that I4MB prediction is conducted for samples a-p of a block using samples A-Q. There are in total eight "prediction directions" and one DC prediction mode for I4MB prediction, The name of prediction modes and the assigned directions are as follows [4]: Vertical (0), Horizontal (1), DC (Mean of neighboring pixels), Diagonal down left (3), Diagonal down right (4), Vertical right (5), Horizontal down(6),Vertical left (7), Horizontal up (8).

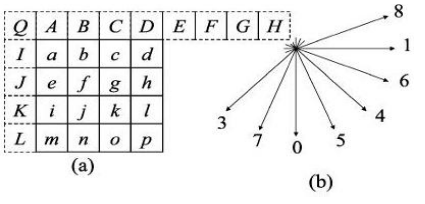


Fig. 4. (a) I4MB prediction coding is conducted for samples a-p of a block using samples A-Q. (b) Eight "prediction directions" for I4MB prediction

The selected prediction model for a block in intra frame implies the way that pixels in the block are related to neighboring pixels of the block. So, we propose to detect object with combining prediction model and motion information in intra frame.

Since the motion vectors for an I-frame block don't exist, a forward projection is used to project the motion vectors of immediate adjoint P-frame to I-frame. After the motion vectors projected, a method same as the section 3.2 is used to find the moving object and some examples of segmentation results are shown in fig.7. Since not all 4x4 block contain real motion information (the aperture problem and the limitation of codec), the moving objects are not detected completely.

After segmenting motion object initially using only motion information, a refinement process is implemented depending on the intra prediction model. In the process, we regard those blocks which have been previously detected as 'seed blocks' and check the immediate neighboring blocks of these seed blocks to judge whether some

blocks should be add to motion object by the rules described as follows, where $Mask$, MV and GMV are M*N matrix, where M and N denote the width and height of video frame in 4x4 block. $Mask$ represents the initial segmentation result. The element of $Mask$ which is equal to zero implies that a block belongs to background, otherwise, foreground. Both MV and GMV are used to store MV and global MV for every 4x4 block respectively.

The refinement process of segmentation result by using the intra prediction

If $Mask_{ij} = 0$ and $MV_{ij} \neq GMV_{ij}$

If ($Mode_{ij} = 1$ and $Mask_{i-1\,j} = 1$) or ($Mode_{i+1\,j} = 1$ and $Mask_{i+1,\,j} = 1$)

$Mask_{ij} = 1$

End if

if ($Mode_{ij} = 0$ and $Mask_{ij-1} = 1$) or ($Mode_{ij+1} = 0$ and $Mask_{ij+1} = 1$)

$Mask_{ij} = 1$

End if

If ($Mode_{ij} = 3$ or $Mode_{ij} = 7$) and ($Mask_{ij-1} = 1$ and $Mask_{i+1\,j-1} = 1$)

$Mask_{ij} = 1$

End if

If ($Mode_{ij} = 4$ or $Mode_{ij} = 5$ or $Mode_{ij} = 6$ or $Mode_{ij} = 2$)

and ($Mask_{ij-1} = 1$ and $Mask_{i-1\,j} = 1$)

$Mask_{ij} = 1$

End if

End if

In the rules above, the refinement process using only 4x4 prediction mode is described, and the 16x16 prediction mode can be utilized by the similar rules. In addition, we ignore the planar prediction in 16x16 prediction mode because it shows poor directivity and occurs scarcely.

3.5 Refinement in Non-intra Frame

When we have gotten the independent moving object in previous frame, the goal to the section is to refine the object mask in current frame(non-intra frame). We treat the problem as follows.

These 4x4 blocks in object mask are backward projected to current frame by the minus MVs. However, due to the object possibly deform, the previous object mask is not always coincident with current moving object. So, a calibration process is necessary to fit object mask to moving object. The process is composed of two steps and is described as follows.

1) Expansion

The object mask is expanded when some blocks are not covered by the object mask but have motion vector similar to the one in the object mask.

2) Contraction

When some blocks are covered by the object mask and have MVs similar to global motion, these blocks are eliminated from the moving object.

It must be mentioned that the MVs used in expansion and contraction have been filtered in the space and time as described in section 3.3.

4 Experiment and Result

The proposed method has been evaluated on several video sequences compressed using the H.264 encoder of JM9. The encoder configuration set as follows: baseline profile (including non B frame), the interval of I-frames is 12, quantization parameter (QP) is 30, and the MV search range is [-32, 32].

In Fig.5, the segmentation result is shown from *Coastguard* sequence. The first column illustrates the original frames and the last two columns are the object mask and the final result using the proposed method. In addition, the second column is the object mask using the method in [2]. It can be seen that the proposed method extracts object with more precise edge from two object masks in two middle columns. The approach from [2] extracts object correctly, however, with a high false positive. It is reason that the motion vectors on the object edge rarely present the real motion. On the other hand, in the proposed method, spatio-temporal filter can exclude these unreliable MV and the refinement processes as described in section 3.4 and 3.5 ensure the precision of segmentation.

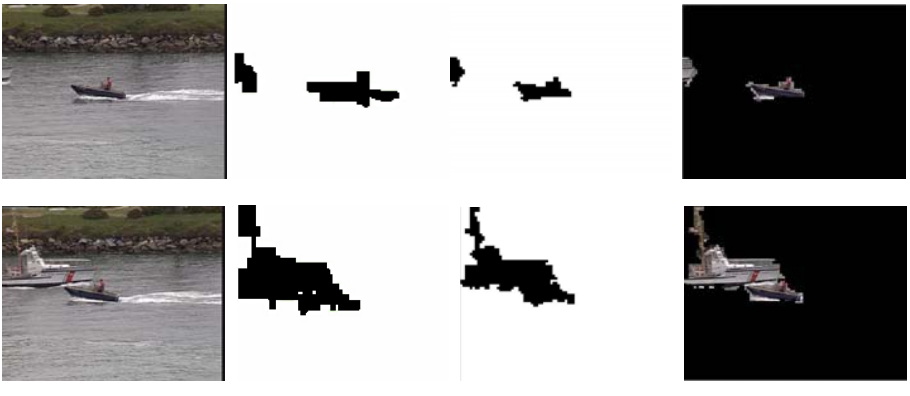


Fig. 5. The result of segmentation in [2]'s approach and the proposed approach. The first row is the 1st frame in Coastguard and encoded as I-frame. The second row is the 37st frame and encoded as P-Frame.

Furthermore, for the purpose of objective evaluation, two commonly used measurements, i.e. precision and recall, are calculated. The high precision means less over-segmentation, while high recall means less miss-segmentation. In Fig.6, the precision and recall curves of Coastguard sequence for both the proposed approach and the approach in [2] are shown. From these curves, it can be clearly seen that the proposed approach improves the precision and maintains the recall at the same time.

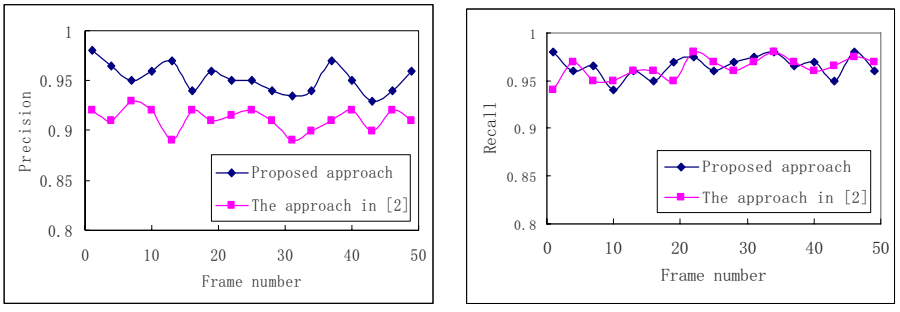


Fig. 6. Precision and recall curves for the two video segmentation approaches. The left figure is precision curve, and the right figure is recall curve.

In Fig. 7, we show the final results of segmentation and intermediate results for other sequences. The two frames in the first and third column are intra-fame. For the intra frames, the intermediate results are produced by only these projected MVs. The final results come from the refinement process using intra prediction information as described in section 3.4. For the inter frame, the intermediate results are obtained by projecting the object mask from the previous frame. From Fig.7, we can see the projected object mask is not fitted to the real object completely. By the refinement process in non-intra frame, we obtain a precise segmentation result in the end.

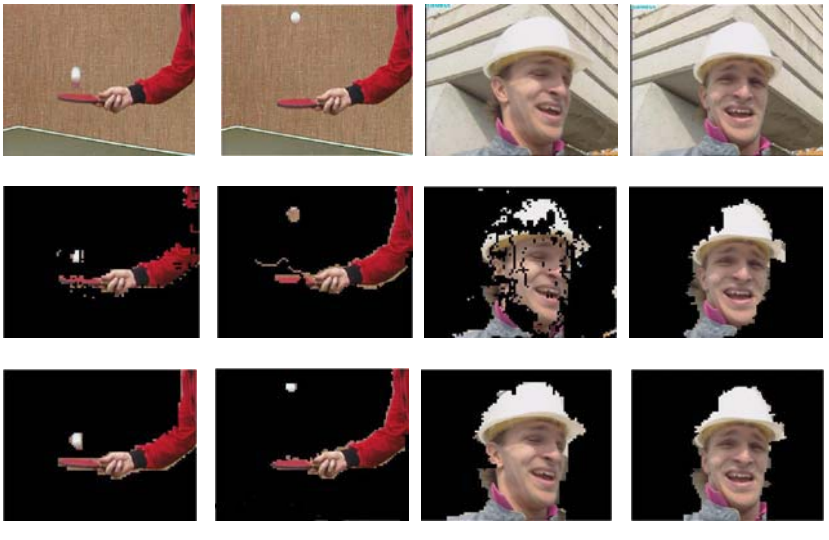


Fig. 7. The results of moving object segmentation for tennis and forman sequence. The first row is origin picture, the second is intermediate result, the last is final result. The fist and second column is 12st and 17stframe in tennis sequence respectively. The third and fourth column is 12st and 17st in foreman sequence respectively.

5 Conclusions

In this paper, we present a novel moving object segmentation algorithm in the H.264 compressed domain. Partition and intra prediction mode besides motion is used to refine the segmentation and no more decoding required. A spatio-temporal filter is employed to reduce the MVs noise. The proposed approach is demonstrated to reliably segment moving objects with good quality from the H.264 compressed video.

References

[1] Zeng, W., Du, J., Gao, W., Huang, Q.M.: Robust moving object segmentation on H.264/AVC compressed video using the block-based MRF model. Real-Time Imaging 11, 290–299 (2005)
[2] Liu, Z., Lu, Y., Zhang, Z.: Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain. Journal of visual communication and image representation 18, 275–290 (2007)
[3] Kas, C., Nicolas, H.: An Approach to Trajectory Estimation of Moving Objects in the H. 264 Compressed Domain. In: PSIVT, Japon, Tokyo (2009)
[4] Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding, Final Draft International Standard, ISO/IEC FDIS 14 496-10 (December 2004)
[5] Babu, R.V., Ramakrishnan, K., Srinivasan, S.: Video object segmentation: a compressed domain approach. IEEE Transactions on Circuits Systems for Video Technology 14(4), 462–474 (2004)
[6] Porikli, F.: Real-time video object segmentation for MPEG encoded video sequences. In: Proc. SPIE Conference on Real-Time Imaging VIII, San Jose, vol. 5297, pp. 195–203 (2004)
[7] Manerba, F., Benois-Pineau, J., Leonardi, R., et al.: Multiple moving object detection for fast video content description in compressed domain. EURASIP Journal on Advances in Signal Processing 2008 (2008)
[8] Wang, W., Yang, J., Gao, W.: Modeling Background and Segmenting Moving Objects from Compressed Video. IEEE Transactions on Circuits Systems for Video Technology 18(5), 670–681 (2008)
[9] Yeo, C., Ahammad, P., Ramchandran, K., Sastry, S.: High-Speed Action Recognition and Localization in Compressed Domain Videos. IEEE Transactions on Circuits Systems for Video Technology 18(8), 1006–1015 (2008)
[10] Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. IEEE Int. Conf. Comput. Vis., Nice, France, October 2003, pp. 726–733 (2003)
[11] Wang, D., Vincent, A., Blanchfield, P.: Hybrid De-Interlacing Algorithm Based on Motion Vector Reliability. IEEE Transactions on Circuits Systems for Video Technology 15(8), 1019–1026 (2005)

Video Segmentation Using Iterated Graph Cuts Based on Spatio-temporal Volumes

Tomoyuki Nagahashi[1], Hironobu Fujiyoshi[1], and Takeo Kanade[2]

[1] Dept. of Computer Science, Chubu University
Matsumoto 1200, Kasugai, Aichi, 487-8501 Japan
nagahashi@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp
http://www.vision.cs.chubu.ac.jp
[2] The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213-3890 USA
tk@cs.cmu.edu

Abstract. We present a novel approach to segmenting video using iterated graph cuts based on spatio-temporal volumes. We use the mean shift clustering algorithm to build the spatio-temporal volumes with different bandwidths from the input video. We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from graph cuts in the previous process, and set the probability as the t-link of the graph for the next process. The proposed method can segment regions of an object with a stepwise process from global to local segmentation by iterating the graph-cuts process with mean shift clustering using a different bandwidth. It is possible to reduce the number of nodes and edges to about 1/25 compared to the conventional method with the same segmentation rate.

1 Introduction

The video segmentation that extracts object's region in a video sequence captured by a hand-held camera is a difficult problem. This technique is extremely important because it is often used in preprocessing for object recognition, and gesture recognition.

The interactive graph cuts proposed by Boykov *et al.* [1] [2] haves been used in recent years for segmenting images. The energy function in interactive graph cuts is minimized by creating the graph from the correct-answer label and the input image that the user gave, and using a minimum cut/maximum flow algorithm. Nagahashi *et al* proposed image segmentation using iterated graph cuts based on multi-scale smoothing[3].

This segmentation of image based on graph cuts can be applied to video segmentation using the same framework. However, the size of the graph for a video sequence increases because we have to create the graph by making all pixels in the video. This causes, two main problems, i.e., we need large amounts of memory and it increases the computation cost. To overcome these problems, a graph constructed from spatio-temporal volumes has been used to reduce the

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 655–666, 2010.

size of the graph[4][5]. However, it is difficult to precisely video segment video due to its low resolution.

We propose a method that represents spatio-temporal space as video that extends the technique of iterated graph cuts based on multi-scale smoothing[3] to spatio-temporal volumes obtain a stepwise process from global to local segmentation by iteration. Our approach uses mean shift clustering to build the spatio-temporal volumes with different bandwidths from the input video. We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from graph cuts in the previous process, and set the probability as the t-link of the graph for the next process. The proposed method can segment the regions of an object with a stepwise process from global to local segmentation by iterating the graph-cuts process with mean shift clustering using different bandwidth.

2 Graph Cuts for Video Segmentation

This section describes the graph-cuts-based segmentation proposed by Boykov and Jolly[1].

2.1 Graph Cuts for Image Segmentation

An image-segmentation problem can be posed as a binary-labeling problem. Let us assum that the image is a graph $G = (V, E)$, where V is the set of all nodes and E is the set of all arcs connecting adjacent nodes. The nodes are usually pixels p on the image P and the arcs have adjacency relationships with four or eight connections between neighboring pixels $q \in N$. The labeling problem is to assign a unique label L_i to each node $i \in V$, i.e., $L_i \in \{\text{“}obj\text{”}, \text{“}bkg\text{”}\}$. The solution $\boldsymbol{L} = \{L_1, L_2, \ldots, L_p, \ldots, L_{|P|}\}$ can be obtained by minimizing the Gibbs energy $E(\boldsymbol{L})$:

$$E(\boldsymbol{L}) = \lambda \cdot \sum_{p \in P} R_p(L_p) + \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(L_p, L_q) \tag{1}$$

where

$$\delta(L_p, L_q) = \begin{cases} 1 \text{ if } & L_p \neq L_q \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

The coefficient, $\lambda \geq 0$, in Eq. (1) specifies the relative importance of the region properties term $R_p(L_p)$ versus the boundary properties term $B_{\{p,q\}}$. Regional term assumes that the individual penalties for assigning pixel p to “obj” and “bkg”, corresponding to $R_p(\text{“}obj\text{”})$ and $R_p(\text{“}bkg\text{”})$ are given. For example, $R_p(\cdot)$ may reflect how the intensity of pixel p fits into a known intensity model (e.g., a histogram) of the object and background. Term $B_{\{p,q\}}$ comprises the “boundary” properties of segmentation $\boldsymbol{L}$. Coefficient $B_{\{p,q\}} \geq 0$ should be interpreted as a penalty for the discontinuity between p and q. $B_{\{p,q\}}$ is normally large when

pixels p and q are similar (e.g., in intensity) and $B_{\{p,q\}}$ is close to zero when these two differ greatly. The penalty $B_{\{p,q\}}$ can also decrease as a function of distance between p and q. Costs $B_{\{p,q\}}$ may be based on the local intensity gradient, Laplacian zero-crossing, gradient direction, or other criteria.

Table 1 lists the edge cost of the graph. The regional and boundary terms in

Table 1. Edge cost

Edge		Cost	For
n-link	$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$
t-link	$\{p,S\}$	$\lambda \cdot R_p(\text{"}bkg\text{"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		K	$p \in \mathcal{O}$
		0	$p \in \mathcal{B}$
	$\{p,T\}$	$\lambda \cdot R_p(\text{"}obj\text{"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		0	$p \in \mathcal{O}$
		K	$p \in \mathcal{B}$

Table 1 are calculated by

$$R_p(\text{“}obj\text{”}) = -\ln \Pr(I_p|\mathcal{O}) \tag{3}$$

$$R_p(\text{“}bkg\text{”}) = -\ln \Pr(I_p|\mathcal{B}) \tag{4}$$

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)} \tag{5}$$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}}. \tag{6}$$

Let $\mathcal{O}$ and $\mathcal{B}$ define the “object” and “background” seeds. The seeds are given by the user. The boundary between the object and the background is segmented by finding the minimum cost cut [6] on the graph, G.

2.2 Problems with Conventional Method

Interactive Graph Cuts [2] create a graph from video.Thus, the size of the graph from video increases when placing individual pixels into a node. For example, the total number of the edges will be 25 million, when the input video is 360×240 with 100 frames. Therefore, we need copious amounts of memory and it takes a long time for processing with the minimum cut/maximum flow algorithm. One common technique of solving such this problem is to reduce the size of the graph by using a spatio-temporal volume. However, it is difficult to precisely segment regions and boundaries because segmentation using spatio-temporal volumes has low resolution. To overcome this problem, we propose a method that represents spatio-temporal space as video that extends the technique of iterated graph cuts based on multi-scale smoothing[3] to spatio-temporal volumes to obtain a stepwise process from global to local segmentation by iteration.

3 Iterated Graph Cuts Using Spatio-temporal Volumes

3.1 Proposed Method

We extend the technique of iterated graph cuts based on multi-scale smoothing [3] to spatio-temporal volumes.

Objects that move fast may be divided into different volumes between frames in a row when using spatio-temporal volumes. Therefore, it is difficult to create an optimal graph by only using adjoining volumes. To solve these problems, we introduced two kinds of edges, i.e., a volume that adjoins as an n-link, and a volume obtained from a search for corresponding points between frames.

Energy Function. A volume pair that adjoins as the n-link, and a volume pair obtained by searching for the corresponding points between frames are used in the proposed method. Therefore, we extend the energy function using the graph cuts discussed Section 2.1 as follows:

$$E(\boldsymbol{L}) = \lambda \cdot \sum_{p \in P} R_p(L_p) + \sum_{\{p,q\} \in N} B_{N\{p,q\}} \cdot \delta(L_p, L_q) + \sum_{\{p,q\} \in C} B_{C\{p,q\}} \cdot \delta(L_p, L_q) \quad (7)$$

where $p, q \in P$ is a spatio-temporal volume, N is a neighboring volume of p and C represents corresponding points between frames. By using $B_{C\{p,q\}}$ in the energy function, we obtain robust segmentation results even if divided into different volume between frames.

Flow of Proposed Method. Figure 1 shows the flow for of the new approach. First, the seeds, "foreground" and "background", are given by the user. Next, we obtain the spatio-temporal volume using mean shift clustering using bandwidth h. Graph cuts are done to segment the video into an object or a background. The Gaussian Mixture Model (GMM) is then used to make a color distribution model for the object and background classes from the segmentation results obtained from the graph cuts. The prior probability is updated from the distance transform by the object and background classes of GMM. The t-links for the

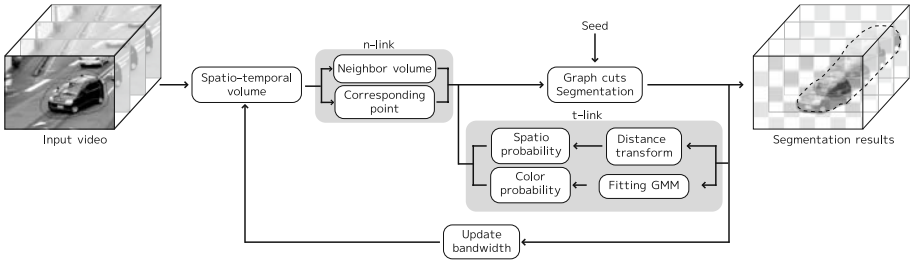


Fig. 1. Overview of proposed method

next graph-cuts process are calculated as a posterior probability which is computed a prior probability and GMMs, and h is updated as, $h = \alpha \cdot h$. These processes are repeated until $h < th$.

The processes are as follows.

Step 1 Input seeds
Step 2 Create spatio-temporal volume
Step 3 Search corresponding points
Step 4 Do graph cuts
Step 5 Calculate the posterior probability from the segmentation results and set as the t-link
Step 6 Update $h = \alpha \cdot h$, and Steps 1-5 are repeated until $h < th$.

The details of each process are given in what follows.

3.2 Spatio-temporal Volume

We employ mean shift clustering [7] to obtain the spatio-temporal volume. Let the space, time, and color information vector denote $\mathbf{x}_i = \{\mathbf{x}_i^s, \mathbf{x}_i^t, \mathbf{x}_i^r\}$, the filtering result denote $\mathbf{z}_i$, and each label denote L_i. $\{\mathbf{y}_j\}_{j=1,2,\ldots}$ is defined as

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} \tag{8}$$

$$g(\mathbf{x}) = \frac{C}{h_s^2 h_t h_r^p} k\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^t}{h_t}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|^2\right), \tag{9}$$

where h_s, h_t, h_r is the bandwidth by space, time, and color, C is the normalizing constant, $k(\mathbf{x})$ is the kernel function (e.g., Gaussian distribution). Mean shift clustering involves main four steps and an optional one.

1. Initialize $y_{i,j} = x_i$
2. Compute $y_{i,j+1} Ck \leftarrow k + 1$ until convergence $\mathbf{z}_i = \mathbf{y}_{i,c}$ is reached.
3. Identify clusters $\{C_p\}_{p=1,\ldots,m}$ of convergence points by linking together all $\mathbf{z}_i$ that are closer than 0: 5 from one an other in the joint domain.
4. $L_i = \{p | \mathbf{z}_i \in C_p\}$
5. Optional: Eliminate spatial regions smaller than M pixels.

Figure 2 shows examples of spatio-temporal volumes with different bandwidths. In Fig. 2, we can see that each volume is decreased by decreasing the bandwidth. We represent global and local information using a spatio-temporal volume with different bandwidths in the proposed method. Then, a graph is created from nodes that correspond to spatio-temporal volumes segmented by mean shift clustering.

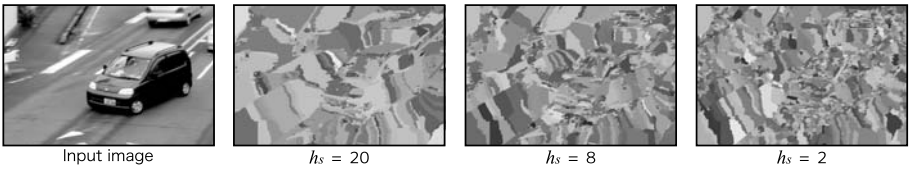


Fig. 2. Examples of Spatio-temporal volumes

3.3 Add Edges Using Corresponding Points

Objects that move fast may be divided into different volumes between frames when using spatio-temporal volumes. Figure 3 shows an example of adding an edge using corresponding points. The edge has not been calculated because two the volumes are not neighbors. In our approach, we add an edge from the corresponding points. The corresponding points are computed by matching keypoints using SIFT [8] in two frames. This helps to correct two volumes, that are not in the neighborhood, corresponding to the same object. Consequently, volumes that are not in the neighborhood are represented as the same object.

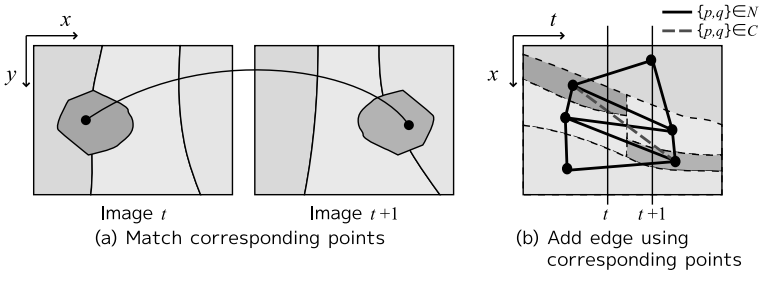


Fig. 3. Example of adding edge using corresponding point

3.4 Iterated Graph Cuts

We have discussed segmenting of video using graph cuts using a spatio-temporal volume that is created from video using mean shift clustering and employing iteration from large to small bandwidths. We will not describe the method of updating the n- and t-links, and the effect of iterated processing.

Update n-link. The n-link represents information between neighboring nodes. The volume pair that adjoins n-link $B_N(\boldsymbol{L})$, and the volume pair obtained by searching for corresponding points between frames $B_C(\boldsymbol{L})$ are used in the new method. $B_N(\boldsymbol{L})$, $B_C(\boldsymbol{L})$ is given by

$$B_{\{p,q\}} = \exp\left(-\frac{\|\boldsymbol{I}_p - \boldsymbol{I}_q\|^2}{2\sigma^2}\right), \tag{10}$$

where I_p is the color in volume p.

Update t-link. We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from the graph cuts in the previous process, and set the probability as the t-link using

$$R'_p(\text{“}obj\text{”}) = -\ln \Pr(\mathcal{O}|I_p) \tag{11}$$

$$R'_p(\text{“}bkg\text{”}) = -\ln \Pr(\mathcal{B}|I_p) \tag{12}$$

where $\Pr(\mathcal{O}|I_p)$ and $\Pr(\mathcal{B}|I_p)$ are given by

$$\Pr(\mathcal{O}|I_p) = \frac{\Pr(\mathcal{O})\Pr(I_p|\mathcal{O})}{\Pr(I_p)} \tag{13}$$

$$\Pr(\mathcal{B}|I_p) = \frac{\Pr(\mathcal{B})\Pr(I_p|\mathcal{B})}{\Pr(I_p)}. \tag{14}$$

$\Pr(I_p|\mathcal{O})$ and $\Pr(I_p|\mathcal{B})$ are the computed color probabilities and $\Pr(\mathcal{O})$ and $\Pr(\mathcal{B})$ are computed spatial probabilities from the segmentation results using graph cuts in the previous process.

Updating color probability. The color probabilities $\Pr(I_p|\mathcal{O})$ and $\Pr(I_p|\mathcal{B})$ are computed by using GMM [9]. The GMM for the RGB color space is obtained by

$$\Pr(I_p|\cdot) = \sum_{i=1}^{K} \alpha_i p_i(I_p|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{15}$$

where $p_i(I_p|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is gaussian distribution. We used the EM algorithm to fit the GMM [10].

Updating spatial probability. The spatial probabilities $\Pr(\mathcal{O})$ and $\Pr(\mathcal{B})$ are updated by spatial information from the graph cuts in the previous process. The next segmentation label is uncertain in the vicinity of the boundary. Therefore, the spatial probability is updated by using the results of a distance transform [11]. The distance from the boundary is normalized from 0.5 to 1. Let d_{obj} denote the distance transform of the object, and d_{bkg} denote the distance transform of the background. The prior probability is given by

$$\Pr(\mathcal{O}) = \begin{cases} d_{obj} & \text{if } d_{obj} \geq d_{bkg} \\ 1 - d_{bkg} & \text{if } d_{obj} < d_{bkg} \end{cases} \tag{16}$$

$$\Pr(\mathcal{B}) = 1 - \Pr(\mathcal{O}). \tag{17}$$

Color probability can be spatially controled using spatial probability. Consequently, the segmentation that is observed in the boundary possible in the next graph cuts. Therefore, we can obtain more robust segmentation even if the video contaions the same objects.

Iteration. Finally, using $\Pr(I_p|\mathcal{O})$ and $\Pr(I_p|\mathcal{B})$ from GMM, and $\Pr(\mathcal{O})$ and $\Pr(\mathcal{B})$ from the distance transform, posterior probability can be computed by means of Eqs. (11) and (12). We compute the prior probability obtained by the likelihood from a color histogram and the distance transform, and set the probability as the t-link of the graph for the next process using the segmentation results obtained by using the graph cuts in the previous process.

4 Experimental Results

4.1 Experiment Outline

We used 13 videos including those of a vehicle moving, a human walking, a flower, and a leaf captured with a hand-held camera outdoors. A seed was only given to the first frame. We evaluated the segmentation results for the 10th frame comparing them with those from a manually correct mask. We defined a true positive (TP) as the number of objects of correct detection pixels, a false positive (FP) as the number of backgrounds of missed detection pixels, and a false negative (FN) as the number of objects of missed detection pixels. We evaluated the recall , precision, and F-measure as

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{19}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{20}$$

We compared three conventional methods and two methods we propose.

Conventional method 1. This involves Boykov's graph cuts approach [2]. Each pixel is a node obtained by using a graph.

Conventional method 2. This uses the spatio-temporal volume.

Conventional method 3. This involves iterating segmentation such as GrabCut [12] with a spatio-temporal volume .

Proposed method 1. Our approach involves iterating segmentation with a spatio-temporal volume using different bandwidths. However, we did not use spatial probability.

Proposed method 2. Our approach involves iterating segmentation with a spatio-temporal volume using different bandwidths with spatial probability.

4.2 Comparison with Conventional Method

Figure 4 is a bar chart with the segmentation rate and Fig. 5 shows example segmentation results with three of the methods.

Effect Using Spatio-temporal Volume. Conventional method 1 in FIg.4 can obtain better segmentation than Conventional method 2 whose using spatio-temporal volume has lower resolution than that of the formaer. Therefore, poor segmentation is obtained with Conventional method 2.

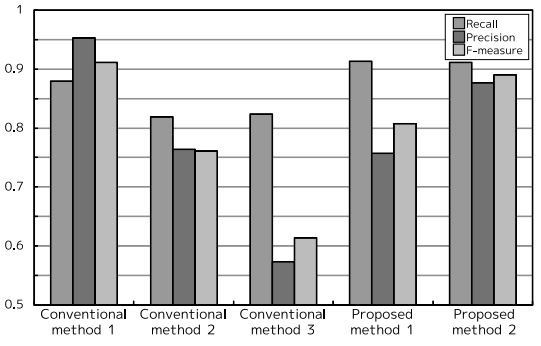


Fig. 4. Segmentation rate

Effect Iterating Process. We compared Conventional method 2 with Conventional method 3, which had repetition processing added. The recall was same rate, but precision with Conventional method 3 was lower than with Conventional method 2. It was difficult to detect the background when a spatio-temporal volume was used (see Fig. 4, Conventionals method 1 and 2). Therefore, the background color was learned as an object color in the iterating process. Figure 5(c)(d) shows the segmentation results for Conventional methods 2 and 3. We can see that the false detection of the background has gradually been extended by the iterating process.

Effect of Iterating Process by Changing Bandwidth. Proposed method 1 is better at segmentation, its recall is better at 0.09Cits precision is better at 0.18Cand its f-measure is better at 0.19, than those of Conventional method 3.

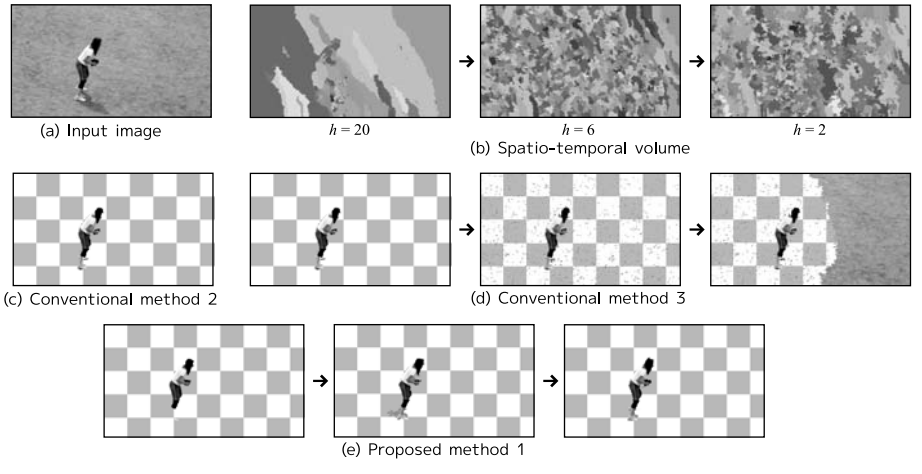


Fig. 5. Example segmentation results

Figure 5(e) shows the segmentation results for the iterating process obtained by changing the bandwidth. We can see that Proposed method 1 can reduce missed detection in the background. Figure 5(d) shows Conventional method 3 detects many incorrect small volumes in the background because the color looks like the object. When the bandwidth in mean shift clustering is large, the spatio-temporal volume is large as shown in Fig. 5(b). Although we obtained coarse segmentation results, these were not incorrect volumes. By changing the bandwidth, we could obtain more precise segmentation like that in coarse-to-fine approach.

Effect of Spatial Probability by Distance Transform. Proposed method 2 using spatial probability has better Precision at 0.12 than Proposed method 1. Figure 6 shows the segmentation results in a sequence that has the same object. Proposed method 1 that only uses color probability cannot segment correctly, e.g., it detects the leaf, which has not been specified. However, we can see that Proposed method 2 can detect the leaf, which has been specified.

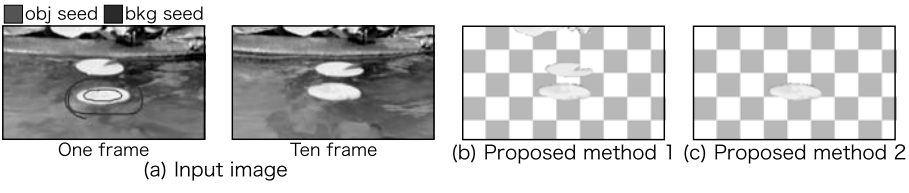


Fig. 6. Example segmentation results using distance transform

Overall, Proposed method 2 using spatio-temporal volumes could obtain a segmentation rate comparable to that of Conventional method 1.

4.3 Comparison of Graph Size

Table 2 lists the graph size with each method. The bandwidth of Conventional methods is $h = 2$, and the results by using Proposed method were obtained by changing bandwidth h from 20 to 2. Compared with Conventional method 1, the proposal technique was able to reduce the number of edges about 6.1% and the number of nodes to about 3.3%. It was possible to reduce the number of nodes and edges to about 1/25 compared to the conventional method with the same segmentation rate.

Table 2. Graph size

	Conventional method 1 [2]	Conventional method 2, 3	Proposed method 1, 2
Node	864,000	52,993	43,140 - 52,993
Edge	2,505,600	81,239	17,477 - 81,549

4.4 Effect of Adding Edge Using Corresponding Points

We evaluated how effective it was to add edges using corresponding points. It is difficult to segment objects with Conventional method when they moves fast. We used video at 6 fps in this experiment. We compared Proposed method where edges were added using corresponding points and Conventional method where edges were not added using corresponding points. The proposed method could obtain better segmentation than the conventional method. Fewer errors were detected because corresponding points were matched between volumes that were not neighbors by frame.

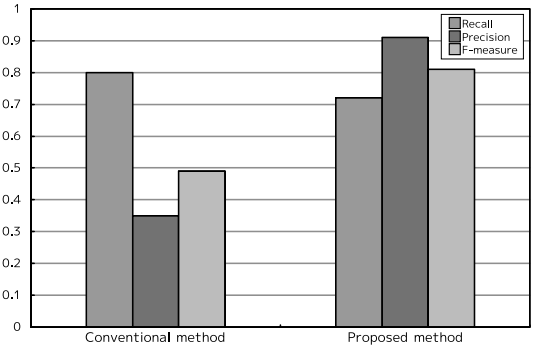


Fig. 7. Segmentation results

5 Conclusion

We presented a novel approach to video segmentation using iterated graph cuts based on spatio-temporal volumes. We used the mean shift clustering algorithm to build the spatio-temporal volumes with different bandwidths from the input video. We computed the prior probability obtained by the likelihood from the color probability and the spatial probability using the segmentation results from graph cuts in the previous process, and set the probability as the t-link of the graph for the next process. It is possible to reduce the number of nodes and edges to about 1/25 comparing to the conventional method with the same segmentation rate.

We would like to investigate features other than color in the future. In addition, we would like to accelerate segmentation processing.

References

1. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: ICCV 2001, vol. 01, p. 105 (2001)
2. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. Int. J. Comput. Vision 70(2), 109–131 (2006)
3. Nagahashi, T., Fujiyoshi, H., Kanade, T.: Image segmentation using iterated graph cuts based on multi-scale smoothing. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 806–816. Springer, Heidelberg (2007)

4. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Interactive video cutout. In: SIGGRAPH 2005: ACM SIGGRAPH 2005 Papers, pp. 585–594. ACM Press, New York (2005)
5. Li, Y., Sun, J., Shum, H.Y.: Video object cut and paste. In: SIGGRAPH 2005: ACM SIGGRAPH 2005 Papers, pp. 595–600. ACM Press, New York (2005)
6. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)
7. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
9. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition (CVPR 1999), Los Alamitos, pp. 246–252 (1999)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
11. Toyofumi, S., Junichiro, T.: Euclidean distance transformation for three dimensional digital images. The transactions of the Institute of Electronics, Information and Communication Engineers 76(3), 445–453 (1993)
12. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)

Spectral Graph Partitioning Based on a Random Walk Diffusion Similarity Measure

Xi Li[1,⋆], Weiming Hu[1], Zhongfei Zhang[2], and Yang Liu[1]

[1] National Laboratory of Pattern Recognition, CASIA, Beijing, China
{lixi,wmhu,yangliu}@nlpr.ia.ac.cn
[2] State University of New York, Binghamton, NY 13902, USA
zhongfei@cs.binghamton.edu

Abstract. Spectral graph partitioning is a powerful tool for unsupervised data learning. Most existing algorithms for spectral graph partitioning directly utilize the pairwise similarity matrix of the data to perform graph partitioning. Consequently, they are incapable of fully capturing the intrinsic structural information of graphs. To address this problem, we propose a novel random walk diffusion similarity measure (*RWDSM*) for capturing the intrinsic structural information of graphs. The *RWDSM* is composed of three key components—emission, absorbing, and transmission. It is proven that graph partitioning on the *RWDSM* matrix performs better than on the pairwise similarity matrix of the data. Moreover, a spectral graph partitioning objective function (referred to as *DGPC*) is used for capturing the discriminant information of graphs. The *DGPC* is designed to effectively characterize the intra-class compactness and the inter-class separability. Based on the *RWDSM* and *DGPC*, we further develop a novel spectral graph partitioning algorithm (referred to as *DGPCA*). Theoretic analysis and experimental evaluations demonstrate the promise and effectiveness of the developed *DGPCA*.

1 Introduction

In recent years, spectral graph partitioning has been successfully applied to many domains such as circuit layout[1][2], load balancing [3] and image segmentation [4][5][6][7]. Based on local evidence from similarities among data points, spectral graph partitioning finds out the best graph cuts by optimizing a particular partitioning objective function through eigendecomposition. With effectiveness in clustering data of complex structure, spectral graph partitioning is promising for multiclass data learning.

Much work has been done in spectral graph partitioning. Shi and Malik [4] propose a normalized cut criterion for segmenting the similarity graph. Gdalyahu *et al.* [8] present a "typical cut" algorithm for graph partitioning. Ding *et al.* [9] present a min-max cut algorithm for graph partitioning and data clustering. Balanced partitions are obtained by the min-max cut algorithm. Ng *et al.* [10] present a clustering algorithm based on K-Means after the spectral relaxation. The aforementioned spectral graph partitioning methods have a common problem that they only considers the pairwise relations of nodes on the graph without characterizing the interaction information among different nodes. Based on [10], Zelnik-Manor and Perona [11] propose an improved spectral

⋆ The author has moved to CNRS, TELECOM ParisTech. Email: xi-li@telecom-paristech.fr

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 667–676, 2010.

clustering algorithm, which computes the affinity matrix via a local scale scheme. The number of clusters is determined by exploiting the structure of the eigenvectors of the normalized graph Laplacian. The disadvantage of this algorithm is the sensitivity to significant noise. Yu and Shi [12] propose a multiclass spectral clustering algorithm, which gives a nearly global-optimal discrete clustering solution by using singular value decomposition and non-maximum suppression in an iterative procedure. But this algorithm performs poorly in capturing the intrinsic structural information of data samples on the graph. In [13], Verma and Meila make a comparison of popular spectral clustering algorithms. Nadler *et al.* [14] give a diffusion based probabilistic interpretation of spectral clustering algorithms based on the eigenvectors of the normalized graph Laplacian. Nadler and Galun [15] discuss the fundamental limitations of spectral clustering and propose a novel diffusion based measure for evaluating the coherence of individual clusters. Further, Li *et al.* [16] propose a noise robust spectral clustering algorithm. But this algorithm is poor in characterizing the discriminant information of graphs. Moreover, Li *et al.* [17] present a discriminant analysis based spectral clustering algorithm for effectively capturing the graph's local marginal information characterized by the intra-class compactness and the inter-class separability. However, the limitation of this algorithm is the ignorance of the intrinsic structural information of graphs. Consequently, it is very important for spectral graph partitioning algorithms to capture the intrinsic structural and discriminant information of graphs simultaneously.

In this paper, we propose a novel random walk diffusion similarity measure (*RWDSM*), which characterizes the potential interactions among the nodes of a graph by using random walk on the graph. Specifically, the *RWDSM* is composed of three components—emission, absorbing, and transmission. We perform graph partitioning on the *RWDSM* matrix instead of the pairwise similarity matrix of the data. Moreover, a spectral graph partitioning objective function (referred to as *DGPC*) is used to fully capture the discriminant information of graphs. The *DGPC* effectively characterizes the intra-class compactness and the inter-class separability. By maximizing the inter-class separability and intra-class compactness, the *DGPC* obtains an optimal graph partitioning solution.

2 Spectral Graph Partitioning

An N-node weighted graph $\mathbb{G} = (\mathbb{V}, \mathbb{E}, W)$ is used to represent the intrinsic relationships among N data points, where $\mathbb{V} = \{1, \ldots, N\}$ is the node set, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the edge set, and $W = (w_{ij})_{N \times N}$ is a similarity matrix with the element w_{ij} being the edge weight between nodes i and j. Clustering N data points into K classes is equivalent to partitioning $\mathbb{V}$ into K disjoint subsets, namely, $\mathbb{V} = \bigcup_{l=1}^{K} \mathbb{V}_l$ s.t. $\mathbb{V}_m \bigcap \mathbb{V}_n = \emptyset$, $\forall m \neq n$. For convenience, let $\Gamma_{\mathbb{V}}^{K} = \{\mathbb{V}_1, \mathbb{V}_2, \ldots, \mathbb{V}_K\}$. Given $\mathbb{V}_a, \mathbb{V}_b \subset \mathbb{V}$, links$(\mathbb{V}_a, \mathbb{V}_b)$ is defined as the sum of the total weighted connections between $\mathbb{V}_a$ and $\mathbb{V}_b$: links$(\mathbb{V}_a, \mathbb{V}_b) = \sum_{i \in \mathbb{V}_a} \sum_{j \in \mathbb{V}_b} w_{ij}$ Moreover, the degree of $\mathbb{V}_a$ is defined as the total links of nodes in $\mathbb{V}_a$ to all the nodes in $\mathbb{V}$, i.e., degree$(\mathbb{V}_a)$ = links$(\mathbb{V}_a, \mathbb{V})$. Subsequently, one classic objective function (i.e., K-way normalized cuts) for spectral graph partitioning is defined as:

$$\Gamma_{cut} = \arg\min_{\Gamma_{\mathbb{V}}^{K}} kncut(\Gamma_{\mathbb{V}}^{K}) = \arg\min_{\Gamma_{\mathbb{V}}^{K}} \frac{1}{K} \sum_{k=1}^{K} \frac{\text{links}(\mathbb{V}_k, \mathbb{V} \backslash \mathbb{V}_k)}{\text{degree}(\mathbb{V}_k)} \tag{1}$$

2.1 Solving K-Way Normalized Cuts

In [12], it has been proven that solving the K-way normalized cuts is equivalent to finding the optima of an optimization program:

$$\begin{aligned} \textbf{max} \quad & f(X) = \frac{1}{K}\sum_{n=1}^{K}\frac{X_n^T W X_n}{X_n^T D X_n} \\ \textbf{s.t.} \quad & X \in \{0,1\}^{N\times K}, X\mathbb{1}_K = \mathbb{1}_N \end{aligned} \tag{2}$$

where X is an $N \times K$ partition matrix, $\mathbb{1}_d$ denotes a $d \times 1$ vector with each element being 1, D is an $N \times N$ diagonal matrix with the m-th diagonal element being the sum of the elements belonging to the m-th row of W for $1 \leq m \leq N$, and X_n is the n-th column of X for $1 \leq n \leq K$. In [12], an iterative procedure is adopted to obtain the optimal solution to Eq. (2). Please see [12] for details.

3 Random Walk Diffusion Similarity Measure

3.1 Our Defined One-Step Random Walk on a Graph

Starting from node i, we directly move to node j after one step. Let $p_{ij}^{s_1} = p_{i\to j}$ be the one-step probability that the random walk starts from node i and stops on node j. $p_{ij}^{s_1}$ consists of three components. One is the emission probability $e_{ij}^{s_1} = p_{e,i\to j}$, another is the absorbing probability $a_{ij}^{s_1} = p_{a,i\to j}$, and the other is the transmission probability $r_{ij}^{s_1} = p_{r,i\to j}$ containing both emission and absorbing information. Fig. 1 illustrates the aforementioned three components. In this way, we have $p_{ij}^{s_1} = (e_{ij}^{s_1} + a_{ij}^{s_1} + r_{ij}^{s_1})/3$ with $e_{ij}^{s_1} = \frac{w_{ij}}{\sum_k w_{ik}}$, $a_{ij}^{s_1} = \frac{w_{ij}}{\sum_k w_{kj}}$, and $r_{ij}^{s_1} = (e_{ij}^{s_1} + a_{ij}^{s_1})/2$. Please see [14] for fundamental properties of random walk on a graph.

In what follows, we give a brief description of notations used hereinafter. Let $D_r = \text{Diag}(d_{11}^r, d_{22}^r, \ldots, d_{NN}^r)$ be the diagonal matrix with the i-th diagonal element being $d_{ii}^r = \sum_k w_{ik}$, $D_c = \text{Diag}(d_{11}^c, d_{22}^c, \ldots, d_{NN}^c)$ be the diagonal matrix with the j-th diagonal element being $d_{jj}^c = \sum_k w_{kj}$, $E^{s_1} = (e_{ij}^{s_1})_{N\times N}$ be the one-step emission probability matrix, $A^{s_1} = (a_{ij}^{s_1})_{N\times N}$ be the one-step absorbing probability matrix, $R^{s_1} = (r_{ij}^{s_1})_{N\times N}$ be the one-step transmission probability matrix, and $P^{s_1} = (p_{ij}^{s_1})_{N\times N}$ be the one-step random walk probability matrix. Apparently, we have $E^{s_1} = D_r^{-1}W$, $A^{s_1} = WD_c^{-1}$, and $R^{s_1} = (D_r^{-1}W + WD_c^{-1})/2$. Thus, the one-step random walk probability matrix $P^{s_1} = (p_{ij}^{s_1})_{N\times N}$ can be formulated as:

$$P^{s_1} = (E^{s_1} + A^{s_1} + R^{s_1})/3. \tag{3}$$

3.2 Our Generalized m-Step Random Walk on a Graph

Starting from node i, we indirectly move to node j after m steps (s.t. $m > 1$). Without loss of generality, we denote any m-step random walk path as $i \to k_1 \to \cdots \to k_{m-1} \to j$ with k_l being a hidden random variable for $k_l \in \{1, \ldots, N\}$ and $l \in \{1, \ldots, m-1\}$. For simplicity, $i \to k_1 \to \cdots \to k_{m-1} \to j$ is referred to as $\mathscr{P}^m$. Thus, the random walk probability along $\mathscr{P}^m$ is defined as:

$$p_{\mathscr{P}^m} = (p_{e,\mathscr{P}^m} + p_{a,\mathscr{P}^m} + p_{r,\mathscr{P}^m})/3, \tag{4}$$

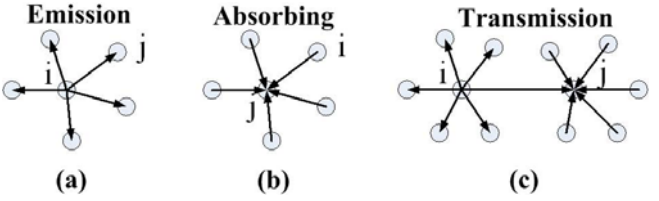


Fig. 1. Illustration of random walk on a graph. (a) shows the emission random walk from node i to node j; (b) displays the absorbing random walk from node i to node j; (c) exhibits the transmission random walk (containing both emission and absorbing information) from node i to node j.

where

$$\begin{aligned} p_{e,\mathscr{P}^m} &= e^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} e^{s_1}_{k_l k_{l+1}}\right) e^{s_1}_{k_{m-1}j}, \\ p_{a,\mathscr{P}^m} &= a^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} a^{s_1}_{k_l k_{l+1}}\right) a^{s_1}_{k_{m-1}j}, \\ p_{r,\mathscr{P}^m} &= r^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} r^{s_1}_{k_l k_{l+1}}\right) r^{s_1}_{k_{m-1}j}. \end{aligned} \tag{5}$$

Moreover, the terms $e^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} e^{s_1}_{k_l k_{l+1}}\right)$, $a^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} a^{s_1}_{k_l k_{l+1}}\right)$, and $r^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} r^{s_1}_{k_l k_{l+1}}\right)$ are just the $(m-1)$-step emission, absorbing, and transmission probabilities along the path $i \to k_1 \to \cdots \to k_{m-1}$, respectively. Therefore, $e^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} e^{s_1}_{k_l k_{l+1}}\right) = e^{s_{m-1}}_{ik_{m-1}}$. Similarly, $a^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} a^{s_1}_{k_l k_{l+1}}\right) = a^{s_{m-1}}_{ik_{m-1}}$ and $r^{s_1}_{ik_1}\left(\prod_{l=1}^{m-2} r^{s_1}_{k_l k_{l+1}}\right) = r^{s_{m-1}}_{ik_{m-1}}$. In this way, Eq. (5) can be further simplified as:

$$p_{e,\mathscr{P}^m} = e^{s_{m-1}}_{ik_{m-1}} e^{s_1}_{k_{m-1}j}, \quad p_{a,\mathscr{P}^m} = a^{s_{m-1}}_{ik_{m-1}} a^{s_1}_{k_{m-1}j}, \quad p_{r,\mathscr{P}^m} = r^{s_{m-1}}_{ik_{m-1}} r^{s_1}_{k_{m-1}j}. \tag{6}$$

In addition, we let $e^{s_m}_{ij}$, $a^{s_m}_{ij}$, and $r^{s_m}_{ij}$ be the m-step emission, absorbing, and transmission probabilities from node i to node j, respectively. Then, we have:

$$\begin{aligned} e^{s_m}_{ij} &= \textstyle\sum_{k_{m-1}} p_{e,\mathscr{P}^m} = \sum_{k_{m-1}} e^{s_{m-1}}_{ik_{m-1}} e^{s_1}_{k_{m-1}j}, \\ a^{s_m}_{ij} &= \textstyle\sum_{k_{m-1}} p_{a,\mathscr{P}^m} = \sum_{k_{m-1}} a^{s_{m-1}}_{ik_{m-1}} a^{s_1}_{k_{m-1}j}, \\ r^{s_m}_{ij} &= \textstyle\sum_{k_{m-1}} p_{r,\mathscr{P}^m} = \sum_{k_{m-1}} r^{s_{m-1}}_{ik_{m-1}} r^{s_1}_{k_{m-1}j}. \end{aligned} \tag{7}$$

Further, we let $E^{s_m} = (e^{s_m}_{ij})_{N\times N}$, $A^{s_m} = (a^{s_m}_{ij})_{N\times N}$, and $R^{s_m} = (r^{s_m}_{ij})_{N\times N}$ be the m-step emission, absorbing, and transmission probability matrices, respectively. Hence, $p^{s_m}_{ij}$ can be further simplified as:

$$p^{s_m}_{ij} = \tfrac{1}{3}[E^{s_{m-1}}(i,:)E^{s_1}(:,j) + A^{s_{m-1}}(i,:)A^{s_1}(:,j) + R^{s_{m-1}}(i,:)R^{s_1}(:,j)]. \tag{8}$$

According to Eq. (8), we have:

$$E^{s_m} = E^{s_{m-1}}E^{s_1}, \quad A^{s_m} = A^{s_{m-1}}A^{s_1}, \quad R^{s_m} = R^{s_{m-1}}R^{s_1}. \tag{9}$$

After solving the above recursive equation, we have:

$$E^{s_m} = (E^{s_1})^m, \quad A^{s_m} = (A^{s_1})^m, \quad R^{s_m} = (R^{s_1})^m. \tag{10}$$

Consequently, the final m-step random walk probability matrix $P^{s_m} = (p^{s_m}_{ij})_{N\times N}$ is formulated as:

$$P^{s_m} = \tfrac{1}{3}\left[E^{s_m} + A^{s_m} + R^{s_m}\right] = \tfrac{1}{3}\left[(E^{s_1})^m + (A^{s_1})^m + (R^{s_1})^m\right]. \tag{11}$$

Given: A data set $Z = \{z_1, z_2, \ldots, z_N\}$ and the number of classes K:

1. Create a weighted graph with no self-loops $\mathbb{G} = (\mathbb{V}, \mathbb{E}, W)$, where $\mathbb{V} = \{1, \ldots, N\}$ is the node set, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ represents the edge set, and $W = (w_{ij})_{N \times N}$ is a similarity matrix with the element w_{ij} being the edge weight between nodes i and j:

$$w_{ij} = \begin{cases} \exp(-\text{dist}(z_i, z_j)/2\sigma^2) & \text{if} \quad i \neq j \\ 0 & \text{otherwise} \end{cases}$$

in which σ is a scaling factor, and $\text{dist}(\cdot)$ denotes a distance function. In the experiments, $\text{dist}(z_i, z_j) = \|z_i - z_j\|_F^2$.

2. Compute the *RWDSM* matrix Φ by Eq. (13).

3. Obtain $Q = \mathbb{D} - \Phi$ where $\mathbb{D}$ is the diagonal matrix with $\mathbb{d}_{ii} = \sum_j \phi_{ij}$ for $1 \leq i \leq N$. If Q is a singular matrix, it should be replaced with $Q + \epsilon I_N$, where ϵ is a small positive constant and I_N is an $N \times N$ identity matrix.

4. Form $\widetilde{P}$ by the normalized K largest eigenvectors of $Q^{-1}\Phi$.

5. Obtain a candidate graph partitioning solution $\widetilde{X}$ by: $\widetilde{X} = \text{Diag}(\text{diag}^{-\frac{1}{2}}(\widetilde{P}\widetilde{P}^T))\widetilde{P}$.

6. Perform the iterative refining procedure [12] on $\widetilde{X}$ to find an optimal solution X to Eq. (14). The refining procedure is discussed in detail in steps four to eight of the algorithm in [12].

Fig. 2. The specific procedure of *DGPCA*

3.3 Random Walk Diffusion Similarity Measure

In order to capture the topological structural information of graphs, we propose a novel random walk diffusion similarity measure (*RWDSM*), which is the sum of the aforementioned random walk probability matrices after different steps. For convenience, we denote the *RWDSM* matrix corresponding to the graph $\mathbb{G}$ as $\Phi = (\phi_{ij})_{N \times N}$, which is formulated as:

$$\begin{aligned} \Phi &= \textstyle\sum_{m=1}^{M} P^{s_m} = \frac{1}{3}\sum_{m=1}^{M}[E^{s_m} + A^{s_m} + R^{s_m}] \\ &= \textstyle\frac{1}{3}\sum_{m=1}^{M}[(E^{s_1})^m + (A^{s_1})^m + (R^{s_1})^m] \\ &= \textstyle\frac{1}{3}\sum_{m=1}^{M}\left[(D_r^{-1}W)^m + (WD_c^{-1})^m + \left(\frac{D_r^{-1}W + WD_c^{-1}}{2}\right)^m\right] \end{aligned} \tag{12}$$

where I_N is an $N \times N$ identity matrix, and M is a controlling factor ($M = 6$ in the experiments). Consequently, any element ϕ_{ij} of Φ captures the random walk diffusion information between nodes i and j on the graph $\mathbb{G}$. When $\mathbb{G}$ is an undirected graph, D_r is equal to D_c. In this case, the *RWDSM* matrix $\Phi = (\phi_{ij})_{N \times N}$ is a symmetric matrix. Let $D_r = D_c = D$. After substituting D into Eq. (12), we have:

$$\Phi = \frac{1}{3}\sum_{m=1}^{M}\left[(D^{-1}W)^m + (WD^{-1})^m + \left(\frac{D^{-1}W + WD^{-1}}{2}\right)^m\right] \tag{13}$$

4 Our Spectral Graph Partitioning Objective Function

Before discussing our spectral graph partitioning objective function (referred to as *DGPC*), we first give a brief introduction to the notations used hereinafter. Let $\mathbb{D}$ be

the diagonal matrix with the i-th diagonal element being $\mathbb{d}_{ii} = \sum_j \phi_{ij}$ for $1 \leq i \leq N$ and $Q = \mathbb{D} - \Phi$. Consequently, the *DGPC* is written as:

$$\begin{aligned} \textbf{max } g(X) &= \frac{1}{K}\sum_{n=1}^{K} \frac{X_n^T \Phi X_n}{X_n^T Q X_n} \\ &= \frac{1}{K}\sum_{n=1}^{K} \frac{[X_n (X_n^T X_n)^{-\frac{1}{2}}]^T \Phi [X_n (X_n^T X_n)^{-\frac{1}{2}}]}{[X_n (X_n^T X_n)^{-\frac{1}{2}}]^T Q [X_n (X_n^T X_n)^{-\frac{1}{2}}]} \\ \textbf{s.t. } & X \in \{0,1\}^{N \times K}, X \mathbb{1}_K = \mathbb{1}_N \end{aligned} \tag{14}$$

where X_n represents an $N \times 1$ vector formed by the n-th column of X. The analytical proof of the above objective function is given as follows.

The intra-class compactness and the inter-class separability are respectively captured by $X_n^T \Phi X_n = \sum_{i \in \mathbb{V}_n} \sum_{j \in \mathbb{V}_n} \phi_{ij}$ and $X_n^T Q X_n = \sum_{i \in \mathbb{V}_n} \sum_{j \notin \mathbb{V}_n} \phi_{ij}$, where $\mathbb{V}_n$ denotes the node set belonging to the n-th class. The larger the value of $X_n^T \Phi X_n$, the more compact the intra-class samples. The smaller the value of $X_n^T Q X_n$, the more separable the inter-class samples. As a result, an optimal graph partitioning solution is obtained by maximizing the $g(X)$ in (14).

4.1 Finding Optimal Solutions

After a sequence of simplification operations, the graph partitioning objective function (14) becomes: $g(X) = \frac{1}{K} tr\{(P^T Q P)^{-1}(P^T \Phi P)\}$, where tr denotes the trace of a matrix, and $P = X(X^T X)^{-\frac{1}{2}}$ which is constrained by: $P^T P = [X(X^T X)^{-\frac{1}{2}}]^T \cdot [X(X^T X)^{-\frac{1}{2}}] = I_K$ where I_K is a $K \times K$ identity matrix, and $X^T X$ is a diagonal matrix. Thus, the objective function (14) can be rewritten as:

$$\begin{aligned} \textbf{max } & h(P) = \frac{1}{K} tr\{(P^T Q P)^{-1}(P^T \Phi P)\} \\ \textbf{s.t. } & P^T P = I_K. \end{aligned} \tag{15}$$

The optimization problem (15) has been addressed in multiclass LDA (linear discriminant analysis) learning [18]. A solution $\widetilde{P}$ to (15) consists of the K principal eigenvectors (i.e., corresponding to the K largest eigenvalues) of the matrix $Q^{-1}\Phi$. If Q is a singular matrix, $Q^{-1}\Phi$ should be replaced with the matrix $(Q + \epsilon I_N)^{-1}\Phi$, where I_N is an $N \times N$ identity matrix and ϵ is a small positive constant ($\epsilon = 1e-6$ in the paper). As a result, a candidate solution $\widetilde{X}$ to (14) is obtained by: $\widetilde{X} = \text{Diag}(\text{diag}^{-\frac{1}{2}}(\widetilde{P}\widetilde{P}^T))\widetilde{P}$, where $\text{Diag}(\cdot)$ denotes a diagonal matrix formed from its vector argument, and $\text{diag}(\cdot)$ represents a vector formed from the diagonal elements of its matrix argument. Subsequently, the iterative refining procedure [12] may be used to find the optimal graph partitioning solution X to (14). Finally, we have the spectral graph partitioning algorithm (referred to as *DGPCA*) with its specific procedure listed in Fig. 2.

5 Experiments

In order to evaluate the performance of the *DGPCA*, nine datasets are used in the experiments. The first seven datasets are synthetic toy datasets, as shown in Figs. 3 and 4.

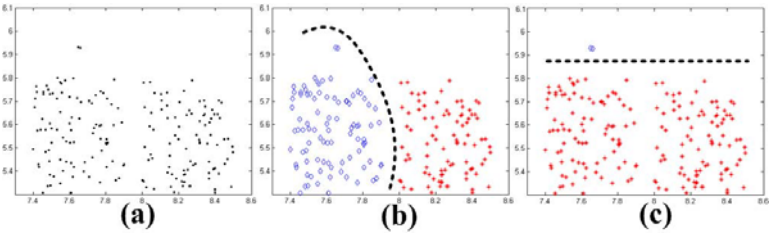


Fig. 3. Clustering performances of *MSC* and *DGPCA* over the first dataset in outlier removal. (a) shows the original data samples. (b) and (c) display the clustering results of *MSC* and *DGPCA*, respectively. It is clear that *DGPCA* succeeds in removing outliers (i.e., uppermost isolated samples in (a)) while *MSC* fails.

The eighth dataset is the Yale face dataset.[2] It contains 165 images of 15 persons. Each person has 11 images. The last dataset is a subset of the US Postal Service handwritten digits recognition corpus (USPS) dataset. It consists of four hundred 16×16 images of four-class handwritten digits (i.e., digits 1, 2, 3, and 4). Each class has 100 images.

Three experiments are conducted to demonstrate the claimed contributions of our spectral graph partitioning algorithm (referred to as *DGPCA*). In these three experiments, we compare clustering results of *DGPCA* with those of a spectral clustering algorithm [4], referred to here as *MSC*. *MSC* is a representative spectral clustering algorithm, which can efficiently obtain the nearly global-optimal graph partitioning solution by solving a discrete optimization problem. In contrast to *MSC*, *DGPCA* is based on the *RWDSM* which captures the intrinsic properties of graphs. Besides, it uses a spectral graph partitioning objective function (referred to as *DGPC*) to fully capture the discriminant information of graphs. Consequently, it is interesting and desirable to make a comparison between *MSC* and *DGPCA*. More details of *MSC* are given in [4]. In addition, we introduce the learning accuracy to make quantitative evaluations of *MSC* and *DGPCA*. The learning accuracy $\mathscr{L}$ is defined as: $\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\frac{n_i}{N_i}$, where N denotes the learned number of classes, N_i represents the number of the samples belonging to the i-th learned class and n_i is the number of the samples whose true class labels have the highest proportion in the i-th learned class.

The first experiment is to compare the clustering performances of *MSC* and *DGPCA* using the first seven datasets. During the process of constructing the corresponding graphs of *MSC* and *DGPCA*, the scaling factor σ is chosen as 0.6. The final clustering results are shown in Figs. 3 and 4. More specifically, Fig. 3(a) plots the original data samples from the first dataset. The partitioning results of *MSC* and *DGPCA* are respectively exhibited in Figs. 3(b) and (c), where the two dashed curves denote the corresponding partitioning boundaries. From Figs. 3(b) and (c), it is seen that *DGPCA* succeeds in removing outliers (i.e., uppermost isolated samples in Fig. 3(a)) while *MSC* fails. For a better demonstration of the effectiveness of *DGPCA* in capturing the intrinsic structural information of graphs, we make experimental evaluations over the remaining six datasets consisting of data samples of complex geometric shapes. The final experimental results are shown in Fig. 4, where Figs. 4(a)-(f) are associated with Datasets 2-7,

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

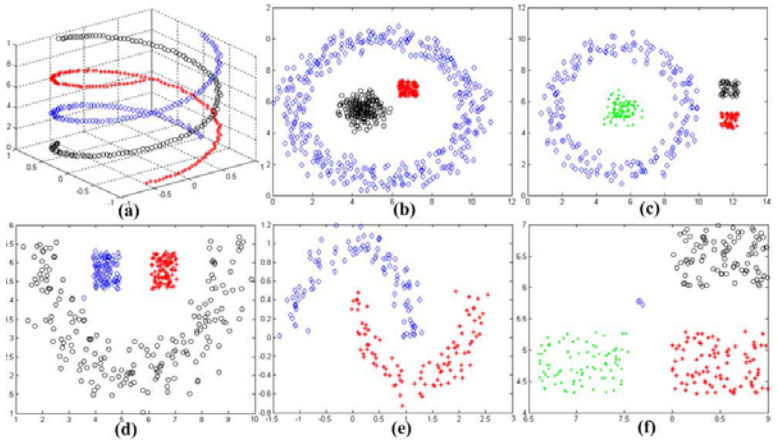


Fig. 4. Clustering results of *DGPCA* over Datasets 2-7 corresponding to (b)-(f), respectively

respectively. From Fig. 4, we see that *DGPCA* performs well in capturing the intrinsic structural information of data samples.

The second experiment is performed to make a quantitative comparison between *MSC* and *DGPCA* in learning accuracy using the eighth dataset. The scaling factor σ is set as 100. Similar to the second experiment, five subsets of the eighth dataset are used for experimental evaluations. They are obtained by independently sampling 3-class samples from the eighth dataset for five times. For these subsets, the ratios of 3-class samples are different. Specifically, their ratios are 100:100:100, 100:60:60, 80:50:30, 30:50:30, and 60:10:60, respectively. Some exemplar samples from the eighth dataset

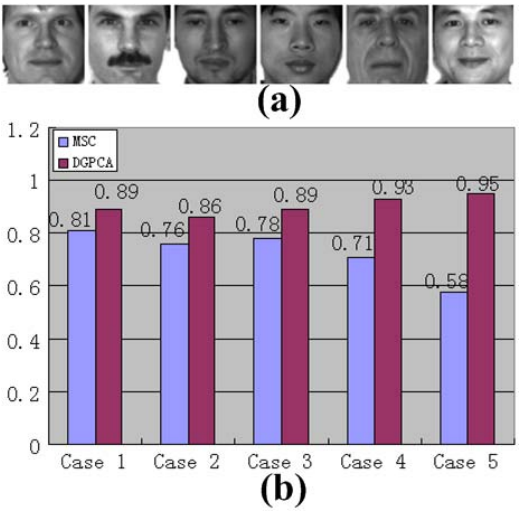


Fig. 5. Clustering results of *MSC* and *DGPCA* over the eighth dataset. (a) shows some exemplar samples while (b) reports the learning accuracies of *MSC* and *DGPCA* in different cases.

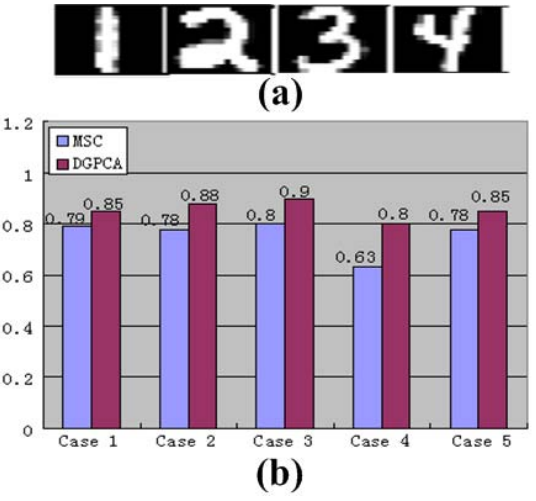


Fig. 6. Clustering results of *MSC* and *DGPCA* over the ninth dataset. Some samples are shown in the right-most side of this figure. (a) shows some exemplar samples while (b) reports the learning accuracies of *MSC* and *DGPCA* in different cases.

are displayed in Fig. 5(a). The final clustering results are reported in Fig. 5(b), where x-axis corresponds to the five different sampling cases while y-axis is associated with the learning accuracy. From Fig. 5(b), we see that *DGPCA* performs better in multiclass data learning than *MSC*.

The last experiment is conducted for a comparison between *MSC* and *DGPCA* in learning accuracy using the ninth dataset. The scaling factor σ is assigned as 6. Similar to the third experiment, five subsets of the ninth dataset are used for experimental evaluations. They are obtained by independently sampling 4-class samples from the ninth dataset for five times. For these subsets, the ratios of 4-class samples are different. Specifically, their ratios are 60:60:60:30, 80:60:60:30, 60:60:60:60, 60:30:60:20, and 50:50:30:20, respectively. Some exemplar samples from the ninth dataset are displayed in Fig. 6(a). The final clustering results are reported in Fig. 6(b), where x-axis corresponds to the five different sampling cases while y-axis is associated with the learning accuracy. From Fig. 6(b), we see that *DGPCA* achieves better learning performances than *MSC*.

In summary, we observe that *DGPCA* outperforms *MSC* in multiclass data learning. *DGPCA* makes a full use of the intrinsic structural information of data samples by constructing the *RWDSM* matrix. Especially for imbalanced multiclass samples, *DGPCA* achieves much better performances than *MSC*, as demonstrated in the last two experiments. In comparison, *MSC* directly uses the similarity matrix for data learning, leading to the weakness in characterizing the intrinsic structural information of graphs. Consequently, *DGPCA* is a promising method for multiclass data learning.

6 Conclusion

In this paper, a novel *RWDSM* (random walk diffusion similarity measure) has been presented to capture the intrinsic properties of graphs. The *RWDSM* is composed of

three components—emission, absorbing, and transmission. Instead of the pairwise similarity matrix of the data, the *RWDSM* matrix is used for graph partitioning. In addition, the *DGPC* (spectral graph partitioning objective function) has been used to fully capture the discriminant information of graphs. The *DGPC* is capable of effectively characterizing the intra-class compactness and the inter-class separability. By maximizing the inter-class separability and intra-class compactness, the *DGPC* obtains an optimal graph partitioning solution. Further, we have developed a spectral graph partitioning algorithm (referred to as *DGPCA*) for multiclass data learning. Experimental results have demonstrated the robustness and promise of the developed *DGPCA*.

References

1. Alpert, C.J., Kahng, A.B.: Multiway partitioning via geometric embeddings, orderings and dynamic programming. IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems 14(11), 1342–1358 (1995)
2. Chan, P.K., Schlag, M.D.F., Zien, J.Y.: Spectral k-way ratio-cut partitioning and clustering. IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems 13(9), 1088–1096 (1994)
3. Hendrickson, B., Leland, R.: An improved spectral graph partitioning algorithm for mapping parallel computations. SIAM J. Sci. Comput. 16(2), 452–459 (1995)
4. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on PAMI 22(8), 888–905 (2000)
5. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. In: IJCV (2001)
6. Weiss, Y.: Segmentation using eigenvectors: A unifying view. In: Proc. ICCV, pp. 975–982 (1999)
7. Meila, M., Shi, J.: Learning segmentation by random walks. In: NIPS, pp. 873–879 (2000)
8. Gdalyahu, Y., Weinshall, D., Werman, M.: Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. IEEE Trans. on PAMI 23(10), 1053–1074 (2001)
9. Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: Proc. ICDM, pp. 107–114 (2001)
10. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: NIPS. MIT Press, Cambridge (2001)
11. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS, pp. 1601–1608 (2005)
12. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: Proc. ICCV, vol. 1, pp. 313–319 (2003)
13. Verma, D., Meila, M.: A comparison of spectral clustering algorithms. Technical Report 03-05-01, University of Washington Department of Computer Science (2003)
14. Nadler, B., Lafon, S., Coifman, R., Kevrekidis, I.: Diffusion maps, spectral clustering and eigenfunctions of fokkerplanck operators. In: NIPS (2005)
15. Nadler, B., Galun, M.: Fundamental Limitations of Spectral Clustering. In: NIPS (2006)
16. Li, Z., Liu, J., Chen, S., Tang, X.: Noise Robust Spectral Clustering. In: Proc. ICCV (2007)
17. Li, X., Zhang, Z., Wang, Y., Hu, W.: Multiclass Spectral Clustering Based on Discriminant Analysis. In: Proc. ICPR (2008)
18. Ma, J., Sancho-Gomez, J.L., Ahalt, S.C.: Nonlinear multiclass discriminant analysis. IEEE Signal Processing Letters 10(7), 196–199 (2003)

Iterated Graph Cuts for Image Segmentation

Bo Peng[1], Lei Zhang[1,⋆], and Jian Yang[2]

[1] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
[2] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing , 210094, China
{csbpeng,cslzhang}@comp.polyu.edu.hk, csjyang@mail.njust.edu.cn

Abstract. Graph cuts based interactive segmentation has become very popular over the last decade. In standard graph cuts, the extraction of foreground object in a complex background often leads to many segmentation errors and the parameter λ in the energy function is hard to select. In this paper, we propose an iterated graph cuts algorithm, which starts from the sub-graph that comprises the user labeled foreground/background regions and works iteratively to label the surrounding un-segmented regions. In each iteration, only the local neighboring regions to the labeled regions are involved in the optimization so that much interference from the far unknown regions can be significantly reduced. To improve the segmentation efficiency and robustness, we use the mean shift method to partition the image into homogenous regions, and then implement the proposed iterated graph cuts algorithm by taking each region, instead of each pixel, as the graph node for segmentation. Extensive experiments on benchmark datasets demonstrated that our method gives much better segmentation results than the standard graph cuts and the GrabCut methods in both qualitative and quantitative evaluation. Another important advantage is that it is insensitive to the parameter λ in optimization.

Keywords: Image segmentation, graph cuts, regions merging.

1 Introduction

Interactive foreground/background segmentation is a practical and important problem in computer vision. Over the last decade, a number of interactive segmentation techniques have been proposed, such as snakes [1], livewire [2], level sets [3],watershed cuts [4] and random walkers [5]. Another preferable method which becomes very popular in recently years is graph cuts [6,7]. Graph cuts addresses segmentation in a global optimization framework and guarantees a globally optimal solution for a wide class of energy functions.

A number of recent publications further extend the pioneer work of Boykov and Jolly [6] and develop the use of regional cues [8,13] or various object

⋆ Corresponding author.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 677–686, 2010.

segmentation cues [14,15]. Lombaert et al. [9] studied the use of graph cuts for high-resolution data. They proposed a multilevel banded heuristic for the computation of graph cuts. The use of a smaller graph in all resolutions reduces the running time and memory consumption compared with the original graph cuts algorithm. Because the graph cuts technique can involve a wide range of visual cues, some researchers used the shape prior as an effective cue in the graph cuts framework. Freedman and Zhang [10] defined the shape prior as a single fixed template, which was specified as a distance function inspired by the idea of level sets. Das and Veksler [11] developed a graph cuts based segmentation algorithm by assuming the object is of compact shape. Further more, Veksler [12] exploited the star shape prior, which is a kind of generic shape prior, into graph cuts segmentation.

Although the user input is valuable in steering the segmentation process to reduce the ambiguities, too much interaction would lead to a tedious and time-consuming work. Usually, the extraction of foreground objects in a complex environment, from which the background can not be trivially subtracted, often requires a lot of user interaction. Moreover, the complex content of an image also makes it hard to give user guide for accurate segmentation while keeping the interaction as less as possible. Thus some algorithms allow the further user edit based on the previous segmentation result [8,22], yet this requires additional user interaction.

In this paper, we explore the graph cuts algorithm by extending it to a region merging scheme. Specifically, we perform mean shift [16] algorithm on the original image for an initial segmentation, which partitions the image into many homogenous regions. Starting from seeds regions given by the user, we run graph cuts on a propagated sub-graph where the segmented regions by mean shift algorithm, instead of the pixels in the original image, are regarded as the nodes of the graph. An iterated conditional mode (ICM) on graph cuts is studied and, whereas it does not provide a global solution in the whole graph, global optima can be obtained on the growing subgraphs.

Our method is a novel extension of the standard graph cuts algorithm. It has many advantages and merits. First, using sub-graph can reduce significantly the complexity of background content in the image. The many unlabeled background regions in the image may have unpredictable negative effect on graph cuts optimization. This is why the global optimum obtained by graph cuts often does not lead to the most desirable result. However, by using a sub-graph and blocking those unknown regions far from the labeled regions, the background interference can be much reduced, and hence better results can be obtained under the same amount of user interaction. Second, the algorithm is run on the sub-graph that comprises foreground/background regions and their surrounding un-segmented regions, thus the computational cost is significantly less than running graph cuts on the whole graph which is based on image pixels. Third, as a graph cuts based region merging algorithm, our method obtains the optimal segmentation on each sub-graph in the iteration. Forth, the object and background color models are

updated after the segmentation on each sub-graph. Thus they can provide more informative guide for the next round of segmentation.

The paper is organized as follows. A brief review of standard graph cuts algorithm is in Section 2. An iterated conditional mode on graph cuts is proposed in Section 3, followed by the iterated graph cuts algorithm. Section 4 presents experimental results of our method on 50 benchmark images in comparison with standard graph cuts and Grabcut. Finally the conclusion is made in Section 5.

2 Image Segmentation by Graph Cuts

Segmentation of an object from the background can be formulated as a binary labeling problem. Given a set of labels L and a set of sites S, the labeling problem is to assign a label $f_p \in L$ to each of the sites $p \in S$. The graph cuts framework proposed by Boykov and Jolly [6] addresses the segmentation of a monochrome image, which solves a labeling problem with two labels. The label set is $L = \{0, 1\}$, where 0 corresponds to the background and 1 corresponds to the object.

Let $f = \{f_p | f_p \in L\}$ stand for a labeling, i.e. label assignments to all pixels. An energy function is formulated as:

$$E(f) = \sum_{p \in S} D_p(f_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}} \omega_{pq} \cdot T(f_p \neq f_q) \tag{1}$$

On the right hand side of (1), the first term is called data term, which consists of constraints from the observed data and measures how sites like the labels that f assigns to them. where D_p measures how well label f_p fits site p. A common approach, and the one we use in our work, is to build the foreground and background histograms models from the user input seeds, respectively. Then the $D_p(f_p)$ are defined as the negative log likelihoods of the constructed foreground/background models.

The second term is called the smoothness term and measures the extent to which f is not piecewise smooth. where $\mathcal{N}$ is a neighborhood system, such as a 4-connected neighborhood system or an 8-connected neighborhood system. The smoothness term typically used for image segmentation is the Potts Model [20]. Here $T(f_p \neq f_q)$ is 0 if $f_p = f_q$ and 1 otherwise. This model is a piecewise constant model because it encourages labelings consisting of several regions where sites in the same region have the same labels.

In image segmentation, we want the boundary to lie on the edges in the image. A typical choice for $\omega_{p,q}$ is : $\omega_{pq} = e^{-\frac{(I_p - I_q)^2}{2\delta^2}} \cdot \frac{1}{dist(p,q)}$, where I_p and I_q are the color values of sites p and q, and $dist(p, q)$ is the distance between sites p and q. Parameter δ is related to the level of variation between neighboring sites within the same object. The parameter λ is used to control the relative importance of the data term versus the smoothness term. Minimization of the energy function can be done using the min-cut/max-flow algorithm as described in [6].

3 The Iterated Graph Cuts

3.1 Iterated Conditional Mode

Graph cuts technique provides a globally optimal solution to image segmentation; however the complex content of an image makes it hard to precisely segment the whole image all at once. The iterated conditional mode (ICM) proposed by Besag [21] is a deterministic algorithm which maximizes local conditional probabilities sequentially. It uses the "greedy" strategy in the iterative local maximization to approximate the maximal joint probability of a Markov Random Field (MRF). Inspired by ICM, we consider the graph cuts algorithm in a "divide and conquer" style: finding the minima on the sub-graph and extending the sub-graph successively until reach the whole graph. The proposed method works iteratively, in place of the previous one-shot graph cuts algorithm [6].

Let d_i be the observed data of site i, f_i be the label of site i and $f_{S-\{i\}}$ be the set of labels which is at the sites in $S-\{i\}$, where $S-\{i\}$ is the set difference. We sequentially assign each f_i by maximizing conditional probability $P(f_i|d_i, f_{S-\{i\}})$ under the MAP-MRF framework. Here we have two assumptions in calculating $P(f_i|d_i, f_{S-\{i\}})$. First, the observed data $d_1, \ldots, d_m$ are conditionally independent given f and that each d_i depends only on f_i. Second, f depends on labels in the local neighborhood, which is Markovianity, i.e. $P(f_i|f_{S-\{i\}}) = P(f_i|f_{N_i})$, where N_i is a neighborhood system of site i. Markovianity depicts the local characteristics of labeling.

With the two assumptions we have:

$$P(f_i|d_i, f_{S-\{i\}}) = \frac{P(d_i|f_i) \cdot P(f_i|f_{N_i})}{P(d)} \tag{2}$$

where $P(d)$ is a normalizing constant when d is given. There is:

$$P(f_i|d_i, f_{S-\{i\}}) \propto P(d_i|f_i) \cdot P(f_i|f_{N_i}) \tag{3}$$

where $\propto$ denotes the relation of direct proportion. The posterior probability satisfies:

$$P(f_i|d_i, f_{S-\{i\}}) \propto e^{-U(f_i|d_i, f_{N_i})} \tag{4}$$

where $U(f_i|d_i, f_{N_i})$ is the posterior energy and satisfies:

$$\begin{aligned} U(f_i|d_i, f_{N_i}) &= U(d_i|f_i) + U(f_i|f_{N_i}) \\ &= U(d_i|f_i) + \sum_{i' \in N_i} U(f_i|f_{i'}) \end{aligned} \tag{5}$$

$U(d_i|f_i)$ is the data term corresponding to function (1), and $\sum_{i' \in N_i} U(f_i|f_{i'})$ is the smoothness term which relates to the number of neighboring sites whose labels $f_{i'}$ differ from f_i. The MAP estimate is equivalently found by minimizing the posterior energy:

$$f^{k+1} = \arg\min_f U(f|d, f_N^k) \tag{6}$$

where f_N^k is the optimal labeling of graph nodes obtained in previous k iterations. The labeling result in each iteration is reserved for later segmentation. This process is done until the whole image is labeled.

3.2 The Iterated Graph Cuts Algorithm

In the original graph cuts 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, the segmentation result may not be smooth, especially along the edges. These problems can be solved by introducing some low level image segmentation techniques, such as watershed [17] and mean shift [16], to graph cuts. In [22], Li et al. used watershed for initial segmentation to speed up the graph cuts optimization process in video segmentation. In this paper, we choose to use mean shift for initial segmentation because it produces less over-segmentation and has better edge preservation than watershed. Fig.1(b) shows the mean shift initial segmentation of the image in Fig.1(a).

The initial labeling f^0 of graph cuts is given by a group of foreground/ background seeds from the user. Regions which have pixels marked as foreground are called foreground seed regions, while the regions with background seeds are thus called background seed regions. The initial sub-graph contains only seed regions. Start from the initial sub-graph, in the iteration only the adjacent regions to the previously labeled regions are added into the updated sub-graph. Running graph cuts algorithm on the updated sub-graph, an updated optimal segmentation is obtained. The iteration stops when all the region nodes are labeled as either foreground (i.e. object) or background.

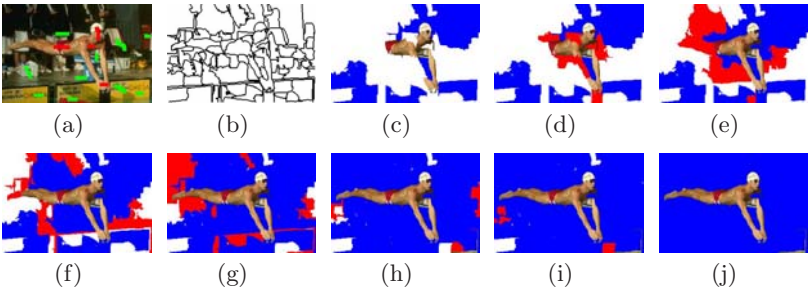


Fig. 1. The iterated segmentation process. (a) Original image with user input seeds. (b) Initial mean shift segmentation. (c) The user input seed regions. The background is shown in blue color. (d)-(i) show the intermediate segmentation results in the iteration. The newly added regions in the sub-graphs are shown in red color and the background is shown in blue color. In (j), the target objects are well segmented from the background within 6 iterations.

Fig.1 illustrates the iterated segmentation process. In the first iteration, regions chosen to be labeled are those which are only adjacent to the foreground regions, as shown in Fig.1(d). In the following iterations (Figs.2c-2h), new regions which are only in the neighborhood of previous foreground regions are added into the sub-graph for further labeling. In practice we have found that

adding regions which are adjacent to either the foreground or the background or both of them does not make much difference for the segmentation results. The desired objects are extracted as shown in Fig.1(j). The iterated graph cuts algorithm is summarized in Algorithm 1. We assume that the foreground regions are connected unless separated parts of the foreground are initially marked by the user. Therefore, the regions which can not be involved in the iterations will be labeled as the background regions.

Algorithm 1. Iterated GraphCuts.
The input are mean shift initial segmentation of the given image and a graph G whose nodes consist of the user input foreground/background seed regions R. The output is the segmentation result.

1. Add adjacent regions of foreground regions into G.
2. Construct foreground and background data models from seed regions R.
3. Use graph cuts algorithm to solve $\arg\min_f U(f|d, f_N^k)$.
4. Add foreground and background regions resulting from step 3 into R.
5. Add adjacent regions of the foreground seeds into G.
6. Go back to step 2, until no adjacent regions can be found.
7. Set labels of the remaining regions to be the background.
8. Return the segmentation result.

4 Experimental Results

In this section, we validate the segmentation performance of our method in comparison with the standard graph cuts algorithm [6] and GrabCut [8]. Since the proposed iterated graph cuts algorithm uses mean shift for initial segmentation, for a fair comparison we also extended the standard graph cut to a region based scheme, i.e. use the mean shift segmented small regions, instead of the pixels, as the nodes in the graph. Usually this yields better results than the original graph cuts. The GrabCut algorithm is an interactive segmentation technique based on graph cuts and has the advantage of reducing user's interaction under complex background. It allows the user to drag a rectangle around the desired object. Then the color models of the background and foreground are constructed according to this rectangle. Similarly to our method, an iterative estimation scheme of color models is used in GrabCut to segment the object.

We use the mean shift segmentation software- the EDISON System[1] -to obtain the initial segmentation. Experiments are performed on a database which contains 50 benchmark test images selected from online resources [2,3], where 10 of them contain objects with simple background and 40 are natural images with relatively complex background. Every image in our database has a figure-ground assignment labeled by human subjects.

[1] http://www.caip.rutgers.edu/ riul/research/code/EDISON/doc/segm.html
[2] http://www.research.microsoft.com/vision/cambridge/segmentation/
[3] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

4.1 Comparison with Standard Graph Cuts

We first compare the proposed iterated graph cuts with the standard graph cuts. In this subsection we use several example images to evaluate them qualitatively. The quantitative evaluation will be given in subsection 4.3. Fig.2 includes some images with simple background (Fig.2(a)-2(b)) and some with complex background (Fig.2(c)-2(e)). In the later ones, camouflage makes the objects containing weak boundaries due to poor contrast and noise, and the colors of some background regions are very close to those of the objects. Given the same amount of user input, the iterated graph cuts method achieves much better segmentation than standard graph cuts.

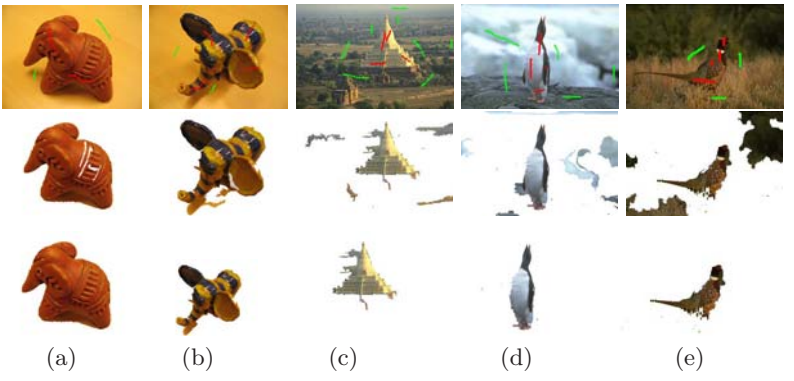


Fig. 2. Segmentation results of images with simple or complex background. The first row shows the original images with seeds. The second row shows the segmentation obtained by standard graph cuts. The third row shows the segmentation of iterated graph cuts.

4.2 Comparison with GrabCut

The ways of user input are different for GrabCut and Graph cuts. Graph cuts requires user to indicate some background and foreground regions, while GrabCut only needs the user to drag a rectangle around the object. In experiments, we choose the user inputs that lead to the best results for GrabCut.

An comparison with GrabCut is shown in Fig. 3. The first row shows the original images with the user inputs. The red and green seeds are for the proposed iterated graph cuts, while the blue rectangles are for the GrabCut. The second row shows the segmentation results of GrabCut. Implementation of GrabCut uses 5 GMMs to the model RGB color data and parameter λ is fixed to be 50. The third rows are results of iterated graph cuts. When the objects to be segmented contain similar colors with the background, GrabCut might fail to correctly segment them. Although overall graph cuts may use more user interaction than GrabCut, it can produce more precise segmentation results.

Fig. 3. Segmentation results of GrabCut and proposed iterated graph cuts. The first row shows the original images with seeds. Red and green strokes represent the object and background seeds for graph cuts. User inputs for GrabCut are denoted by blue rectangles. The second row shows the results of GrabCut . The third row shows the results of iterated graph cuts. The proposed method can segment more accurately the desired objects than GrabCut.

4.3 Quantitative Evaluation

Quantitative evaluation of the segmentations is given by comparing with ground truth labelings. The qualities of segmentation are calculated by using four measures: the true-positive fraction (TPF), false-positive fraction (FPF), true-negative fraction (TNF) and false-negative fraction (FNF), which are defined as follows:

$$TPF = \frac{|A_A \cap A_G|}{|A_G|}, FPF = \frac{|A_A - A_G|}{|\overline{A_G}|}, TNF = \frac{|\overline{A_A \cup A_G}|}{|\overline{A_G}|}, FNF = \frac{|A_G - A_A|}{|\overline{A_G}|}$$

where A_G represents the area of the ground truth of foreground and its complement is $\overline{A_G}$; A_A represents the area of segmented foreground by the tested segmentation method.

Table 1 lists the results of TPF, FPF, FNF and TNF by the three methods over the 50 test images. We see the iterated graph cuts method achieves the best FPF, TNF and FNF results. The GrabCut method has higher TPF index than iterated graph cuts because it usually leads to a bigger segmentation area, which includes both foreground and background. Thus it also has much higher FPF rate.

Table 1. The TNF, TPF, FNF and FPF results by different methods

Algorithms	TPF(%)	FPF(%)	TNF(%)	FNF(%)
GrabCut	93.88	16.35	96.59	16.35
Graph cuts	84.23	4.65	95.87	9.26
Ours	90.90	2.97	97.59	6.42

4.4 Discussion

In graph cuts based segmentation, parameter λ has great effect on segmentation results. It is used to tune the balance between different terms in the energy function. When given different images, a fixed value of λ can not give satisfactory segmentation. Since the appropriate λ values would vary largely among different images, the user may have to spend a significant amount of time searching for it. In the recent works [18,19], much effort has been made to study the selection of λ. From our experiments, parameter λ was easier to set up for our method and thus brings much benefit for users in real applications.

5 Conclusion and Future Work

An iterated graph cuts algorithm was developed in this paper. It performs segmentation on the sub-graph which is updated in each iteration. The proposed iterated graph cuts can reduce the interference of unknown background regions far from the labeled regions so that more robust segmentation can be obtained. Qualitative and quantitative comparisons with standard graph cuts and GrabCut show the efficiency of the proposed method. With the same amount of user input, the proposed method can achieve better segmentation results than the standard graph cuts, especially when extracting the foreground from complex background. Moreover, the search space of parameter λ can also be reduced by our method.

Standard graph cuts can be viewed as a special case of the proposed iterated graph cuts when there is only one iteration in segmentation and all regions are involved in the optimization. Future work will be focused on how to reduce its dependency on the initial segmentation result and how to reduce the user interaction while preserving the segmentation accuracy.

Aknowledgement

This work is supported by the Hong Kong RGC General Research Fund (PolyU 5351/08E) and partially supported by the Program for New Century Excellent Talents in University of China, the NUST Outstanding Scholar Supporting Program, and the National Science Foundation of China under Grants No. 60632050.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 2, 321–331 (1988)
2. Falaco, A.X., Udupa, J.K., Samarasekara, S., Sharma, S.: User-steered image segmentation paradigms: Live wire and live lane. Graphical Models and Image Processing 60, 233–260 (1998)
3. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithm based on hamilton jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)

4. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)
5. Grady, L.: Random Walks for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1768–1783 (2006)
6. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation. In: International Conference on Computer Vision, vol. I, pp. 105–112 (2001)
7. Boykov, Y., Funka Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision 69(2), 109–131 (2006)
8. Rother, C., Kolmogorov, V., Blake, A.: Grabcut-interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics 23(3), 309–314 (2004)
9. Lombaert, H., Sun, Y., Grady, L., Xu, C.: A multilevel banded graph cuts method for fast image segmentation. In: International Conference on Computer Vision, vol. 1, pp. 259–265 (2005)
10. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape prior. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 755–762 (2005)
11. Das, P., Veksler, O., Zavadsky, V., Boykov, Y.: Semiautomatic Segmentation with Compact Shape Prior. Image and Vision Computing 27(1-2), 206–219 (2009)
12. Veksler, O.: Star Shape Prior for Graph-Cut Image Segmentation. In: European Conference on Computer Vision, pp. 454–467 (2008)
13. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. In: IEEE Conference of Computer Vision and Pattern Recognition, pp. 407–414 (2005)
14. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: International Conference on Computer Vision, vol. I, pp. 26–33 (2003)
15. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: International Conference on Computer Vision, vol. I, pp. 564–571 (2005)
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
17. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(6), 583–598 (1991)
18. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of Parametric Maxflow in Computer Vision. In: International Conference on Computer Vision, pp. 1–8 (2007)
19. Peng, B., Veksler, O.: Parameter Selection for Graph Cut Based Image Segmentation. In: British Machine Vision conference (2008)
20. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with effcient approximations. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–655 (1998)
21. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, series B 48, 259–302 (1986)
22. Li, Y., Sun, J., Shum, H.: Video Object Cut and Paste. ACM Transactions on Graphics 24(3), 595–600 (2005)

Contour Extraction Based on Surround Inhibition and Contour Grouping

Yuan Li, Jianzhou Zhang, and Ping Jiang

College of Computer Science
Sichuan University, Chengdu, 610065, P.R. China

Abstract. Extraction of object contours from the natural scene is a difficult task because it is hard to distinguish between object contour and texture edge. To overcome this problem, this paper presents a contour extraction method inspired by visual mechanism. Firstly, a biologically motivated surround inhibition process, improved by us, is applied to detect contour elements. Then we utilize visual cortical mechanisms of perceptual grouping to propose a contour grouping model. This model consists of two levels. At low level, a method is presented to compute local interaction between contour elements; at high level, a global energy function is suggested to perceive salient object contours. Finally, contours having high energy are retained while the others, such as texture edge, are removed. Experimental results show our method works well.

1 Introduction

Edge detection, an important task in computer vision, is a fertile field of ongoing research. In the last two decades, many edge detection algorithms have been proposed including linear filtering [1], nonlinear diffusion [2], optimal edge detector [3], etc. Nearly most the detectors react to all local change of intensity, and don't distinguish between object contour and edge originating from textured region. Thus, for these edge detection models, it is extremely difficult to isolate objects in a cluttered scene. However, human visual system can rapidly and effectively perceive object contours.

Neurophysiological researches [4] [5] show that in early stages of visual information processing, the human visual system deploys a visual mechanism to discriminate between texture edge and object contour elements so that it can suppress texture edge and then relatively enhance saliency of object contour elements. Furthermore, effects produced by object contour elements are not independent in primary visual cortex. According to [6] [7], visual neurons in area V1 are interrelated and compose a whole functional network by lateral and vertical linkage, hence it can invoke interactions between these effects, and then based on information of the interaction form integrative perception for salient contour.

In this paper, we propose a computational method, inspired by the above-mentioned visual mechanisms, to extract contour from nature scenes. The paper is organized as follows. In Sect. 2, we utilize a method referred as surround suppression to detect elements of object contour. Then these local elements are grouped into some object contours in Sect. 3. In Sect. 4, experimental results are given.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 687–696, 2010.

2 Surround Inhibition Process

Receptive field is area in which stimulation leads to response of a particular sensory neuron, and the response of neuron cell to stimulus in receptive field can be suppressed by stimuli presented in surrounding regions. Neuroscientists refer to the visual effect as nonclassical receptive field inhibition. Further proof [8] shows that degree of surround suppression depends on the similarity of orientations and distance of stimuli inside and outside the RF.

2.1 Original Computational Method

We firstly introduce an original computational model of surround inhibition based on Grigorescu's model [9]. Some psychophysical findings show that inhibitory intensity declines with increasing distance to the center of the RF. The distance between stimuli inside and outside the RF is taken into account by a weighting function, and the difference of two Gaussian functions model is adopted to define the weigh tion $w_\sigma(x, y)$ as follows:

$$DoG_\sigma(x, y) = \frac{1}{2\pi(4\sigma)^2}\exp\left(-\frac{x^2+y^2}{2(4\sigma)^2}\right) - \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \quad (1)$$

$$w_\sigma(x, y) = \frac{1}{\|H(DoG_\sigma)\|_1} H\big(DoG_\sigma(x, y)\big). \quad (2)$$

$$H(z) = \begin{cases} 0, z < 0 \\ z, z \geq 0 \end{cases}. \quad (3)$$

where $||.||_1$ denote the L_1 norm. $w_\sigma(x, y)$ is illustrate as Fig. 1.

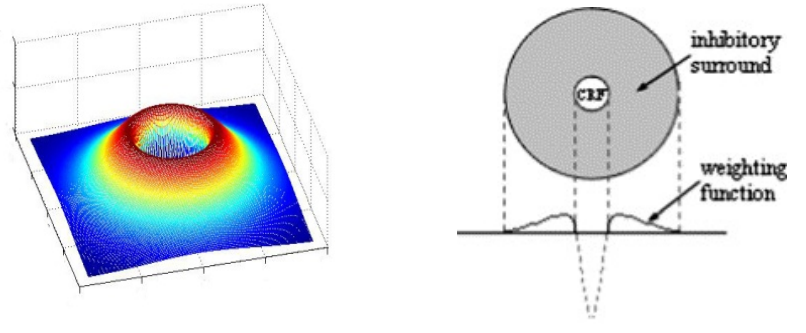


Fig. 1. The left is three-dimensional graph of weigh function $w_\sigma(x, y)$, and the right represents top view of $w_\sigma(x, y)$ and its profile

Kinerim and Van Essen's research [10] manifested that relative orientation of the centre and surround stimuli is also an important factor in determining inhibitory degree. The response to stimulus is suppressed significantly by similarly oriented stimuli in the surround. The inhibitory intensity is reduced when the orientation of the surround stimuli is different from that of the stimulus in the RF; the suppression is the strongest when they are same and is the weakest when they are orthogonal. To

describe the degree of suppression varying with the orientation similarity, an orientation contrast-based weighting function is given by:

$$w_\theta(\theta_\Delta) = 1 - \frac{\theta_\Delta}{\pi/2}. \tag{4}$$

Where $\theta_\Delta$denote orientation contrast.

The total inhibitory intensity of image point (x, y) is computed by weighted summation of the inhibitory effects in the inhibition surround, we use following equation to define it:

$$S_\sigma(x, y) = \sum_{(x\prime,y\prime)\in\Omega} w_\sigma(x' - x, y' - y) \times w_\theta(\theta_\Delta) \times m(x', y') . \tag{5}$$

Where m(x, y) is the gradient magnitude value of image, Ω is the domain that $w_\sigma(x' - x, y' - y)>0$ and represents inhibition surround of point (x, y).

$S_\sigma(x, y)$ is supposed to be large in textured areas and low on object contours thus leading to the suppression of texture while retaining contour elements.

Response of an image(x, y), after surround inhibition, is denoted as:

$$R_\sigma(x, y) = H(m(x, y) - \alpha \times S_\sigma(x, y)) . \tag{6}$$

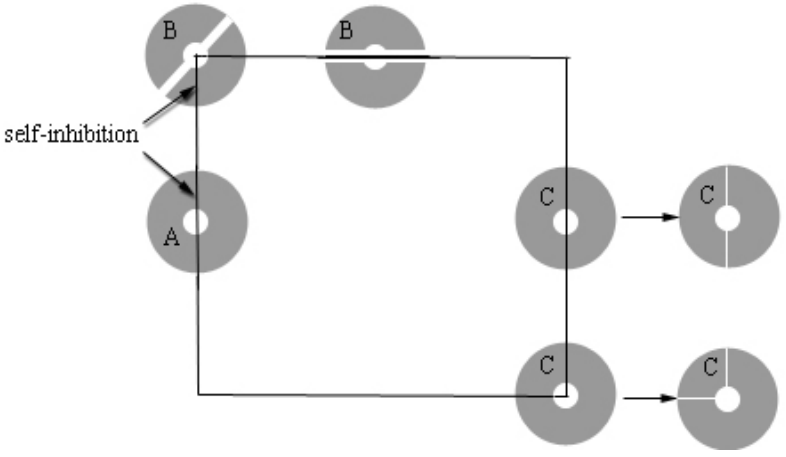


Fig. 2. Inhibition surrounds of different model are on a gradient image of square. These surrounds are represented by semitransparent shadow. A is surround in [9] and have self-inhibition on contour. B is surround improved by [11] and have self-inhibition on corner. C is surround in this paper and always avoids self-inhibition.

2.2 Improved Scheme

The above-mentioned method that calculates effect of surround inhibition derives from model proposed by [9], is considered as a conventional surround suppression model. However, the method has a drawback: according to (5), $S_\sigma(x, y)$ can't be zero on an isolated object contour, even is large: the point (x, y) is a point of a contour element while other parts of the same contour element shall fall in inhibition surround of the point, see surround A in Fig. 2. Consequently, it can bring about no small effect

of surround suppression, and then some contour elements with low gradient magnitude may be eliminated. The result isn't what we desire. This drawback is referred to as self-inhibition.

Afterwards, an improvement of the model is presented by [11] to aim to eliminate the drawback. Their idea is to exclude from the annular inhibition surround Ω of a point a band region of given width oriented along the contour, see surround B in Fig. 2. However, there still exist two disadvantages in the improvement: 1. the model expects that all local parts of contour should be inserted into the band region, such as surround B on a side of square in Fig. 2, thus the contour element is sure to be approximately straight, or it is still unable to avoid self-inhibition, such as surround B on a corner of square in Fig. 2; 2. the width of band region can't change dynamically, and choice of width is difficult: if it's not enough wide and the contour is thick, the band region can't contain the all local parts of contour and then self-inhibition exists yet; if too wide, inhibition surround become deformed seriously and the work loses original intention.

So we propose a scheme that improves the model [9]. The model modified by us isn't subjected to the self-inhibition drawback in all case. The notion is to extract from the annular inhibition surround Ω of the current responding point a local edge which the current responding point belong to, and make the points on the extracted local edge don't bring about inhibition effect to response of the current responding point:

$$S'_{\sigma}(x,y) = \sum_{(m,n)\in(\Omega-E)} w_{\sigma}(m,n) \times w_{\theta}(\theta_{\Delta}) \times m(x+m, y+n) \,. \tag{7}$$

E indicates a set of coordinate of the local edge points, these points with the responding point belong to the same local edge. In order to find the corresponding set E of the current responding point, we adopt region growing method to extract the local edge including the current responding point. Coordinate of points on the local edge form set E. The procedure base on gradient magnitude m(x, y):

1. Firstly observe the current responding point, if its gradient magnitude value lager than a specific Threshold q, we consider this point lie in an edge and then continue next step, or E is ∅ and the procedure is over.
2. Take current responding point as seed point, start with it and grow region by a given criterion. The Growing process is confined in outer circular of inhibition surround Ω of the responding point.
3. The procedure will stop when no more point satisfy the criterion. Finally, take set of coordinate of points on the region as set of E.

Become the processes base on gradient magnitude m(x, y), the region extracted is just the local edge that the current responding point belong to. Our computational method of surround inhibition is adaptive on the local edge. For instance, in Fig. 2, inhibition surround Ω of C on the corner automatically excludes the region of the local edge so that self-inhibition is eliminated absolutely.

Computation complexity of the model is not as high as it appears to be, because that the growing process merely occurs on the point considered as a point on an edge, and is confined in outer circular of inhibition surround. In addition, above all, there is a skill used in implementation of an algorithm: each time growing process can utilize the information of result of the last growing process if on same edge, set E of this process is almost identical with set E of last process, the only thing that needs to do in

process is to update the set E slightly. The skill is inspired by Huang's algorithm for median filter [12]. Thus computation complexity of the model can't be very high.

2.3 Thinning and Binarization

For the sake of necessity of following task, an additional step is to thin edge and generate binary representation. Through thinning by non-maxima suppression applied to $S'_\sigma(x, y)$ and binarization by hysteresis thresholding respectively, a binary map is obtained. Non-maxima suppression and hysteresis thresholding are introduced in canny operator (see [3] for details). In our paper, the two approaches aren't stated.

3 Contour Group

In this task, we aim at extracting contours of object. Although the above task has suppressed quite a few texture edges by surround suppression, it is inevitable that there exist a great deal of non-contour elements. These undesired non-contour elements have devastating effect on quality of our whole work, and many further tasks based on contour of object, such as shape-based object recognition, hardly carry on. So in this step, disorganized and unrelated local contour elements are grouped to form the organized global contours, which are retained as the final result. The contour-grouping model that we propose comprises two levels. At low level, we concentrate our attention on context interaction between contour elements in local. At high level, we suggest global contour perception method to perceive salient contours [13]. These ideas arise from early visual perception mechanisms [14]. The entire behavior of perceptual grouping is primarily bottom-up process.

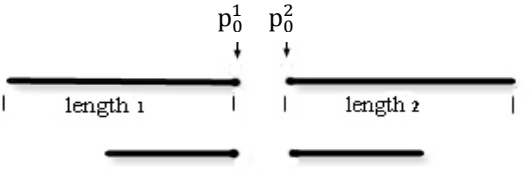


Fig. 3. Above two lines represent a pair of contour causing local interaction of proximity. Below two lines are the other pair, their local interaction of proximity is obviously weaker than the above.

3.1 Local Interactions of Contour Elements

Let's consider first the local interaction for contour elements. Through thinning and binarization, a contour element appears as line. We take two factors into account in this respect. The two factors are suggested according to parts of gestalt laws of perceptual organization (see work of C. Kanizsa [15] for details about gestalt theory).

Proximity. Law of proximity is one of gestalt laws, which plays an important role in contour grouping. It states that elements near each other tend to be grouped together. Interaction of proximity of a pair of contour elements is proposed as follow:

$$E_p = \frac{length_1 + length_2}{|p_0^1 - p_0^2|} . \quad (8)$$

Where p_0^1, p_0^2 represent two endpoints that are the closest each other for the pair of contour elements, as shown in Fig. 3, and $|p_0^1 - p_0^2|$ is the distance of the two points. $length_1$, $length_2$ are each length of the two contour elements.

In equation (8), the E_p that indicates interaction of proximity is defined by distance factor and length factor of contour element. It stands to reason that the distance is regarded as one of the factors because distance factor embodies proximity, while the length factor of contour element actually implies the scale factor. As illustrated in Fig. 3, the proximity of the above pair of contour elements superior to the below pair's obviously although their distance factors are identical.

Continuity. Law of continuity has also great influence for grouping contour. Continuity interaction of a pair of contour elements is suggested in the following way:

$$E_C = \sum_{i=1}(|p_i^1 - p_0^2| - |p_0^1 - p_0^2|)w_{\sigma^1}(i) + \sum_{j=1}(|p_j^2 - p_0^1| - |p_0^2 - p_0^1|)w_{\sigma^2}(j) . \quad (9)$$

Where p_i^1 represent the i.th nearest point to the endpoint p_0^1 in the one contour element and p_j^2 denotes the j.th nearest point to the endpoint p_0^2 in the other. The $w_\sigma(i)$ indicates a weighting function. Next, let's describe how continuity of two contour elements is represented by E_C in equation (9).

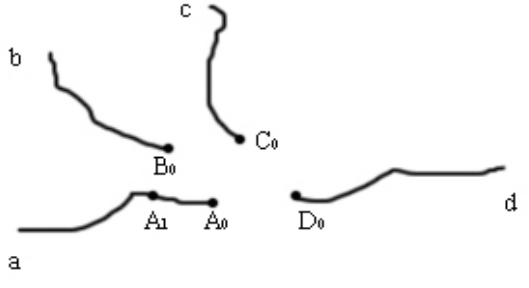


Fig. 4. Four lines (a, b, c, d) represent four contour elements. A_0, B_0, C_0, D_0 represent endpoints that are the closest each other for the four contour elements.

What model can represent continuity of a pair of contour elements has always been a noticeable problem in contour grouping. There are lots of algorithms in literature to deal with this problem. However, all these algorithms [16] [17] [18] that we have read are based on relationship of orientation of lines and just can be applied to calculate the continuity between straight lines. The reason is obvious: it is extremely difficult to define orientation of a curve line. As a matter of fact, contour elements of object can hardly appear as a straight line but a curve line. Therefore, these algorithms can't work practically, but our method isn't subjected to the limitation.

We use Fig .4 to illustrate relationship of E_C and continuity of two lines. In the Fig .4 (a, b, c, d indicate four contour elements in figure), it's so apparent for perceptual organization that the a and d are most likely to be perceived together, and the

continuity of the a and c is worse than of a and d but better than that of a and b. Through elaborate comparison and study, we find a law that these points subsequent to endpoint are the farther away from the endpoint of the other contour element, the better the continuity is. From Fig .4, a relation inequality can be drawn:

$$|A_1 - B_0| - |A_0 - B_0| < |A_1 - C_0| - |A_0 - C_0| < |A_1 - D_0| - |A_0 - D_0| . \quad (10)$$

Each term, such as $|A_1 - B_0| - |A_0 - B_0|$, actually correspond to the term $(|p_i^1 - p_0^2| - |p_0^1 - p_0^2|)$ of (9). The above description merely shows that relationship of E_C and continuity of a pair of contour elements, but doesn't deduce it. For subsequent points, their influence is different: the nearer point from its own side endpoint, the higher its weight is. So it's necessary to have weight function $w_\sigma(i)$ as coefficients.

Effect of local interaction. According to principle of local interaction, any two contour elements have their E_p and E_c even though their proximity and continuity are so bad that these two contour elements can't be grouped together. Thus we make an assumption that a pair of contour elements have effect of local interaction only when E_p, E_c are greater than their threshold values respectively.

3.2 Global Contour Perceptions

Contour perception is a global process that based on the information of local interaction [19]. In the process, local contour elements are grouped to form perception for salient contour. We suggest an energy function of global contour perception as follow:

$$E = \alpha \sum E_P + \beta \sum E_C - \gamma \cdot N + \sum_{(x,y)\in C} R(x,y) . \quad (11)$$

Any two adjoining contour elements on a perceptual contour must have effect of local interaction. $\sum E_P$, $\sum E_C$ represent proximity and continuity between contour elements on a contour. Each term in $\sum E_P$ corresponds to E_P of each two adjoining contour elements on a perceptual contour, as well as $\sum E_C$. N in equation (11) is the number of contour elements on a perceptual contour. $R(x, y)$ denotes response of an image(x, y) and C represents the contour. α, β and γ are factors that control influence, $(\alpha \sum E_P + \beta \sum E_C - \gamma \cdot N)$ in (11) is ensured to not larger than zero.

E represents the global saliency of the contour. Each contour's E can be obtained by backtracking traversal search method. The larger E a contour has, the more salient perceptually the contour is. Thus we retain a certain number of contours with great E as salient object contours, while plenty of unwanted textures are eliminated.

4 Experiment

Some experimental results are presented and discussed in this section. For choice of parameters of the model, σ is set to 1.6, and α is set to 2 in surround inhibition model, p in hysteresis thresholding is chosen to be 0.2. In Sect. 3, α, β and γ are set to 5,100,500. If $(\alpha \sum E_P + \beta \sum E_C - \gamma \cdot N)$ is larger than zero, we set it to be zero. $w_\sigma(i)$ in equation (9) is chosen to be standard normal function.

4.1 Experimental Results

The whole experimental process consists of surround inhibition and contour grouping. Firstly, surround inhibition method improved by us is used to detect contour elements

Table 1. Performance measures for the images presented in Fig. 5

	Near horse	Tree	Far horse
Canny	0.41	0.51	0.29
Surround inhibition[9]	0.40	0.51	0.34
Our model	0.52	0.55	0.43

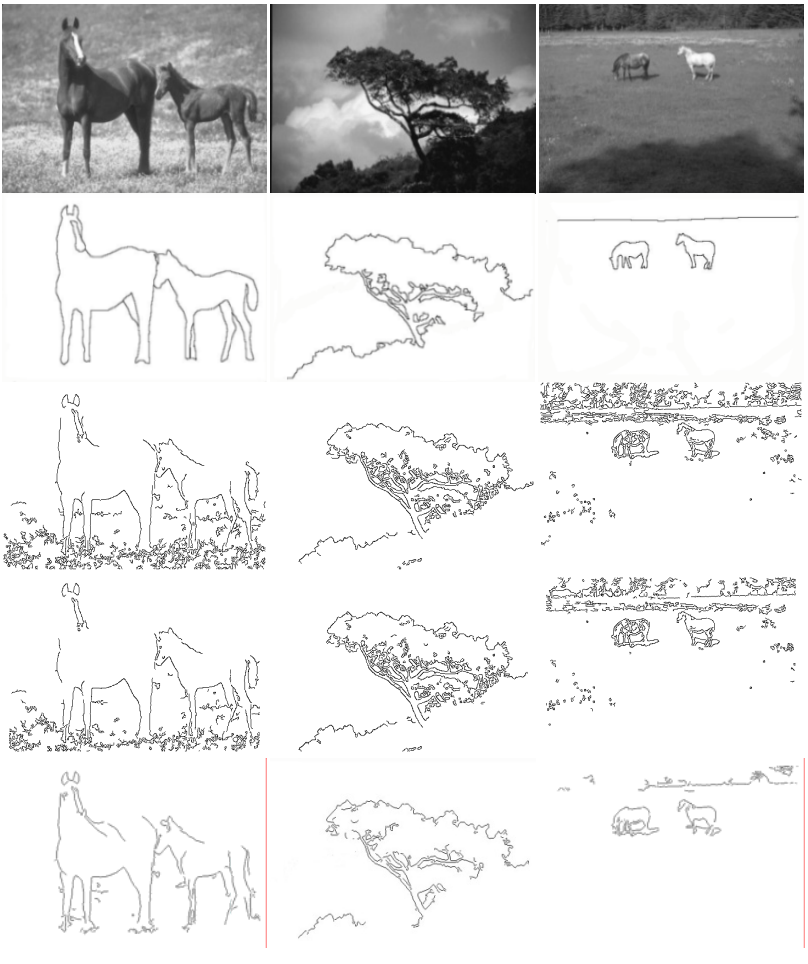


Fig. 5. The results of detecting contour with various operators. Panels from top to bottom correspond to input images, desired output, the contours obtained by canny operator [3], the contours obtained by surround inhibition in [9], and the contours obtained by our model.

(It is natural that the output also includes texture edge), and then group the local elements into global contours. Lastly, all candidate contours are ranked on energy value of global contour perception, and then select contours whose energy larger than a given threshold as final result. The choice of the threshold is a trade-off between the completeness of contours and the removal of texture edge.

We take some natural images from Berkeley images dataset to verify the capability of our model for contour extraction and compare the proposal with the surround inhibition operator by [9] and canny operator [3]. From experimental results of over twenty images, we select some typical cases are presented in Fig. 5, we can observe that our model can not just obtain nearly complete object contour, but can also eliminate almost all texture edge. Surround inhibition by [9] can remove more texture edge than canny operator, but still many texture edges remain in results due to lack of processes of Sect. 3. It is worth noting to the left case, the neck contour of horse in fourth row is inhibited by self and can't be retained in final result, while such failure never happen on our model.

Finally, we adopt the performance measure method proposed by [9] to evaluate the performance of experimental results. The method is introduced for detail in [9]. Table 1 shows the performance measures of experimental results shown in Fig. 5 and argues that our model are indeed better than the others on the application of extracting contours from natural image.

5 Conclusion

In this paper, we refer to some recent findings of neuroscience and psychophysics on visual perception to propose a bottom-up model of contour extraction. The model incorporates the two submodels: surround inhibition and contour group. As for surround inhibition, our method has property of self-adaptation, and can completely eliminate self-inhibition at the cost of a little additional computing time. For contour grouping, in local context, we suggest a computational method that is used to calculate interaction of a pair of contour elements with respect to proximity and continuity; in terms of global perception, we propose an energy function to perceive salient object contours. Our contour grouping model can be applicable to nature image, whereas the great majority models for contour grouping can only be used to synthetic image. Finally, experiment results show that our approach is feasible to extract contour from nature scene.

The present model still has plenty of scope for improvement. At further work, we adopt more image local and global feature and consider more psychophysical factors. Besides, human visual attention mechanism is also our further research work.

References

1. Marr, D., Hildreth, E.C.: Theory of edge detection. Proceedings of the Royal Society 207, 187–217 (1980)
2. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. PAMI 12(7), 629–639 (1990)
3. Canny, J.: A Computational Approach to Edge Detection. PAMI 8(6), 679–698 (1986)

4. Vinje, W.E., Gallant, J.L.: Natural Stimulation of the Nonclassical Receptive Field Increases Information Transmission Efficiency in V1. J. Neurosci. 22(7), 2904–2915 (2002)
5. Chen, G., Dan, Y., Li, C.Y.: Stimulation of non-classical receptive field enhances orientation selectivity in the cat. J. Physiol. 564(1), 233–243 (2005)
6. Grossberg, S., Mingolla, E., Ross, W.D.: Visual brain and visual perception: how does the cortex do perceptual grouping? Trends in neurosciences 20(3), 106–111 (1997)
7. Lee, T.S.: Computation in the early visual cortex. J. Physiol. 97(2), 121–139 (2003)
8. Walker, G., Ohzawa, I., Freeman, R.D.: Asymmetric Suppression Outside the Classical Receptive Field of the Visual Cortex. J. Neurosci. 19(23), 10536–10553 (1999)
9. Grigorescu, C., Petkov, N., Westenberg, M.A.: Contour and boundary detection improved by surround suppression of texture edges. Journal of Image and Vision Computing 22(8), 583–679 (2004)
10. Knierim, J.J., Van Essen, D.C.: Neuronal responses to static texture patterns in area V1 of the alert macaque monkey. J. Neurophysiol. 67(4), 843–856 (1995)
11. Papari, G., Campisi, P., Petkov, N.: A biologically motivated multiresolution approach to contour detection. Journal on Applied Signal Processing, 177–205 (2007)
12. Huang, T.S.: Two-Dimensional Signal Processing II: Transforms and Median Filters. Springer, Berlin (1981)
13. Saarinen, J., Levi, D.M., Shen, B.: Integration of local pattern elements into a global shape in human vision. PNAS 94, 8267–8271 (1997)
14. Von Der Heydt, R., Peterhans, E.: Mechanisms of contour perception in monkey visual cortex. I. Lines of pattern discontinuity. J. Neurosci. 9, 1731–1748 (1989)
15. Kanizsa, G.: Organization in Vision: Essays on Gestalt Perception. Praeger, New York (1979)
16. Field, D.J., Hayes, A., Hess, R.F.: Contour integration by the human visual system: evidence for a local 'association field'. Vision Res. 33(2), 173–193 (1993)
17. Li, Z.: A neural model of contour integration in the primary visual cortex. Neural Comput. 10, 903–940 (1998)
18. Ursino, M., La Cara, G.E.: A model of contextual interaction and contour detection in primary visual cortex. Neural Networks 17, 719–735 (2004)
19. Kovace, I., Julesz, B.: Perceptual sensitivity maps within globally defined visual shapes 370, 644–646 (1994)

Confidence-Based Color Modeling for Online Video Segmentation

Fan Zhong[1], Xueying Qin[2,⋆], Jiazhou Chen[1], Wei Hua[1], and Qunsheng Peng[1,⋆]

[1] State Key Lab. of CAD&CG, Zhejiang University,
Hangzhou, 310027, P.R. China
[2] Department of Computer Science, Shandong University,
Jinan, 250101, P.R. China

Abstract. High quality online video segmentation is a very challenging task. Among various cues to infer the segmentation, the foreground and background color distributions are the most important. However, previous color modeling methods are error-prone when some parts of the foreground and background have similar colors, to address this problem, we propose a novel approach of Confidence-based Color Modeling (CCM). Our approach can adaptively tune the effects of global and per-pixel color models according to the confidence of their predictions, methods of measuring the confidence of both type of models are developed. We also propose an adaptive threshold method for background subtraction that is robust against ambiguous colors. Experiments demonstrate the effectiveness and efficiency of our method in reducing the segmentation errors incurred by ambiguous colors.

1 Introduction

Extracting foreground object from image and video has been an active research topic for a long time[1,2,3,4,5,6]. In recent years, high quality online video segmentation has attracted more and more attention because of its potential applications in teleconferencing and augmented reality, etc. In these applications high quality segmentation that can be used for background substitution is desired.

In [3] the authors introduces an effective binocular segmentation method, but its application is limited due to the requirement of binocular inputs. The succeeding works are all for monocular segmentation with stationary background [4,5,6], which adopt color, motion and contrast as the main cues to infer segmentation. These cues are combined into an optimization framework that can be solved efficiently with max-flow/min-cut [7].

Color distribution of the foreground and background is the most important cue, which can be represented with global and per-pixel color models. The global model describes the global color distribution of foreground and background, and per-pixel model represents the background color distribution at the location of each pixel, which is in fact the background model be used for background subtraction [8]. As is well known, segmentation methods easy to produce inaccurate

⋆ These two authors are corresponding authors.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 697–706, 2010.

(a) (b) (c) (d)

Fig. 1. The error caused by ambiguous colors. (a) input image; (b) probability map produced by the global color model (a pair of GMMs learned according to the ground truth of (a)), the pixels with greater intensity are more likely to be foreground; (c) the background image. Note that the background image is incomplete (in the large white region), and the reason is explained in section 4; (d) the result of background subtraction. From (b)(d) one can find a lot of misclassified pixels due to ambiguous colors (in the red rectangles).

segmentation when foreground and background have similar colors. However, this problem gained little attention in previous works, in which it seems that the color modeling process is always safe. Fig.1 demonstrates that both global and per-pixel color models may introduce notable errors when ambiguous colors present.

In previous methods, the most often adopted global color model is the Gaussian Mixture Model (GMM). Generally, the global color model can be any classifier that can output probability, so besides GMM, other learning algorithms, including k-NN and SVM, can also be used to build the global color model (if speed is not considered). However, because there is no learning algorithm can avoid introducing errors, the output of global color model is not always trustworthy (Fig.1(b)). The same for the per-pixel color model, although many adaptive threshold methods were proposed for background subtraction, none of them is capable of dealing with ambiguous colors. Consequently, when the overlapped parts of foreground and background have similar colors, foreground may be misclassified as background (Fig.1(d)).

When multiple types of cues are jointly considered, the impact of different cues can be adjusted through their weights. In previous methods, however, the weights of each type of cues are uniform for all pixels, which implies that the predictions of color models are treated equally regardless their correctness. Since the case of every pixel may be different, with uniform weights it would be difficult to achieve the optimal combination of cues at every pixel. We therefore propose to assign each pixel an individual weight based on the confidence of color models at each pixel. In this way we can reduce the impact of incorrect predictions of color models by assigning them lower weights.

Notice that the confidence of prediction is in general not the probability of the predicted class because the latter can be seriously biased due to imperfect inductive biases [9]. A common misunderstanding about the probability is that if a color is ambiguous, a classifier would automatically assign it nearly equal probabilities of belonging foreground and background. This is not true for most

classifiers. Fig.1(b) shows the case of GMM. In fact, in this case most classifiers would classify input feature to be the class whose samples occur more often, which would definitely cause features of the other class be misclassified. In the domain of machine learning there are already some attempts to measure the confidence (or reliability) [9,10] and to design classifiers with controlled confidence [11]. Despite their solid theoretical foundation, they are yet not practical for our problem due to their large computational cost.

The main contribution of this paper can be summarized in three aspects. First, we demonstrate that traditional segmentation model based on uniform weights is error-pone in dealing with ambiguous colors, and then present an confidence-based segmentation model. Second, we propose efficient methods to measure the confidence of both global and per-pixels color models. Third, we introduce an adaptive threshold approach for background subtraction which is shown to be robust against ambiguous colors. Our work focuses on the problems caused by ambiguous colors, which have been noted for a long time but have not been solved yet.

The rest of this paper is organized as follows. Section 2 introduces our confidence-based segmentation model. Section 3 presents the proposed global (section 3.1) and per-pixel (section 3.2) color models capable of measuring confidence and estimating adaptive thresholds, as well as the method to determine the weights of each pixel (section 3.3). Section 4 presents our experimental results, and compares the proposed method with previous video segmentation methods. Finally, we conclude our method in section 5.

2 Confidence-Based Segmentation Model

Let $\mathbf{z} = (z_1, \cdots, z_i, \cdots, z_N)$ be an array of pixel color that represents the input image, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_i, \cdots, \alpha_N)$ be the corresponding segmentation result, where $\alpha_i \in \{0, 1\}$ is the state of the i-th pixel. The segmentation $\boldsymbol{\alpha}$ then can be obtained by minimizing the following energy function:

$$E(\boldsymbol{\alpha}) = \sum_i \dot{\omega}_i E_1(\alpha_i) + \lambda \sum_{(i,j)\in\mathcal{E}} E_2(\alpha_i, \alpha_j) \tag{1}$$

where E_1 is the data term measuring the cost under the assumption that the state of the i-th pixel is α_i, E_2 is the smooth term encoding our prior knowledge about the segmentation, and λ is a free parameter used to trade-off between the data and smooth terms. $\dot{\omega}_i$ is a weighting function encoding the confidence of data terms (in previous methods $\dot{\omega}_i \equiv 1$).

The smooth terms E_2 are not dependent on the color distributions, and we will focus on the data terms E_1 and the weighting function $\dot{\omega}_i$. E_1 is typically computed as the negative log of the foreground color model $p(z_i|F)$ and the background color model $p(z_i|B)$:

$$E_1(\alpha_i) = \begin{cases} -\log p(z_i|F) & \text{if} \quad \alpha_i = 1 \\ -\log p(z_i|B) & \text{if} \quad \alpha_i = 0 \end{cases} \tag{2}$$

Without loss of generality, $p(z_i|F)$ and $p(z_i|B)$ can be assumed to be normalized, that is, $p(z_i|F) + p(z_i|B) = 1$, by giving either of them we can determine both. For clarity we use $p(z_i)$ to denote the normalized (foreground) color model:

$$p(z_i) = \frac{p(z_i|F)}{p(z_i|F) + p(z_i|B)} \tag{3}$$

The color model can be used to describe the global color distribution of the foreground and background. Since the background is stationary, the background color at the location of each pixel can also be described with a distribution function. Therefore, we define $p(z_i)$ as the combination of the global and per-pixel color models:

$$p(z_i) = \ddot{\omega}_i p_*(z_i) + (1 - \ddot{\omega}_i) p_i(z_i) \tag{4}$$

where $p_*(z_i)$ is the normalized global color model, and $p_i(z_i)$ is the normalized per-pixel color model regarding the i-th pixel. $\ddot{\omega}_i$ is a weighting function to balance their effects.

The above model is an extension of the segmentation model used in [4]. The main difference is that in [4], $\dot{\omega}_i \equiv 1$ and $\ddot{\omega}_i \equiv c$ are uniform to all pixels, while in our model they may take different value at different pixels. By computing $\dot{\omega}_i$ and $\ddot{\omega}_i$ according to the confidence of corresponding terms, we can emphasize the impact of reliable cues while suppressing the impact of unreliable cues which may lead to incorrect segmentation, in this way the errors introduced by color modeling process can be greatly reduced.

3 Confidence-Based Color Modeling (CCM)

3.1 Global Color Model

We adopt Gaussian Mixture Model (GMM) to represent the global color distribution:

$$p_*(z_i|F) = \sum_{k=1}^{K^F} \pi_k^F N(z_i|\mu_k^F, \Sigma_k^F) \tag{5}$$

where $(\pi_k^F, \mu_k^F, \Sigma_k^F)$ are the parameters of the k-th component, and K^F is the number of Gaussian components. $p_*(z_i|F)$ is the global foreground color model. The global background color model $p_*(z_i|B)$ is defined similarly. $p_*(z_i|F)$ and $p_*(z_i|B)$ can be trained from the foreground and background training color set $\mathcal{S}^F$ and $\mathcal{S}^B$, respectively. After that the normalized global color model $p_*(z_i)$ can be computed easily by equation (3). The probability map in Fig.1(b) is in fact the visualization of $p_*(z_i)$ acquired in this way.

Nevertheless, the global color model obtained in the above way provides bare probability without confidence measurement. The confidence of $p_*(z_i)$ depends on both the quantity of ambiguous colors around z_i and the accuracy of GMM. Specifically, if in color space, z_i falls in the region of many ambiguous colors , or the color distribution in the neighborhood of z_i cannot be accurately described

(a) (b) (c) (d)

Fig. 2. Probability and confidence map. The input image is the same as in Fig.1. (a) detected misclassified pixels (the gray pixels); (b) confidence map of the global color model (visualized in Fig.1(b)), greater intensity implies higher confidence; (c) probability map produced with our per-pixel color model; (d) confidence map of (c).

with GMM, $p_*(z_i)$ should be of low confidence. However, these two conditions are hard to be evaluated in practice, here we propose a simple, yet effective method to measure the confidence.

Note that the training data sets $\mathcal{S}^F$ and $\mathcal{S}^B$ can be used to validate the learned global color model. A color s is misclassified by the learned model if s is a foreground sample ($s \in \mathcal{S}^F$) but $p(s|B) > p(s|F)$, or s is a background sample ($s \in \mathcal{S}^B$) but $p(s|F) > p(s|B)$. Let $\mathcal{S}^U$ denote the set of all misclassified colors in $\mathcal{S}^F$ and $\mathcal{S}^B$, then we can train an additional GMM $p_*(z_i|U)$ from $\mathcal{S}^U$. $p_*(z_i|U)$ is the probability of z_i be misclassified, larger $p_*(z_i|U)$ implies lower confidence of $p_*(z_i)$. If $p_*(z_i|U)$ is larger than both $p_*(z_i|F)$ and $p_*(z_i|B)$, z_i can be considered to be misclassified. Fig.2(a) illustrates the misclassified pixels detected in this way, which shows that our method successfully found out most misclassified pixels. Now we can compute the confidence of $p_*(z_i)$ as:

$$\mathcal{C}(\, p_*(z_i)\,) = 1 - \frac{p_*(z_i|U)}{p_*(z_i|F) + p_*(z_i|B) + p_*(z_i|U)} \tag{6}$$

where $\mathcal{C}(\,\cdot\,)$ is the confidence function. Fig.2(b) visualizes the confidence of $p_*(z_i)$. One can find that the confidence of pixels vary a lot, and the pixels of ambiguous colors are assigned much lower confidence.

3.2 Per-pixel Color Model

Per-pixel color model is in fact the background model, the maintenance of which has been studied much [8,12,13]. We don't plan to survey all of these methods due to space limitation; instead, we suppose that the background model at each pixel has available as a Gaussian distribution $N(z_i|\mu_i, \Sigma_i)$. The mean μ_i can be regarded as the background color at the location of the i-th pixel.

Given the background model, background subtraction can be accomplished by thresholding the difference of the current pixel color and corresponding background color. Specifically, the i-th pixel is regarded as background if $\| \ z_i - \mu_i \ \| < T_i$; otherwise it is regarded as foreground, where T_i is the threshold function. A popular way of computing T_i is to make it vary according to the covariance matrix:

$$T_i = \rho\sqrt{\mathrm{tr}(\Sigma_i)} \tag{7}$$

where ρ is a scale factor, and $\mathrm{tr}(\Sigma_i)$ is the trace of the covariance matrix Σ_i. This method can make T_i adaptive to system noise, but it does not consider ambiguous colors. When the overlapped parts of foreground and background have similar colors, the thresholds computed in this way may cause foreground pixels misclassified, as demonstrated in Fig.1(d). In order to solve this problem, the threshold function must take both noise and ambiguous colors into consideration.

Since the foreground object may move to anywhere, a background pixel can be occluded by any part of the foreground. To find out the safe threshold for background subtraction, we need to know the minimum distance d_i from the background color mean μ_i to all the foreground colors:

$$d_i = \min\{\parallel \mu_i - \mu_k^F \parallel | k = 1, \cdots, K^F\} \tag{8}$$

where μ_k^F is the mean of the k-th Gaussian component of the global foreground color model. We need not to check every foreground color samples to find out the minimum distance, which is not only costly but also sensitive to noise. After getting d_i we can define two threshold functions T_i^B and T_i^F:

$$T_i^B = \min(d_i/2, T_i) \quad T_i^F = \max(d_i, T_i) \tag{9}$$

and then the normalized per-pixel color model can be computed as:

$$p_i(z_i) = \begin{cases} 0 & \text{if } \parallel z_i - \mu_i \parallel < T_i^B \\ 1 & \text{if } \parallel z_i - \mu_i \parallel > T_i^F \\ \frac{\parallel z_i - \mu_i \parallel - T_i^B}{T_i^F - T_i^B} & \text{otherwise} \end{cases} \tag{10}$$

if μ_i is close to some foreground colors, d_i and T_i^B would be small, which prevents foreground pixels from being misclassified as background; on the contrary, if μ_i is far from all foreground colors, d_i and T_i^F would be large, which can suppress noise better than T_i. Fig.2(c) is the probability map produced by this method. Although it still contains some errors, it looks much better than that shown in Fig.1(d), which is produced with the threshold function T_i as in (7).

The confidence of the probability $p_i(z_i)$ is dependent on both its magnitude and the reliability of the background model $N(z_i|\mu_i, \Sigma_i)$, so we compute it as:

$$\mathcal{C}(\, p_i(z_i)\,) = \sqrt{e^{-\beta \mathrm{tr}(\Sigma_i)} * |2p_i(z_i) - 1|} \tag{11}$$

where β is chosen to be $(2 < \mathrm{tr}(\Sigma_i) >)^{-1}$, in which $< \cdot >$ denotes the expectation over all pixels. The background model becomes unreliable if it is polluted by foreground colors, in which case $\mathrm{tr}(\Sigma_i)$ is large and $p_i(z_i)$ would be assigned lower confidence. $|2p_i(z_i) - 1|$ would be 0 if $p_i(z_i) = 0.5$, which implies z_i has equal probability to be both foreground and background. Fig.2(d) is the confidence map computed in this way.

3.3 Optimal Combination

Once the confidence of the global and per-pixel color models is known, we can combine them according to the confidence so that the color model with higher

(a) (b) (c) (d)

Fig. 3. Segmentation results. (a) the combined probability map of global and per-pixel color models; (b) foreground obtained with both confidence and adaptive thresholds; (c) foreground obtained without using confidence ($\dot{\omega} \equiv 1, \ddot{\omega} \equiv 0.5$); (d) foreground obtained without using adaptive thresholds ($T_i^F = T_i^B = T_i$).

confidence can take greater effect. Since the two confidence functions $\mathcal{C}(p_*(z_i))$ and $\mathcal{C}(p_i(z_i))$ are both in the range of $[0, 1]$, they do not need to be re-scaled, and the weighting functions $\dot{\omega}_i$ and $\ddot{\omega}_i$ can be simply computed as:

$$\dot{\omega}_i = \frac{1}{2}(\mathcal{C}(p_*(z_i)) + \mathcal{C}(p_i(z_i))) \tag{12}$$

$$\ddot{\omega}_i = \frac{\mathcal{C}(p_*(z_i))}{\mathcal{C}(p_*(z_i)) + \mathcal{C}(p_i(z_i))} \tag{13}$$

$\dot{\omega}_i$ can be regarded as the confidence of the combined color model $p(z_i)$. If both global and per-pixel color models at pixel z_i are of low confidence, $\dot{\omega}_i$ would be small, and the corresponding data term is assigned low weights, then smooth term would dominate the state of the corresponding pixel. Fig.3(a) shows the combined probability map.

4 Experimental Results

In experiments we adopt the video segmentation data set from Microsoft I2I project[1]. The test environment is a computer with 2.2GHz CPU and 4G RAM. The algorithm is implemented in C++.

Implementation details: The data terms E_1 are computed with the proposed method, and the smooth terms E_2 are computed in the same way of [4]. Since the background image is not provided in the data set, we have to accumulate it in online phase. At the start the background model of all pixels are invalid, after segmenting a frame, the acquired background pixels are used to fill the hole of the background image, and other parts of the background image are also updated as in [4]. Henceforth, the background image we use is incomplete, as shown in Fig.1(c).

The global color model is trained in the initialization phase. In [4] the program is initialized with the background image. [5] proposes an automatic initialization

[1] http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm

Fig. 4. Visual comparison with BC [4]. *Top*: input frames; *Middle*: foreground obtained with BC; *Bottom*: foreground obtained with our method.

method, but it needs labeled videos to train the motion model. Since the background image is not available in our case, we simply initialize our program with the ground truth of the first frame. In practice the initialization method can be chosen freely according to the available information.

The segmentation result is finally obtained by minimizing equation (1) with min-cut [7], then the object boundary is smoothed to suppress flicking.

Computational cost: Our system can achieve a speed of 10 $\sim$ 12 fps for input image sequence of size 320×240. Most computational cost is spent on minimizing the energy equation. To measure the confidence and to compute the adaptive thresholds bring only a little more cost, which is about 12ms in the case of $K^U = K^F = 10$ (lookup table is used to accelerate the computation of the exponential function in GMM).

Effect of CCM: Fig.3 demonstrates the effectiveness of the proposed color modeling method. The input image is hard to be precisely segmented due to the large area of ambiguous colors. Fig.3(c) is the foreground obtained with uniform weights, in which the desktop is mis-segmented as foreground due to the error introduced by the global color model (Fig.1(b)). Fig.3(d) is the foreground obtained without using adaptive threshold. Since the shoulder of the person appears nearly the same as the desktop, it is misclassified as a part of background by the per-pixel color model (Fig.1(c)). By using both nonuniform weights and adaptive thresholds, our method can generate much better segmentation result (Fig.3(b)).

Comparison with other methods: Fig.4 provides some visual comparisons of our method with "Background Cut" (BC, [4]). Since the background image

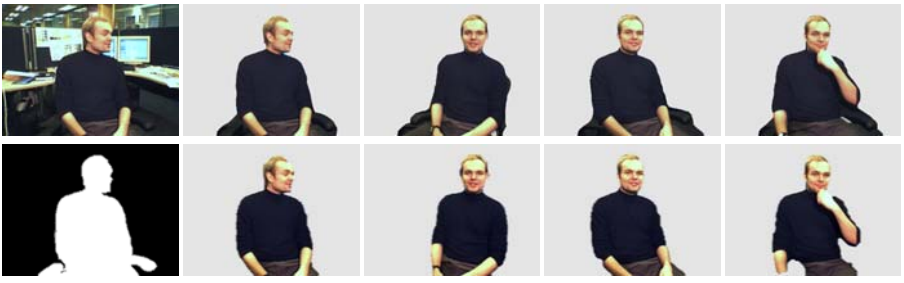


Fig. 5. Visual comparison with TM [5]. The first column is the input image and the ground truth of the frame #130.

Table 1. The error rates (%) of CCM, BC [4] and TM [5]

	JM	MS	AC	VK	50	54	56
CCM	0.13	1.40	0.47	0.68	1.12	0.39	0.33
BC	0.16	2.44	0.56	1.12	1.43	0.52	0.68
TM	0.12	2.59	0.52	-	-	-	-

is not available, our implementation of BC is not exactly the same as described in [4]. The only difference between our implementation of BC and our method exists in the modeling of color distributions, i.e. the computation of E_1 and its weights, so the comparison between them is fair.

Fig.5 is the comparison with [5] (TM), which involves Temporal and Motion priors as its cues. The results of TM are extracted from the published video, so fair comparison is not guaranteed. Tab.1 lists the error rates of CCM, BC and TM. Notice that the ground truth is available only every 5 or 10 frames, so not every frame are evaluated and the error rates may not capture all errors.

In fact, the implementation of our method in this experiment is a version of BC boosted with the proposed color modeling method. Since our color modeling method is independent of how the program is initialized and how other energy terms are computed, it can also be used to boost the performance of any other video segmentation methods that adopt color distribution as segmentation cues.

5 Conclusions

In this paper we propose a confidence-based color modeling method to improve the robustness of online video segmentation against ambiguous colors. A new confidence-based segmentation model is presented, which assigns energy terms nonuniform weights based on their confidence. We developed methods for measuring the confidence of both global and per-pixel color models, and for computing adaptive thresholds for background subtraction. The confidence is then

used to determine the weights of color models and energy terms at each pixel in order for the optimal combination of cues.

Experiments show that the proposed method can greatly enhance the segmentation result, especially for frames with large amount of ambiguous colors present. Our method to measure the confidence is very fast, and brings only a little more computational cost.

The limitation of our work is that it accounts for only ambiguous colors. Besides this, the change of lighting conditions, shadowing and camera shaking, etc. can also lead to errors in the color modeling process. Our future work is to address these problems in the confidence-based framework.

Acknowledgments. This paper is supported by 973 program of china (No. 2009CB320802) and NSF of China (No. 60870003).

References

1. Chuang, Y.-Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proceedings of CVPR, pp. 264–271 (2001)
2. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics 23, 309–314 (2004)
3. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Bi-layer segmentation of binocular stereo video. In: Proceedings of CVPR, pp. 407–414 (2005)
4. 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)
5. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: Proceedings of CVPR, pp. 53–60 (2006)
6. Yin, P., Criminisi, A., Winn, J., Essa, I.: Tree-based classifiers for bilayer video segmentation. In: Proceedings of CVPR (2007)
7. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 359–374 (2004)
8. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of CVPR, vol. 2, pp. 252–259 (1999)
9. Kukar, M., Kononenko, I.: Reliable classifications with machine learning. In: International Conference on Machine Learning (ICML), pp. 219–231 (2002)
10. Nouretdinov, I., Melluish, T., Vovk, V.: Ridge regression confidence machine. In: International Conference on Machine Learning (ICML), pp. 385–392 (2000)
11. Li, M., Sethi, I.K.: Svm-based classifier design with controlled confidence. In: International Conference on Pattern Recognition (ICPR), pp. 164–167 (2004)
12. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallfower: Principles and practice of background maintenance. In: International Conference on Computer Vision, pp. 255–261 (1999)
13. Mahadevan, V., Vasconcelos, N.: Background subtraction in highly dynamic scenes. In: Proceedings of CVPR (2008)

Multicue Graph Mincut for Image Segmentation

Wei Feng[1], Lei Xie[2], and Zhi-Qiang Liu[1]

[1] Media Computing Group, School of Creative Media,
City University of Hong Kong, Hong Kong, China
{weifeng,zq.liu}@cityu.edu.hk
[2] Audio, Speech and Language Processing Group,
Shaanxi Provincial Key Lab of Speech & Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
lxie@nwpu.edu.cn

Abstract. We propose a general framework to encode various grouping cues for natural image segmentation. We extend the classical Gibbs energy of an MRF to three terms: *likelihood energy*, *coherence energy* and *separating energy*. We encode *generative cues* in the likelihood and coherence energy to ensure the goodness and feasibility of segmentation, and embed *discriminative cues* in the separating energy to encourage assigning two pixels with strong separability with different labels. We use a self-validated process to iteratively minimize the global Gibbs energy. Our approach is able to automatically determine the number of segments, and produce a natural hierarchy of coarse-to-fine segmentation. Experiments show that our approach works well for various segmentation problems, and outperforms existing methods in terms of robustness to noise and preservation of soft edges.

1 Introduction

Image segmentation is a classical problem in low-level vision and plays an important role in many high-level applications [1,2]. Although many thoughtful ideas have been attempted, it is still challenging to find a unified solution that produces satisfactory segmentations for various natural images. The difficulty mainly results from the complexity of natural images. As shown in Figure 1, (1) natural images often contain both textured and untextured regions, which may cause massive false alarmed edges in textured regions; (2) natural images may have significant clutter and camouflage between foreground and background, where some perceptually important weak boundaries are easy to be missed.

To handle these problems, integrating multiple cues is a must choice. Several recent works have demonstrated the potentials of multicue image segmentation under these difficult situations. One success line of effort is the multiscale segmentation. For example, Sharon *et al.* proposed a recursive graph coarsening method to produce irregular image pyramid and used region-based cues at multiple scales to conduct the image graph partition [4]. Yu discussed the application of complementary multiscale edges in natural image segmentation, and embedded the multiscale cues within the average cuts scheme [5]. Benezit *et al.* presented a multiscale

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 707–717, 2010.

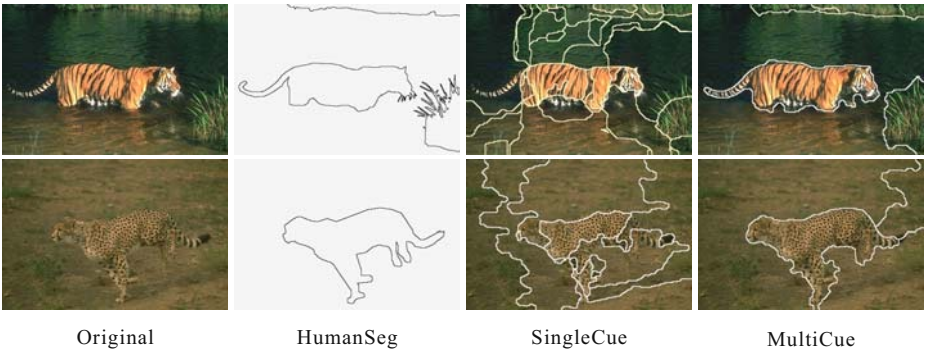


Fig. 1. Segmentation using single-cue vs. multicue graph mincut. The 1st row lists original images. Human labeled segment boundaries [3] are shown in the 2nd row. The 3rd row is segmentations using only color cue; and the 4th row shows the results of our approach using color, texture and multiscale edge cues. Note that multicue graph mincut groups the large torso and thin tail together, and separates both textured and untextured regions satisfactorily.

spectral decomposition algorithm to improve the efficiency of large image graph partition with long-range edges [6]. To cope with both texture and weak edges, Malik *et al.* modeled texture with textons [7] and then learned a probability model to combine local brightness, color and textures to detect image boundaries [8].

The necessity and potentials of multicues in solving difficult image segmentation problems is well recognized. The major differences are what cues should be used and how to combine multiple grouping cues that is computationally efficient and guarantees satisfactory segmentation for a wide range of natural images. Most previous methods focus on some particular cues and mainly use the normalized cut criterion [9] that forms a discriminative model. We would like to show in this paper that, various grouping cues, which can be divided to either generative cues (such as color, texture etc.) or discriminative cues (such as edge and color gradient etc.), can be encoded within a unified framework.

Our approach can be viewed as a combination of global generative model and local discriminative model. The segmentation is formulated as a graduated graph mincut process, each iteration of which is computationally fast and guarantees global optimum. Moreover, unlike most existing methods, our approach is able to automatically determine the number of segments and naturally corresponds to a coarse-to-fine segmentation. In the following, we first elaborate the details of multicue graph mincut framework, then demonstrate experimental results using color, texture and multiscale edge cues and compare with the state of the art.

2 Multicue Graph Mincut Framework

Image segmentation is more than just grouping pixels with homogeneous appearance, the spatial coherence is another important factor that should be

considered. This makes the Gibbs energy minimization, which takes care of the balance between segmentation goodness and spatial coherence, a natural formulation of segmentation problems [10]. Moreover, it has been shown that, given a fixed number of segments, the Gibbs energy minimization is equivalent to the *maximum a posteriori* (MAP) estimation of Markov random field (MRF) [11].

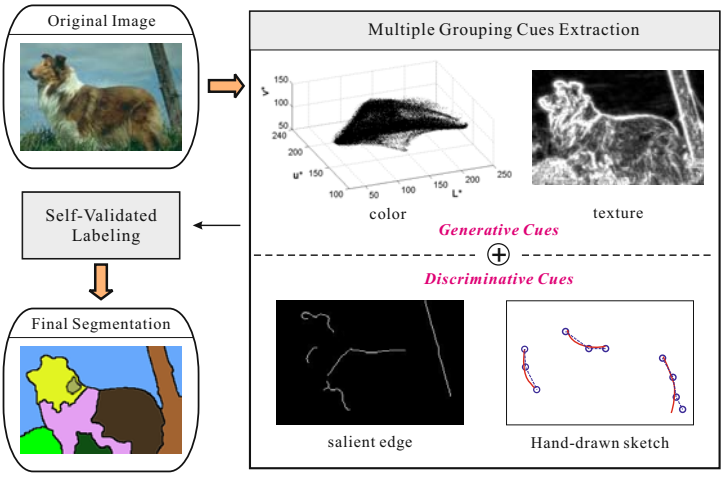


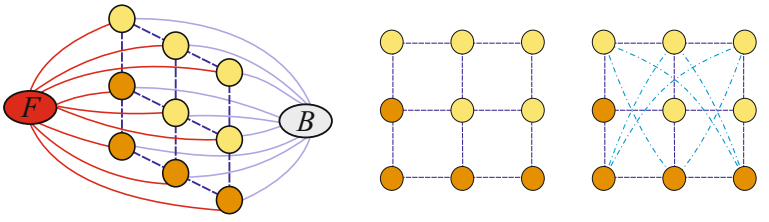
Fig. 2. General working flow of the MCGC framework

In practice, one successful model is the relational graph formulation [12]. Particularly, let $\mathcal{G}(I) = \langle \mathcal{V}, \mathcal{E} \rangle$ denote the graph of image I. Each vertex in $\mathcal{V}$ corresponds to a pixel in I and the arc set $\mathcal{E}$ connecting adjacent pixels represents the pairwise coherence. Let $Y = \{Y_s\}$ be the feature space of I with s denoting a pixel. The MRF $X = \{x_s\}$ depicts the segmentation results that considers both accuracy and contextual constraints, where $x_s \in \mathcal{L}$ and $\mathcal{L}$ is the label space. The goal of segmentation is to estimate the optimal label for each pixel with the minimum overall Gibbs energy [10].

Under the graph formulation, we may consider image segmentation as a *weighted graph multiway mincut* problem. Specifically, graph mincut is a natural way for binary Gibbs energy minimization. Unlike the discriminative model such as normalized cut [9] considering only partitioning likelihood, graph mincut encodes both likelihood and feasibility of a segmentation. For binary labeling problems, graph mincut guarantees global optimum with polynomial complexity [10]. In the following, we will generalize binary graph mincut to solve self-validated labeling problems, i.e., the multiclass labeling problem when the number of labels is unknown.[1]

Since choosing optimal features is task-dependent and remains a crucial problem for different vision and recognition tasks, in this paper, we attempt at finding

[1] Self-validated labeling problem refers to multi-class labeling of an image where the number of labels is unknown.



(a) (b) (c)

Fig. 3. Image graph formulation: (a) binary graph mincut model, where F and B represents the two cluster models (either parametric or nonparametric) and the curves linking cluster models to image grid encode the likelihood energy, and the pairwise edges between image pixels represent the spatial coherence energy; (b) is the classical pairwise linkage configuration of graph mincut [12], while (c) is the long-range edge embedded pixel linkage configuration (see the dot-line curves).

a general way to encode various grouping cues for segmentation. In particular, we will solve the following two problems:

1. How to encode various cues in the graph formulation?
2. How to efficiently solve the multiway graph mincut problem when the number of segments is unknown?

For complex natural images, different image features may provide complementary grouping cues (see Fig. 1). In general, a large variety of features in both spatial and frequency domain can be considered for image segmentation, but according to the their functionality in segmentation, all low-level features fall into the following two categories:

1. *Generative feature* G that represents some properties of single pixels, such as brightness, color and texture etc.,
2. *Discriminative feature* D that indicates the interrelationship between two or more pixels, such as edges and gradients etc.

A unified multicue segmentation should be able to encode both generative and discriminative features. Thus, for the feature space Y of image I, we have $Y = \{G_s, D_s\}$. That is, for each pixel s, the feature vector Y_s is composed of m generative features $\{G_s^1, \ldots, G_s^m\}$ and n discriminative features $\{D_s^1, \ldots, D_s^n\}$. Although we cannot enumerate all possible low-level features, we may propose a unified multicue segmentation framework that is able to encode both generative and discriminative features.

2.1 Multicue Embedded Gibbs Energy

We first consider the problem of binary segmentation problem that wants to label all pixels with a binary MRF $X^b = \{x_p\}$ $(x_p \in \{0, 1\})$ based on the feature space Y. The optimal labeling corresponds to the minimum of the following energy function:

$$E_{bMRF}(X^b) = \sum_{p\in\mathcal{V}} E_{lik}(x_p) + \sum_{(p,q)\in\mathcal{E}} E_{coh}(x_p, x_q), \tag{1}$$

where $E_{lik}(x_p)$ is the likelihood energy representing the goodness of labeling pixel p by x_p, and $E_{coh}(x_p, x_q)$ is the coherence energy denoting the prior of spatial coherence and feasibility of labeling. Note that (1) is the general form of Gibbs energy for binary MRF based image labeling [11] and can be exactly solved by graph mincut with polynomial complexity [12].

To embed both generative and discriminative features, we extend the image graph to $\mathcal{G}(I) = \langle \mathcal{V}, \mathcal{E} \cup \mathcal{E}_{edge} \rangle$, where $\mathcal{E}_{edge}$ represents the arcs that embeds a discriminative cue. For simplicity, we discuss embed only one discriminative cue, i.e., $n = 1$, but the proposed model is readily extended for multiple discriminative cues. Accordingly, the Gibbs energy (1) is extended to three terms

$$E_{bMRF}(X^b) = \sum_{p\in\mathcal{V}} E_{lik}(x_p) + \sum_{(p,q)\in\mathcal{E}} E_{coh}(x_p, x_q) + \sum_{(u,v)\in\mathcal{E}_{edge}} E_{sep}(x_u, x_v), \tag{2}$$

where $E_{sep}(\cdot)$ is the separating energy defined by

$$E_{sep}(x_u, x_v) = -\frac{|x_u - x_v|}{\|u - v\|} \exp\big(-\frac{1}{\gamma} E_d(u, v)\big), \tag{3}$$

where $\|u-v\|$ is the distance between pixel u and v, E_d is the concrete separating energy for the discriminative cue. Note that E_{sep} is a negative energy and larger $|E_{sep}|$ encourages separating u from v. γ control the influence of E_{sep}, the larger γ is, the more important is E_{sep} in E_{bMRF}. Through E_{sep}, we repel the pixels with large separability to have different labels, which can be viewed as a complement to the classical coherence energy. $E_{coh} + E_{sep}$ consists the pairwise terms of the classical Gibbs energy [11]. It is easy to show that E_{sep} satisfies the regular condition defined in [10], thus it can be solved by graph mincut.

To solve binary segmentation defined in (2), we need also define the concrete form of E_{lik} and E_{coh}. To realize this, we first derive a nonparametric two-level characterization of generative features:

Level 1. Dividing all feature samples into two *components* $\mathcal{C}^0$ and $\mathcal{C}^1$.
Level 2. Further dividing $\mathcal{C}^0$ and $\mathcal{C}^1$ into H *subcomponents* $\mathcal{M}_k^0$ and $\mathcal{M}_k^1$.

This is a nonparametric representation of the generative feature space, i.e., $G = \{\mathcal{C}^0, \mathcal{C}^1\} = \{\{\mathcal{M}_k^0\}_{k=1}^H, \{\mathcal{M}_k^1\}_{k=1}^H\}$, based on which the feature distance can be measured using the nearest neighbor criterion. In our experiments, H was empirically set as 2. Both the components and subcomponents are obtained by the K-means algorithm. Then, we can define both E_{lik} and E_{coh} in a nonparametric form. Since the discriminative features only appear in E_{sep}, for simplicity, in the definitions of E_{lik} and E_{coh}, the feature vector y_p denotes only generative features.

$$E_{lik}(x_p) = \Big(\frac{d_p^1 x_p + d_p^0(1 - x_p)}{d_p^1 + d_p^0}\Big)^{\alpha}, \tag{4}$$

where $d_p^0 = D(y_p, \mathcal{C}^0)$ and $d_p^1 = D(y_p, \mathcal{C}^1)$ represent the distance between y_p and the components $\mathcal{C}^0$ and $\mathcal{C}^1$ respectively. Specifically, $D(y_p, \mathcal{C}^c) = \min_k D(y_p, \mathcal{M}_k^c)$, where $\mathcal{M}_k^c$ the k-th subcomponent of $\mathcal{C}^c$. The distance $D(y_p, \mathcal{M}_k^c)$ between y_p and subcomponent $\mathcal{M}_k^c$) is defined as the minimal difference measured by color or texture:[2]

$$D(y_p, \mathcal{M}_k^c) = \min\big(\|C_p - C(\mathcal{M}_k^c)\|, \|T_p - T(\mathcal{M}_k^c)\|\big). \tag{5}$$

In (4), α controls the influence of likelihood energy E_{lik} on the Gibbs energy E_{bMRF}. The larger α is, the more important is the likelihood energy E_{lik} in E_{bMRF}. E_{coh} is defined as the modified Potts model:

$$E_{coh}(x_p, x_q) = \frac{|x_p - x_q|}{\|p - q\|} \exp\big(-\frac{1}{\beta}\|y_p - y_q\|\big). \tag{6}$$

Our definition of E_{coh} encourages close sites or sites with similar features to have the same label. In (6), β controls the influence of the coherence energy E_{coh} on the Gibbs energy E_{bMRF}. The larger β is, the more role the coherence energy E_{coh} plays in E_{bMRF}.

Proposition 1 (Regularity). *With the definition of E_{coh} (6) and E_{sep}, the general Gibbs energy (2) embedding multiple cues satisfies the regularity condition [10], thus can be exactly solved by graph mincut (see Fig. 3(a)) with polynomial time.*

Proof: It is easy to check that the general Gibbs energy (2) still belong to the energy class $\mathcal{F}^2$ [10]. We can write its second-order term as $E^{p,q}(x_p, x_q) = (\frac{|x_p - x_q|}{\|p-q\|})^\rho \cdot \exp(-\frac{\rho}{\beta}\|y_p - y_q\|)$, where $\rho = \gamma E_d(p,q) + 1$. Note that the second part of $E^{p,q}(x_p, x_q)$ lies in $(0, 1]$ because $1 \le \rho \le \gamma + 1$. Thus, $E^{p,q}(0,0) = E^{p,q}(1,1) = 0$ and $E^{p,q}(0,1) = E^{p,q}(1,0) \ge 0$. Therefore, $E^{p,q}(0,0) + E^{p,q}(1,1) \le E^{p,q}(0,1) + E^{p,q}(1,0)$ and the general Gibbs energy (2) satisfies the regularity criterion [10]. □

With (2)-(6) and Proposition 1, we can exactly solve multicue binary segmentation using the graph mincut algorithm approach.

2.2 Self-validated Segmentation

Graph mincut has been successfully used in binary labeling problems, such as interactive cutout [12]. We show here how to use the split-and-merge strategy to gradually approach a self-validated segmentation [13].

The major difficulty of MRF/MAP based methods is that it cannot simultaneously achieve global optimization and computational efficiency. Moreover, when the number of labels is also unknown, the problem becomes more difficult. Our method is based on the observation that there is no way to produce a "perfect" segmentation solely based on low-level cues. Therefore, the generic global optimal segmentation is neither possible nor necessary. Rather than finding a single

[2] This is because the color and texture features are usually complementary to each other. In textured regions, the color distance may be quite large; and vice versa. Think, for instance, the texture of zebra and its color.

"perfect" segmentation, we prefer a hierarchical coarse-to-fine segmentation. We formulate these ideas by converting a K class segmentation problem to a series of binary segmentation problems, which are much simpler and can be exactly solved by graph mincut algorithm. Hence, we call our approach *graduated graph mincut*.

We start with the whole feature space Y and model it with a binary MRF X_0^b. The solution of X_0^b with minimal energy results in two tentative segments, X_1^b and X_2^b. We continue this process iteratively until the whole energy stops decreasing. For each iteration, a tentative segment X_i^b evolves itself from three hypotheses: *remaining*, *merging* or *splitting*. The evolution criterion is Gibbs energy minimization (2). We can see that the iterative segment evolving process is terminated when all tentative segments remain unchanged. We need not specify the number of segments that can be implicitly determined by the segment evolution process, thus this approach is self-validated. Furthermore, since optimizing a binary MRF is much easier than the flat K-class MRF, this approach is much faster than the classical MRF methods.

For each tentative segment X_i^b, we first extract the two-level feature space representation $\langle \mathcal{C}^{0,i} | \mathcal{M}_k^{0,i} \rangle$ and $\langle \mathcal{C}^{1,i} | \mathcal{M}_k^{1,i} \rangle$. Then we choose the optimal hypothesis based on the energy of remaining, merging and splitting.

1. *Remaining energy.* The likelihood energy (i.e., intra-class distance) of X_i^b is measured as the average likelihood energy of all samples in X_i^b:

$$E_{remain}(X_i^b) = \frac{\sum_{s \in \mathcal{V}(X_i^b)} E_{lik}(c_s)}{|\mathcal{V}(X_i^b)|}, \tag{7}$$

where $\mathcal{V}(X_i^b)$ is the pixel set of X_i^b, $|\mathcal{V}(X_i^b)|$ is its size, and c_s is the component label of site s in X_i^b: $c_s = 0$ if $d_s^0 \leq d_s^1$, otherwise $c_s = 1$.

2. *Merging energy.* For a segment X_i^b, its remaining energy denotes its likelihood of remaining unchanged. Similarly, the merge energy between segments X_i^b and X_j^b can be defined as the remaining energy of $X_i^b \cup X_j^b$ (i.e., the union of X_i^b and X_j^b):

$$E_{merge}(X_i^b \cup X_j^b) = E_{remain}(X_i^b \cup X_j^b), \tag{8}$$

where $X_i^b \cup X_j^b$ is also a binary MRF with corresponding components ($\mathcal{C}^{0,i \cup j}$ and $\mathcal{C}^{1,i \cup j}$) and subcomponents ($\mathcal{M}_k^{0,i \cup j}$ and $\mathcal{M}_k^{1,i \cup j}$).

3. *Inter-segment energy.* To make the remaining energy and merging energy comparable, we also need the inter-segment energy:

$$E_{int}(X_i^b, X_j^b) = \sum_{(p,q) \in \mathcal{E}(X_i^b \cup X_j^b)} E_{coh}(c_p, c_q) + \sum_{(u,v) \in \mathcal{E}_{edge}(X_i^b \cup X_j^b)} E_{sep}(c_u, c_v). \tag{9}$$

4. *Splitting energy.* For each segment X_i^b, we will obtain its optimal binary splitting $\{X_i^{b,0}, X_i^{b,1}\}$. We define the splitting energy of X_i^b as the sum of the remaining and inter-segment energies of $X_i^{b,0}$ and $X_i^{b,1}$:

$$E_{split}(X_i^b) = E_{remain}(X_i^{b,0}) + E_{remain}(X_i^{b,1}) + E_{int}(X_i^{b,0}, X_i^{b,1}). \tag{10}$$

5. *The algorithm.* Guided by E_{remain}, E_{merge}, E_{split} and E_{int}, all tentative segments evolve themselves according to the energy minimizing rules. To simplify the problem, for each tentative segment X_i^b, we only test the merging hypothesis with its nearest segment $X_{P(i)}^b$. The distance between two segments is defined as the minimal distance between their subcomponents:

$$D(X_i^b, X_j^b) = \min_{f_i \in \mathcal{M}^i, f_j \in \mathcal{M}^j} D(f_i, f_j), \tag{11}$$

where $D(f_i, f_j)$ is the distance between feature vectors f_i and f_j.

Fig. 4. MCGC segmentation using different cues: (a) color image segmentation using only color and color+edge; (b) comparative segmentation results using color, color+edge, color+texture and color+texture+edge, respectively.

3 Experimental Results

To evaluate the proposed framework, we tested three commonly used features in our experiments: two generative features (color and texture), and one discriminative feature (i.e., salient edge). We show that even these three simple features work well in most cases for both textured and untextured regions and other difficult situations.

As shown in Fig. 1, integrating multicues can significantly improve the performance than single cue segmentation. Fig. 4 shows comparative results using different cues. We can clearly see the improvement when more cues are properly

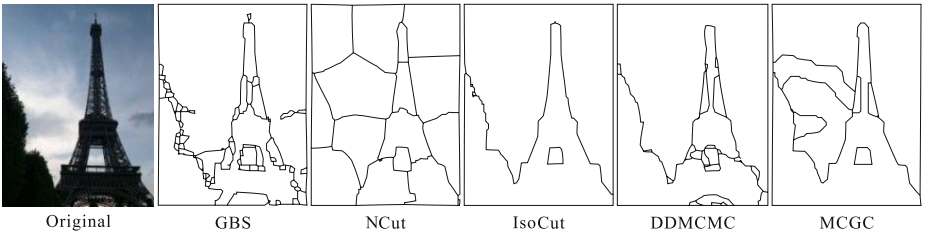


Fig. 5. Comparative segmentation results of GBS, NCut, IsoCut, DDMCMC and the MCGC approach (using color only) for soft boundary preservation

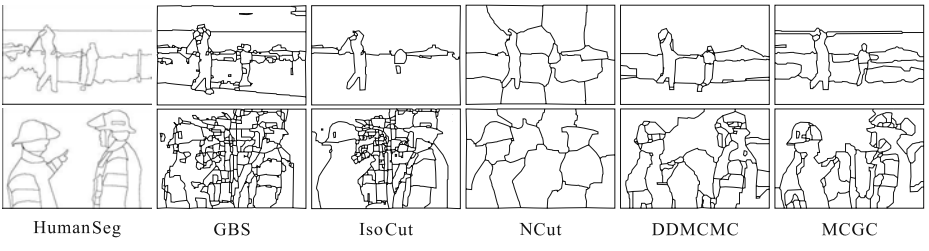


Fig. 6. Comparative segmentation results of GBS, IsoCut, NCut, DDMCMC and the MCGC approach (using color, texture and salient edge) with the ground truth

used. The untextured regions (e.g., the ground) and textured regions (e.g. the torso) in Fig. 4(b) were properly separated by integrating color, texture and salient edge cues. We can also see the role of salient edge in preserving soft boundaries from Fig. 4(a).

We have compared our MCGC approach to several state of art methods: (1) normalized cut (NCut) [9]; (2) isoperimetric cut (IsoCut) [14]; (3) efficient graph based segmentation (GBS) [15]; and (4) the DDMCMC [16]. Compared to these works that either focus on some particular cues or use discriminative models, our approach can be viewed as a kind of combination of global generative model and local discriminative model, which makes our approach good at preserving soft edges than discriminative methods. Fig. 5 shows the comparative results in the preservation of soft boundaries (see the cloud regions). As shown in Fig. 6, among all tested methods, GBS [15] efficiently generates segments according to local similarity. But the bottom-up nature makes GBS tend to oversegment the image and results in many spurious segments. In contrast, IsoCut [14], NCut [9], DDMCMC [16] and the proposed MCGC have different energy functions to guarantee global or near global optimum, which results in a fine balance between segmentation accuracy and spatial coherence. Among all the tested methods, GBS, IsoCut, DDMCMC and the our MCGC approach are self-validated, while the NCut algorithm need to indicate the number of segments. We can also see that compared to other self-validated methods, our approach has general consistent segmentations that are qualitatively similar to human labeled ground truth.

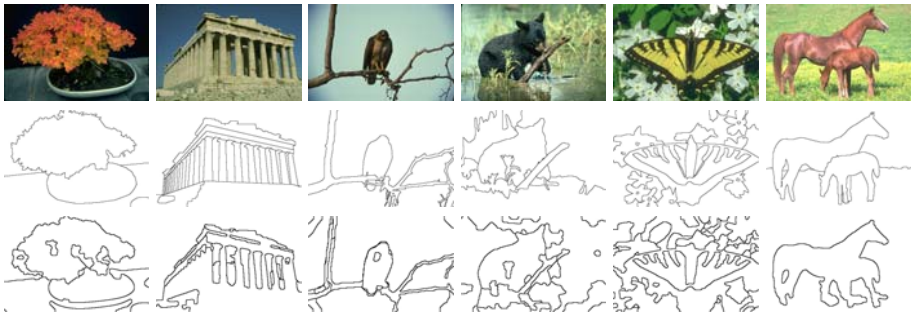


Fig. 7. Comparison to the ground truth. The first row shows the original images. The second row shows the segmentations labeled by human. The third row shows segmentation results of the proposed MCGC approach.

In Fig. 7, we compare the results of our approach with the human labeled object boundaries, which can be viewed as the ground truth of segmentation. The results are generally encouraging. But the last example (the horse) show that perceptually important boundaries may be of weak strength of low-level features. These faint edges are very difficult to detect using only low-level cues no matter how to adjust the influence of likelihood energy E_{lik}, coherence energy E_{coh} and separate energy E_{sep}. To cope with this problem, some high-level cues such as shape or user interaction should be considered.

Numerical efficiency. For an image graph with n vertices and m arcs, the worst-case complexity of binary graph mincut is $O(mn^2)$. Since our method is composed of a series of binary graph mincut and in most cases converges within less than 10 iterations, thus our method is also of polynomial complexity. Particularly, for a image of size 480×320, the average running time is about 80s on a PC with 3.2GHz CPU and 1GB memory. Furthermore, for a large image, we can run our method on the image pyramid. For example, we could conduct the segment evolving process only at a large scale, and conduct only refinement at finer scales.

4 Conclusions

In this paper, we have proposed a unified framework to utilize multiple cues in image segmentation. Unlike most previous works, we consider both generative features and discriminative features. Using graduated graph mincut, our approach produces a natural hierarchical coarse-to-fine segmentation and can implicitly determine the number of segments. Using color, texture and multi-scale edge cues as an example, we applied the proposed method to a wide range of image segmentation problems, and found encouraging results.

The stepwise optimization structure of the proposed framework also exhibits the possibility of combining user interaction in the segmentation process. This

may be helpful to break the limitation of automatic segmentation using only low-level cues. We will continue our work along this interesting direction.

Acknowledgments. The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No.: 9041369 and CityU 7008026).

References

1. Feng, W., Liu, Z.Q.: Region-level image authentication using Bayesian structural content abstraction. IEEE Trans. Image Processing 17(12), 2413–2424 (2008)
2. Feng, W., Xie, L., Zeng, J., Liu, Z.Q.: Audio-visual human recognition using semi-supervised spectral learning and hidden Markov models. Journal of Visual Languages and Computing 20(3), 188–195 (2009)
3. Martin, D., Fowlkes, C., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. of ICCV 2001, vol. 2, pp. 416–423 (2001)
4. Sharon, E., Brandt, A., Basri, R.: Fast multiscale image segmentation. In: Proc. of CVPR 2000, vol. 1, pp. 70–77 (2000)
5. Yu, S.X.: Segmentation using multiscale cues. In: Proc. of CVPR 2004, vol. 1, pp. 247–254 (2004)
6. Benezit, F., Cour, T., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Proc. of CVPR 2005, vol. 2, pp. 1124–1131 (2005)
7. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Intl. Journal of Computer Vision 43(1), 7–27 (2001)
8. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Machine Intell. 26(5), 530–549 (2004)
9. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 22(8), 888–905 (2000)
10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Machine Intell. 26(2), 147–159 (2004)
11. Li, S.Z.: Markov Random Field Modeling in Computer Vision. Springer, Heidelberg (1995)
12. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: Proc. of ICCV 2001, pp. 105–112 (2001)
13. Feng, W., Liu, Z.Q.: Self-validated and spatially coherent clustering with net-structured MRF and graph cuts. In: Proc. of ICPR 2006, vol. 4, pp. 37–40 (2006)
14. Grady, L., Schwartz, E.: Isoperimetric graph partition for image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 28(3), 469–475 (2006)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Intl. Journal of Computer Vision 59(2), 167–181 (2004)
16. Tu, Z., Zhu, S.C.: Image segmentation by data-driven Markov Chain Monte Carlo. IEEE Trans. Pattern Anal. Machine Intell. 24(5), 657–673 (2002)

Author Index