André Gagalowicz
Wilfried Philips (Eds.)

LNCS 6930

Computer Vision/ Computer Graphics Collaboration Techniques

5th International Conference, MIRAGE 2011
Rocquencourt, France, October 2011
Proceedings

Springer

Lecture Notes in Computer Science 6930

André Gagalowicz   Wilfried Philips (Eds.)

Computer Vision/ Computer Graphics Collaboration Techniques

5th International Conference, MIRAGE 2011
Rocquencourt, France, October 10-11, 2011
Proceedings

Springer

Volume Editors

André Gagalowicz
INRIA Rocquencourt
Domaine de Voluceau
78153 Le Chesnay, France
E-mail: andre.gagalowicz@inria.fr

Wilfried Philips
Ghent University
TELIN
St. -Pietersnieuwstraat 41
9000 Ghent, Belgium
E-mail: philips@telin.ugent.be

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-24135-2 e-ISBN 978-3-642-24136-9
DOI 10.1007/978-3-642-24136-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.3, H.5.2, I.4-5, I.2, J.3, I.2.10

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Preface

This volume collects the papers accepted for presentation at MIRAGE 2011.

The MIRAGE conference is recognized internationally with presentations coming from 19 countries despite the large worldwide economical crisis. Submissions from Asia dropped compared with two years ago, and were fewer, than those from Europe. France proved to be the most active scientifically in this area this year again.

All papers were reviewed by three to four members of the Program Committee. The final selection was carried out by the Conference Chairs by strictly following the reviewers' decisions.

At this point, we wish to thank all the Program Committee members for their timely and high-quality reviews. We also thank the invited speakers Peter Eisert and John Paul Lewis for kindly accepting to present very exciting talks that should allure many people to the conference.

MIRAGE 2011 was organized by INRIA Rocquencourt and took place at INRIA, Rocquencourt, close to the Versailles Castle. The next conference will take place in two years in Berlin and will be chaired by Peter Eisert. We believe that the conference was a stimulating experience for the audience, and that everybody had an enjoyable stay in the nice city of Versailles, enjoying our excellent gala dinner which took place in a very cosy castle.

June 2011

A. Gagalowicz
W. Philips

Organization

MIRAGE 2011 was organized by INRIA and Ghent University.

Conference Chair

André Gagalowicz	INRIA Rocquencourt, Le Chesnay, France

Conference Co-chairs

Peter Eisert	Fraunhofer HHI / Humboldt University, Germany
J.P. Lewis	Weta Digital, Victoria University, New Zealand

Organizing Committee

André Gagalowicz	INRIA Rocquencourt, Le Chesnay, France
Chantal Girodon	INRIA Rocquencourt, Rocquencourt, France
Wilfried Philips	Ghent University - IBBT, Ghent, Belgium

Program Committee

Ken Anjyo	OLM Digital, Inc., Japan
Kai-Uwe Barthel	University of Applied Sciences THW Berlin, Germany
Jacques Blanc-Talon	DGA, France
Kadi Bouatouch	IRISA, France
José Braz	Polytechnic Institute of Setúbal, Portugal
Antonio Camurri	University of Genoa, Italy
Leszek Chmielewski	Warsaw University of Life Sciences, Poland
Adrian Clark	University of Essex, UK
John Collomosse	University of Surrey, UK
Silvana Delepiane	University of Genoa, Italy
Silvana Dellepiane	University of Genoa, Italy
Peter Eisert	Fraunhofer HHI / Humboldt University, Germany
Alexandre Francois	Harvey Mudd College, USA
Bernd Froehlich	Bauhaus-Universität Weimar, Germany
Andrea Fusiello	Università degli Studi di Verona, Italy
André Gagalowicz	INRIA Rocquencourt, France
Oliver Grau	BBC, UK

Radek Grzeszczuk	Nokia Research Lab, USA
Cédric Guiard	Agence de Doublures Numériques / l'Etude et la Supervision des Trucages, France
James Hays	Brown University, USA
Derek Hoiem	University of Illinois at Urbana-Champaign, USA
Patrick Horain	Institut Télécom / Télécom SudParis, France
Joachim Hornegger	Friedrich Alexander University of Erlangen-Nuremberg, Germany
Reinhard Klette	The University of Auckland, New Zealand
Andreas Kolb	Universität Siegen, Germany
Ivana Kolingerova	University of West Bohemia, Czech Republic
Juliusz Kulikowski	Institute of Biocybernetics and Biomedical Engineering, Poland
Tosiyasu Kunii	Morpho, Inc., Japan
J.P. Lewis	Weta Digital, Victoria University, New Zealand
Xiaowei Li	Google, USA
Nadia Magnenat-Thalmann	University of Geneva, Switzerland
Marcus Magnor	Technische Universität Braunschweig, Germany
Ronald Mallet	Industrial Light and Magic, USA
Takashi Matsuyama	Kyoto University, Japan
Vittorio Murino	Università degli Studi di Verona, Italy
Ryohei Nakatsu	NUS, Singapore
Heinrich Niemann	Friedrich Alexander Universität, Germany
Kazunori Okada	San Francisco State University, USA
Dietrich Paulus	University of Koblenz, Germany
Wilfried Philips	Ghent University - IBBT, Belgium
Dan Popescu	CSIRO, Australia
Ralf Reulke	Humboldt-Universität zu Berlin, Germany
John Robinson	University of York, UK
Doug Roble	Digital Domain, USA
Christian Roessl	University of Magdeburg, Germany
Bodo Rosenhahn	University of Hannover, Germany
Robert Sablatnig	Vienna University of Technology, Austria
Mateu Sbert	Universitat de Girona, Spain
Franc Solina	University of Ljubljana, Slovenia
Alexeï Sourin	National Technological University NTU, Singapore
Marc Stamminger	University of Erlangen, Germany
Akihiro Sugimoto	National Institute of Informatics, Japan
David Suter	University of Adelaide, Australia
Demetri Terzopoulos	UCLA, USA
Matthias Teschner	University of Freiburg, Germany
Daniel Thalmann	EPFL, Switzerland
Christian Theobalt	Max-Planck Institut, Germany
Emanuele Trucco	University of Dundee, UK

Raquel Urtasun	Toyota Technological Institute at Chicago, USA
Thomas Vetter	Basel University, Switzerland
Jue Wang	Adobe, USA
Josh Wills	Sony Pictures Imageworks, USA
Konrad Wojciechowski	Institute of Automation, Poland
Lior Wolf	Tel Aviv University, Israel
Hau San Wong	City University of Hong Kong, China
Cha Zhang	Microsoft Research, USA
Huijing Zhao	Peking University, P.R. China
Tatjana Zrimec	University of South Wales, Australia

Reviewers

Ken Anjyo	OLM Digital, Inc., Japan
Jacques Blanc-Talon	DGA, France
Kadi Bouatouch	IRISA, France
José Braz	Polytechnic Institute of Setúbal, Portugal
Leszek Chmielewski	Warsaw University of Life Sciences, Poland
Adrian Clark	University of Essex, UK
John Collomosse	University of Surrey, UK
Silvana Dellepiane	University of Genoa, Italy
Peter Eisert	Fraunhofer HHI / Humboldt University, Germany
Alexandre Francois	Harvey Mudd College, USA
Andrea Fusiello	Università degli Studi di Verona, Italy
André Gagalowicz	INRIA Rocquencourt, France
Oliver Grau	BBC, UK
Radek Grzeszczuk	Nokia Research Lab, USA
Cédric Guiard	Agence de Doublures Numériques / l'Etude et la Supervision des Trucages, France
James Hays	Brown University, USA
Derek Hoiem	University of Illinois at Urbana-Champaign, USA
Patrick Horain	Institut Télécom / Télécom SudParis, France
Joachim Hornegger	Friedrich Alexander University of Erlangen-Nuremberg, Germany
Reinhard Klette	The University of Auckland, New Zealand
Andreas Kolb	Universität Siegen, Germany
Tosiyasu Kunii	Morpho, Inc., Japan
J.P. Lewis	Weta Digital, Victoria University, New Zealand
Xiaowei Li	Google, USA
Nadia Magnenat-Thalmann	University of Geneva, Switzerland
Marcus Magnor	Technische Universität Braunschweig, Germany
Ronald Mallet	Industrial Light and Magic, USA
Takashi Matsuyama	Kyoto University, Japan

Vittorio Murino	Università degli Studi di Verona, Italy
Heinrich Niemann	Friedrich Alexander Universität, Germany
Kazunori Okada	San Francisco State University, USA
Dietrich Paulus	University of Koblenz, Germany
Wilfried Philips	Ghent University - IBBT, Belgium
Dan Popescu	CSIRO, Australia
Ralf Reulke	Humboldt-Universität zu Berlin, Germany
John Robinson	University of York, UK
Christian Roessl	University of Magdeburg, Deutschland
Bodo Rosenhahn	University of Hannover, Germany
Mateu Sbert	Universitat de Girona, Spain
Franc Solina	University of Ljubljana, Slovenia
Alexeï Sourin	National Technological University NTU, Singapore
Marc Stamminger	University of Erlangen, Germany
Akihiro Sugimoto	National Institute of Informatics, Japan
Matthias Teschner	University of Freiburg, Germany
Christian Theobalt	Max-Planck Institut, Germany
Emanuele Trucco	University of Dundee, UK
Thomas Vetter	Basel University, Switzerland
Jue Wang	Adobe, USA
Konrad Wojciechowski	Institute of Automation, Poland
Lior Wolf	Tel Aviv University, Israel
Hau San Wong	City University of Hong Kong, China
Cha Zhang	Microsoft Research, USA
Huijing Zhao	Peking University, P.R. China
Tatjana Zrimec	Univesity of New South Wales, Australia
Tatjana Zrimec	University of South Wales, Australia

Table of Contents

Bundle Adjustment for Stereoscopic 3D

Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel

Max Planck Institute for Computer Science (MPII)
Campus E1 4, 66123 Saarbrücken, Germany
ckurz@mpi-inf.mpg.de

Abstract. The recent resurgence of stereoscopic 3D films has triggered a high demand for post-processing tools for stereoscopic image sequences. Camera motion estimation, also known as structure-from-motion (SfM) or match-moving, is an essential step in the post-processing pipeline. In order to ensure a high accuracy of the estimated camera parameters, a bundle adjustment algorithm should be employed. We present a new stereo camera model for bundle adjustment. It is designed to be applicable to a wide range of cameras employed in today's movie productions. In addition, we describe how the model can be integrated efficiently into the sparse bundle adjustment framework, enabling the processing of stereoscopic image sequences with traditional efficiency and improved accuracy. Our camera model is validated by synthetic experiments, on rendered sequences, and on a variety of real-world video sequences.

1 Introduction

In computer vision, stereo image sequences have been employed for a large number of applications over the past decades. However, the largest body of work can be found on robot or autonomous vehicle navigation and motion estimation. Implicated by this predominant area of application, stereo processing pipelines usually have to face restrictive real-time requirements. Furthermore, there are limits on the amount of data the algorithms are allowed to accumulate and process. These requirements influence the types of algorithms employed.

Recently, however, the revival of 3D films using modern stereo 3D (S3D) technology has entailed the creation of an unprecedented amount of high-resolution stereo image data. Today's movies are often augmented with virtual objects, and sometimes even the major part of the movie is computer generated. In order to composite the virtual objects with a real image sequence, the camera parameters of the real camera have to be estimated to render the virtual object with the corresponding virtual camera. Thus, reliable and accurate camera motion estimation for S3D sequences is a crucial part in movie post-processing and essential for the creation of convincing special effects. Given the amount of computation involved, post-processing is inherently done off-line and does not shy away from computationally expensive algorithms.

Considering the increase in demand, some commercially available match-moving packages already incorporate solvers for stereo cameras. However, the

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 1–12, 2011.

employed algorithms are not published and an academic paper presenting a solution to high quality camera motion estimation for stereo cameras is (to the best of our knowledge) not yet available.

We present an approach allowing reliable and accurate camera motion estimation for stereo sequences. In contrast to existing real-time approaches, we employ a large number of automatically extracted feature points and optimize the camera parameters with the gold-standard method: bundle adjustment. As known from literature, the naïve implementation of bundle adjustment is computationally expensive beyond feasability and can be sped up by employing the sparse matrix structure of the Jacobian. The contributions of this paper are:

- An extended camera model for stereo cameras is presented. The model offers great flexibility in terms of its parameters and therefore can be employed for a variety of different cameras, ranging from entry-level consumer 3D camcorders using a 3D conversion lens with a static camera geometry to professional cameras used in movie productions.
- It is shown how the additional constraints introduced by the camera model can be incorporated into the sparse bundle adjustment framework.

The approach is validated on a variety of data sets, from fully synthetic experiments to challenging real-world image sequences.

This paper is organized as follows: Related work will be reviewed in the next section, followed by a brief summary of camera motion estimation in Sec. 3. Sec. 4 introduces our new camera model for stereoscopic bundle adjustment, and the incorporation into bundle adjustment is described in Sec. 5. The results of our new approach are shown in Sec. 6, followed by the conclusion.

2 Related Work

Structure-from-Motion. A general introduction to bundle adjustment can be found in [1, 2]. Of late, research has been done towards processing data of multiple independently moving cameras [3], or entire community photo collections [4], demonstrating orthogonal approaches. Multi-camera systems either assume a static and calibrated camera setup on a moving platform [5, 6] or obtain the calibration by averaging parameters of the independent reconstructions [7]. There exist alternative approaches to SfM, but either the stereo rig is assumed to be calibrated and no bundle adjustment is used [8], or the bundle adjustment remains unaffected by the changes to the reconstruction pipeline [9]. To a certain extent, constraints arising from stereo geometry have been included in bundle adjustment [10], but the model is incorporated into the algorithm by simply adding soft constraints and without addressing the sparse structure of the problem.

Self-Calibration. The problem of self-calibration for an uncalibrated stereo rig with an unknown motion has been explicitly modelled for two pairs of stereo images [11], even with varying vergence angles [12], but the focus of these papers is rather on obtaining a one-time calibration of these two stereo pairs instead of the optimization over a complete image sequence.

Stereo Navigation, Ego-motion Estimation, Visual Odometry. Stereo rigs used in robot or autonomous vehicle navigation and motion estimation are usually assumed to be calibrated. Due to runtime constraints, the problem of motion estimation is often reduced to estimating the parameters of an inter-frame motion model given two distinct sets of 3D points, and then feeding the results to a Kalman filter to achieve robustness (see [13–16], for example). Optimized feature selection and tracking, especially *multi-frame tracking*, is used in [17] to achieve robustness for tracking features over longer sequences. There are attempts at using bundle adjustment in visual odometry, thereby incorporating the data produced by a calibrated stereo rig directly [18–20], but, in contrast to these approaches, we do not assume the calibration of the stereo rig to be known. A *reduced order bundle adjustment* is used in [21], but the processing and parametrization of the input data are again tailored to meet the real-time requirements of the system. In [22], a correlation-based approach to ego-motion and scene structure estimation from stereo sequences is presented. The approach is different from bundle adjustment and the transformation between left and right frames is assumed to be constant.

Uncalibrated Stereo. Various approaches exist to obtain the epipolar geometry of an uncalibrated stereo rig [23–27], but these methods only consider a single pair of images and there is no further optimization. Visual servoing [28–30] and man-machine interaction [31] sometimes rely upon uncalibrated stereo cameras, but the cameras are static and the algorithms avoid explicit 3D reconstructions. For a moving stereo rig, restrictive assumptions on the scene structure have to be made [32]. Quasi-Euclidean epipolar rectification [33] has recently been adapted to work on uncalibrated stereo sequences [34], even with non-linear optimization [35], but the scene representation differs from bundle adjustment.

Optical Flow, Three-Dimensional Scene Flow. While camera setups in optical flow applications frequently employ two [36, 37] or more cameras [38], research in this area is more geared towards recovering the non-rigid scene motion [39], whereupon the cameras are assumed to be calibrated. Optical flow can be adapted for ego-motion estimation [40], but the method uses rectified input images and makes restrictive assumptions on the scene structure.

Commercial Products. Several commercial products feature tools for stereoscopic tracking and stereo solving (PFTrack$^{\text{TM}}$and SynthEyes$^{\text{TM}}$, for example), but the corresponding algorithms have not been published.

3 Structure-from-Motion

Given a sequence of K images I_k, SfM refers to the procedure of deriving a camera matrix $\mathtt{A}_k$ for every image (representing the camera motion), and a set of J 3D object points $\mathbf{P}_j = (P_x, P_y, P_z, 1)^\top$ (representing the static scene structure). The 2D feature point corresponding to $\mathbf{P}_j$ in image I_k is denoted by $\mathbf{p}_{j,\,k}$.

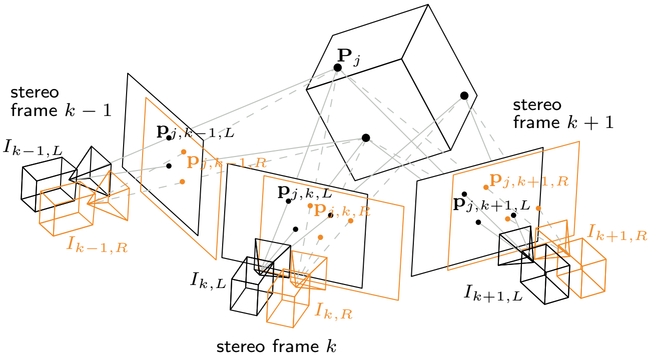


Fig. 1. Each stereo frame consists of a left camera image $I_{k,\,L}$ and a right camera image $I_{k,\,R}$. In contrast to monocular SfM, there are now two sets of corresponding 2D feature points $\mathbf{p}_{j,\,k,\,L}$ and $\mathbf{p}_{j,\,k,\,R}$ for the set of 3D object points $\mathbf{P}_j$.

Traditionally, the SfM pipeline consists of several steps. At first, the 2D feature points are detected and tracked, and outliers are eliminated using geometric constraints (e.g., the fundamental matrix). In the next step, initial camera parameters and 3D object points are established. To obtain initial values for the intrinsic camera parameters, self-calibration is performed. These steps are not described in this paper; details can be found in the literature [1]. As last step, bundle adjustment is employed, which will be discussed in the following.

The goal of bundle adjustment is to minimize the reprojection error given by the cost function

$$\operatorname*{arg\,min}_{\mathtt{A},\mathbf{P}} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} \mathrm{d}(\mathbf{p}_{j,\,k}\,,\, \mathtt{A}_k\,\mathbf{P}_j)^2 \quad , \tag{1}$$

where d(...) denotes the Euclidean distance. Thereby, the error is equally distributed over the whole scene. For numerical optimization of Eq. (1), the sparse Levenberg-Marquardt (LM) algorithm is typically employed [1].

In the case of a stereo camera setup, the input consists of K stereo frames. For convenience, the individual images are now denoted as $I_{k,\,L}$ for the image of the left camera, and $I_{k,\,R}$ for the image of the right camera. Analogical, we get separate projection matrices $\mathtt{A}_{k,\,L}$ and $\mathtt{A}_{k,\,R}$, and we have to distinguish between 2D feature points $\mathbf{p}_{j,\,k,\,L}$ and $\mathbf{p}_{j,\,k,\,R}$, respectively (see Fig. 1).

Introducing $x \in \{L, R\}$, the cost function from Eq. (1) translates to

$$\operatorname*{arg\,min}_{\mathtt{A},\mathbf{P}} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{x} \mathrm{d}(\mathbf{p}_{j,\,k,\,x}\,,\, \mathtt{A}_{k,\,x}\,\mathbf{P}_j)^2 \quad . \tag{2}$$

4 Camera Model

In this section, we first describe the camera model for our stereo bundle adjustment for a metric camera. Bundle adjustment for monocular sequences is often also performed with a projective camera model [1]. However, the representation of the geometric constraints between the left and the right camera is not possible in the projective framework, because transformations in the local camera coordinate system including rotations and translations cannot be parametrized independently from the current projective camera matrix. Thus, we propose to enforce the constraints introduced by our metric stereo camera model after an update from projective to metric space has been performed (cp. [1, 41]).

The 3×4 projection matrix $\mathtt{A}$ of a metric camera can be decomposed as

$$\mathtt{A} = \mathtt{K}\,[\,\mathtt{I}\,|\,\mathbf{0}\,]\begin{bmatrix} \mathtt{R} & -\mathtt{R}\,\mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \quad , \tag{3}$$

where $\mathbf{C}$ is the position of the camera center in world coordinate frame, $\mathtt{R}$ is a rotation matrix representing the camera orientation, and $\mathtt{K}$ is a calibration matrix comprising the intrinisc camera parameters, such as focal length. The index k assigning a projection matrix to the corresponding image is omitted throughout this chapter for the sake of readability.

Considering a standard stereo camera setup as employed in movie productions, our first observation is that the two cameras of the stereo system undergo only dependent motion – if the left camera translates to the right, the right camera will inherently have to follow that same translation. Now, in order to improve over the conventional bundle adjustment algorithm, we exploit this dependency: Instead of treating the left and the right camera as separate entities, we consider them as instances of the same camera system. A change of parameters introduced by the left camera will therefore influence the position and orientation of the right camera, and vice versa.

Secondly, to benefit from the combined camera model, the total number of parameters representing the camera over the whole image sequence has to be reduced. Since modern stereo camera systems allow the point of convergence of the two cameras to change during acquisition, the relative rotation between the cameras can not always be assumed to be constant over the sequence. Therefore, this constraint, which would reduce the number of parameters significantly, is only optionally enforced (however, all our results enforce this constraint).

Assuming the relative position offset of the two camera centers to be unknown but constant is a constraint we always enforce, because the baseline between the cameras is usually not changed. As a matter of principle, there is some freedom in the choice of the stereo system base position. We chose it to coincide with the center of the left camera. The result are two different decompositions for the left and the right camera that can be expressed as

$$\mathtt{A}_L = \mathtt{K}_L \qquad [\,\mathtt{R}_L\,|\,\mathbf{0}\,] \qquad \begin{bmatrix} \mathtt{R} & -\mathtt{R}\,\mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \quad , \tag{4}$$

$$\mathtt{A}_R = \mathtt{K}_R\,[\,\mathtt{R}_R\,|\,-\mathtt{R}_R\mathbf{C_R}\,]\begin{bmatrix} \mathtt{R} & -\mathtt{R}\,\mathbf{C} \\ \mathbf{0} & 1 \end{bmatrix} \quad , \tag{5}$$

where subscripts L and R denote parameters that are exclusive to the left and right camera respectively.

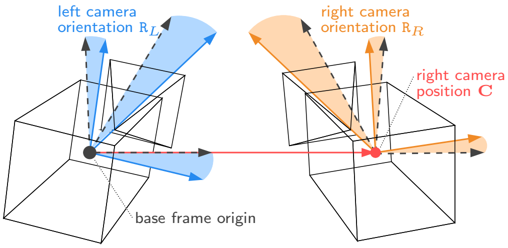


Fig. 2. Our novel camera model for bundle adjustment. The camera geometry of every stereo frame is given by a base frame (dashed lines), whose origin is aligned with the center of the left camera. The orientation $\mathtt{R}_L$ of the left camera is encoded independent from the orientation of the base frame, allowing the position of the right camera to be specified by a single parameter $\mathbf{C}$ (red arrow) for the whole sequence.

The rotation matrix of the left camera $\mathtt{R}_L$ could be omitted for a static stereo setup. However, if the point of camera convergence changes in a dynamic setup, it is necessary to encode the orientation of the left camera separately from the orientation of the stereo system. This is due to the fact that a rotation of the left camera would otherwise inherently lead to a rotation of the coordinate frame in which the relative translation of the right camera takes place (see Fig. 2).

Depending on the actual acquisition system in operation, parameters can be chosen to be estimated for every frame, for a subset of frames, or for the whole sequence. Furthermore, the intrinsic camera parameters can of course be treated as shared between the two cameras, if this was the case at the time of recording.

5 Bundle Adjustment

To optimize Eq. (2), we extend the sparse LM algorithm [1].

First, we assemble a parameter vector $\mathbf{q} = (\mathbf{b}^\top, \mathbf{c}^\top, \mathbf{d}^\top, \mathbf{e}^\top, \mathbf{f}^\top, \mathbf{g}^\top)^\top$. The designation of the corresponding subvector for all parameters of our camera model can be found in Tab. 1, along with a listing of the number of parameters and the number of the respective vector entries.

Most parameters can either be assumed to be variable for each frame or joined (i.e., estimated conjointly) over the whole sequence. The intrinsic parameters can also be shared for both cameras.

It is also possible to restrict $\mathtt{R}_L$ and $\mathtt{R}_R$ in a way that makes them depend on the vergence angle only. Dependent on the degrees of freedom for the convergence point, this results in 1 or 2 degrees of freedom for the rotation matrices $\mathtt{R}_L$ and $\mathtt{R}_R$ (cp. Tab. 1).

For the sake of simplicity, we will assume a static stereo setup with joined and shared intrinsic parameters henceforth, resulting in two single rotation matrices $\mathtt{R}_L$ and $\mathtt{R}_R$ over the whole sequence, and a single calibration matrix $\mathtt{K}$. This would be the case in a stereo setup with a fixed convergence point, e.g., a camcorder with a 3D conversion lens.

Table 1. Stereo model parameters with their typical parameter count, the number of elements in the associated vector, and the designation of the corresponding vector. Example: For a sequence of $K = 10$ images, $\mathbf{b}$ contains 10 elements with 6 parameters each, i.e., 60 entries in total. 'Joined' indicates that the parameters are constant and are jointly estimated over the whole sequence. 'Shared' indicates that the respective parameters of the right camera are estimated in combination with the corresponding parameters of the left camera, so that there are no separate entries for these parameters in the matrix $\mathtt{J}^\top\mathtt{J}$.

Model parameters	# of parameters	# of vector elements	designation
base frame $\mathbf{C}$, $\mathtt{R}$	6	K	$\mathbf{b}$
left orientation $\mathtt{R}_L$	1-3	K, 1 (joined)	$\mathbf{c}$
right position $\mathbf{C}_\mathbf{R}$	3	1	$\mathbf{d}$
right orientation $\mathtt{R}_R$	1-3	K, 1 (joined), 0 (shared)	$\mathbf{e}$
left intrinsics $\mathtt{K}_L$	3	K, 1 (joined)	$\mathbf{f}$
right intrinsics $\mathtt{K}_R$	3	K, 1 (joined), 0 (shared)	$\mathbf{f}$
3D object points $\mathbf{P}_j$	3	J	$\mathbf{g}$

The least squares problem that is the core of bundle adjustment is tackled by the sparse LM algorithm that solves the linear equation system

$$\mathtt{J}\boldsymbol{\delta} = \boldsymbol{\epsilon} \tag{6}$$

with the Jacobian matrix $\mathtt{J} = \partial\mathbf{p}/\partial\mathbf{q}$, the residual vector $\boldsymbol{\epsilon}$, and the update vector $\boldsymbol{\delta}$. The Jacobian matrix $\mathtt{J}$ has the block structure $\mathtt{J} = [\mathtt{B\,C\,D\,E\,F\,G}]$, where $\mathtt{B} = \partial\mathbf{p}/\partial\mathbf{b}$, $\mathtt{C} = \partial\mathbf{p}/\partial\mathbf{c}$, et cetera. In the case of a conventional bundle adjustment that allows to enforce joined intrinsic parameters over the sequence, the Jacobian $\mathtt{J}$ only comprises the matrices $\mathtt{B}$, $\mathtt{F}$, and $\mathtt{G}$. Depending on the parameter interdependencies, $\mathtt{J}$ usually has a lot of zero entries (cp. Fig. 3). The measurement vector $\mathbf{p}$ is constructed by placing all the 2D feature points from all camera images in a single column vector. For the purpose of illustration, we assume them to be sorted by their affiliation to the left or right camera, then their image index k, and finally their corresponding 3D object point index j.

The solution to Eq. (6) is obtained by multiplication with $\mathtt{J}^\top$, thereby directly evaluating $\mathtt{J}^\top\mathtt{J}$ and $\mathtt{J}^\top\boldsymbol{\epsilon}$, leaving the explicit construction of $\mathtt{J}$ unnecessary.

A comparison of the structure of $\mathtt{J}^\top\mathtt{J}$ taken from our stereo bundle adjustment and from a conventional bundle adjustment can be found in Fig. 4. As becomes evident, we only introduce changes to one block in the structure, which is the top left one.Although the structure in the block is no longer sparse, this does not have any influence on the matrix inversion $(\mathtt{J}^\top\mathtt{J})^{-1}$, since other elements added on top during the sparse matrix inversion cause the sparse structure of this block to break down anyway (cp. [1]). Furthermore, the size of this block is significantly reduced due to the reduced number of parameters when using stereo bundle adjustment with constant convergence point, leading to better computational performance.

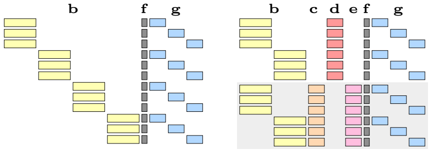


Fig. 3. Block structure of the Jacobian matrix J for a conventional bundle adjustment with joined intrinsic parameters (left), and for our stereo bundle adjustment (right). The individual block matrices are set apart by different coloring. The gray background on the right indicates derivatives contributed by the right camera.

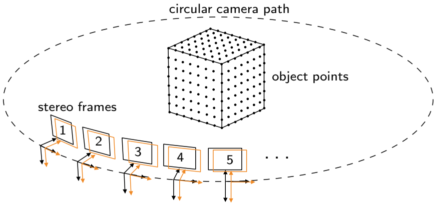


Fig. 5. The setup used in the synthetic experiments for the generation of the ground truth camera and 3D object point parameters.

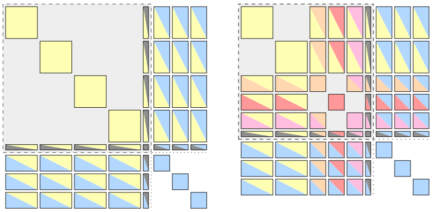


Fig. 4. Structure of the matrix $\mathtt{J}^\top\mathtt{J}$ used in the solution of Eq. (6) for a conventional bundle adjustment with joined intrinsic parameters (left), and for our stereo bundle adjustment (right). The color indicates the contribution of the individual elements in the matrix multiplication. The dashed square indicates the relevant block for matrix inversion.

Table 2. Average translation, rotation, and focal length error, and average time per iteration for the rendered sequence for an unconstrained bundle adjustment, a bundle adjustment with joined focal length, and our stereo bundle adjustment.

RMSE	unconst.	joined	stereo
translation	1.7274 mm	0.6459 mm	0.5964 mm
rotation	0.0112 deg	0.0026 deg	0.0024 deg
focal length	1.3609 mm	0.0975 mm	0.0600 mm
Avg. time	719 ms	860 ms	733 ms

6 Results

In this section we present the evaluation of our stereo bundle adjustment with purely synthetic data, rendered sequences and real-world sequences. The latter can also be found in the video accompanying this paper, which can be downloaded from `http://www.mpi-inf.mpg.de/users/ckurz/`.

Our setup for the synthetic experiments is sketched in Fig. 5. It consists of a virtual stereo configuration composed of two cameras. The cameras execute a circular motion around a set of 296 3D object points arranged in a regular grid on the surface of a cube. The cube has an edge length of 100 mm, the radius of the camera path is 300 mm, and the opening angle of the cameras is 30 degrees.

We generate a total of 40 stereo pairs per trial, providing 80 images per sequence. All the ground truth measurements for the 2D feature points contained in these images are calculated from the known ground truth camera and 3D object points parameters. In a last step before the reconstruction process, Gaussian noise with a standard deviation σ_{syn} is applied to the measurements.

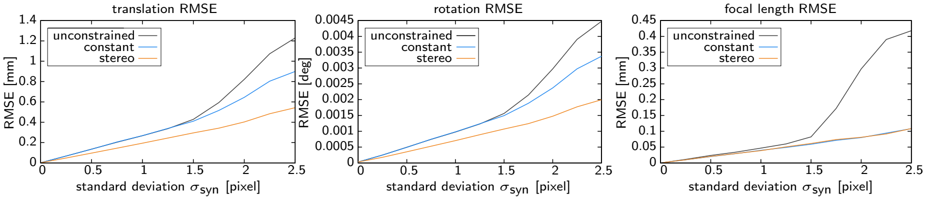


Fig. 6. Average translation, rotation, and focal length error for a given Gaussian error σ_{syn} of the 2D feature points over 1000 trials. The setup sketched in Fig. 5 was used for the generation of the ground truth parameters.

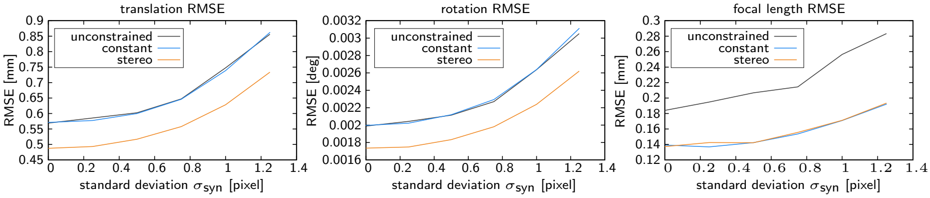


Fig. 7. Average translation, rotation and focal length error for a given Gaussian error σ_{syn} of the 2D feature points over 1000 trials, while 20 percent of the feature points were additionally disturbed by a large offset. The setup sketched in Fig. 5 was used for the generation of the ground truth parameters.

For each value of σ_{syn}, we perform a total of 1000 trials for a conventional bundle adjustment, a conventional bundle adjustment with joined focal length over the sequence, and our novel stereo bundle adjustment, whereas a different random disturbance is introduced in the measurements each time. For each reconstruction, a similarity transformation is estimated to register it to the ground truth, and then the average absolute position and orientation error is calculated. The results can be found in Fig. 6. Our stereo bundle adjustment clearly outperforms the conventional methods in terms of the translation and rotation error, while being on par with the conventional bundle adjustment with joined focal length for the error in the estimated focal length.

Furthermore, to simulate outliers, another test series was conducted. In this series, 20 percent of the measurements were disturbed by an offset of up to 12 pixel in addition to the Gaussian noise. Since not all outliers can be removed in the outlier elimination step, the results, which can be found in Fig. 7, differ. Our stereo bundle adjustment clearly outperforms both competitors again.

The second step in the evaluation was to process a rendered sequence with known ground truth parameters. Again, results were generated for a conventional bundle adjustment, a conventional bundle adjustment with joined focal length, and our stereo bundle adjustment (see Tab. 2). Our algorithm achieves the best results. In addition, Fig. 8 shows two sample stereo frames from the rendered sequence with a wireframe overlay using the estimated camera parameters. As can also be seen in the supplemental video, the wireframe fits the true scene geometry almost perfectly.

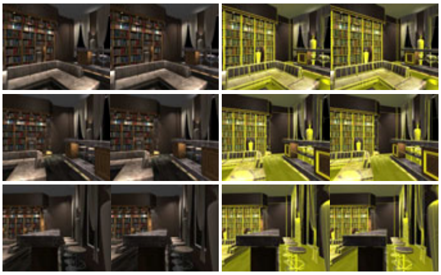


Fig. 8. This figure shows three example stereo frames from a rendered indoor sequence. The left images show the actual frames, whereas the right images the same images augmented with the wireframe model of the scene placed using the estimated camera parameters. The results can also be found in the video accompanying this paper.

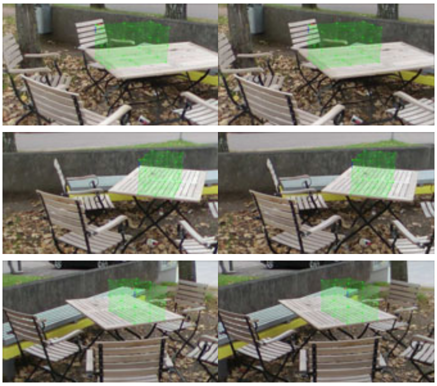


Fig. 9. Real-world sequence shot with a HD camcorder with a 3D conversion lens. The scene has been augmented by a green cuboid to demonstrate the quality of the estimated camera parameters.

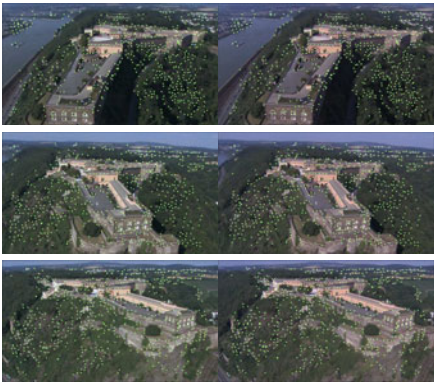


Fig. 10. Stereo frames from the Ehrenbreitstein Fortress sequence. The green dots signify the reprojections of reconstructed 3D points, showing that no drift in the parameters has occured.

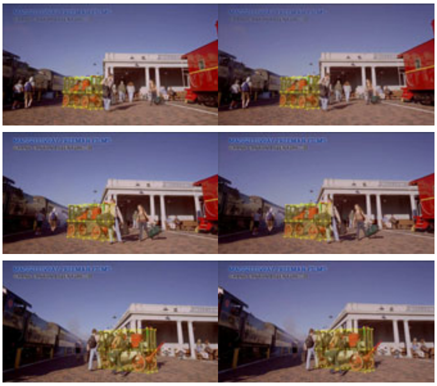


Fig. 11. Stereo frames from the train station sequence. The scene has been augmented by a yellow cuboid to demonstrate the quality of the estimated camera parameters.

The first real-world sequence (see Fig. 9) was captured with a Panasonic HDC-SDT750 camcorder with a 3D conversion lens and depicts some pieces of garden furniture. As can be seen by the overlay geometry, our stereo bundle adjustment was able to obtain excellent results for the camera parameters. The second sequence (see Fig. 10) depicts a flyover over Ehrenbreitstein Fortress in the Upper Rhine valley from the documentary *UNESCO World Heritage - Upper Middle Rhine Valley* (courtesy of *cinovent entertainment*).

In the third sequence (see Fig. 11), a scene at a train station from *Grand Canyon Adventure 3D* (courtesy of *MacGillivray Freeman Films*) is shown.

7 Conclusion and Future Work

We have presented a novel camera model for stereo cameras for use in bundle adjustment. The model has the generality to accommodate a wide range of the stereo cameras used in today's movie productions, and can be incorporated efficiently into the conventional sparse bundle adjustment algorithms. A multitude of tests has been conducted, validating our model.

For future work, we will update the other stages of the SfM pipeline to make full use of the additional information provided by stereoscopic image sequences.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
2. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment – A modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, p. 298. Springer, Heidelberg (2000)
3. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.-P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR (2009)
4. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: Intl. Conference on Computer Vision (2007)
5. Kim, J.H., Li, H., Hartley, R.: Motion estimation for multi-camera systems using global optimization. In: Computer Vision and Pattern Recognition (2008)
6. Stewenius, H., Åström, K.: Structure and motion problems for multiple rigidly moving cameras. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 252–263. Springer, Heidelberg (2004)
7. Frahm, J.-M., Köser, K., Koch, R.: Pose estimation for multi-camera systems. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 286–293. Springer, Heidelberg (2004)
8. Chandraker, M., Lim, J., Kriegman, D.J.: Moving in stereo: Efficient structure and motion using lines. In: International Conference on Computer Vision (2009)
9. Hirschmüller, H., Innocent, P.R., Garibaldi, J.M.: Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In: International Conference on Control, Automation, Robotics and Vision (2002)
10. Di, K., Xu, F., Li, R.: Constrained bundle adjustment of panoramic stereo images for mars landing site mapping. In: Mobile Mapping Technology (2004)
11. Zhang, Z., Luong, Q.T., Faugeras, O.: Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. TRA 12, 103–113 (1996)
12. Brooks, M.J., de Agapito, L., Huynh, D.Q., Baumela, L.: Towards robust metric reconstruction via a dynamic uncalibrated stereo head. Image and Vision Computing 16, 989–1002 (1998)
13. Matthies, L., Shafer, S.A.: Error modeling in stereo navigation. IEEE Journal of Robotics and Automation 3, 239–250 (1987)
14. Molton, N., Brady, M.: Practical structure and motion from stereo when motion is unconstrained. International Journal of Computer Vision 39, 5–23 (2000)
15. Saeedi, P., Lawrence, P.D., Lowe, D.G.: 3d motion tracking of a mobile robot in a natural environment. In: ICRA, pp. 1682–1687 (2000)
16. Weng, J., Cohen, P., Rebibo, N.: Motion and structure estimation from stereo image sequences. Transactions on Robotics and Automation 8, 362–382 (1992)

17. Olson, C.F., Matthies, L.H., Schoppers, M., Maimone, M.W.: Rover navigation using stereo ego-motion. Robotics and Autonomous Systems 43, 215–229 (2003)
18. Sünderhauf, N., Konolige, K., Lacroix, S., Protzel, P.: Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In: AMS (2005)
19. Sünderhauf, N., Protzel, P.: Towards using sparse bundle adjustment for robust stereo odometry in outdoor terrain. In: TAROS, pp. 206–213 (2006)
20. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: CVPR (2004)
21. Dang, T., Hoffmann, C., Stiller, C.: Continuous stereo self-calibration by camera parameter tracking. Transactions on Image Processing 18, 1536–1550 (2009)
22. Mandelbaum, R., Salgian, G., Sawhney, H.: Correlation-based estimation of ego-motion and structure from motion and stereo. In: ICCV, vol. 1, pp. 544–550 (1999)
23. Akhloufi, M., Polotski, V., Cohen, P.: Virtual view synthesis from uncalibrated stereo cameras. In: Multimedia Computing and Systems, pp. 672–677 (1999)
24. Hartley, R., Gupta, R., Chang, T.: Stereo from uncalibrated cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (1992)
25. Ko, J.H., Park, C.J., Kim, E.S.: A new rectification scheme for uncalibrated stereo image pairs and its application to intermediate view reconstruction. In: Optical Information Systems II, Proceedings of SPIE, vol. 5557, pp. 98–109 (2004)
26. Yin, X., Xie, M.: Estimation of the fundamental matrix from uncalibrated stereo hand images for 3d hand gesture recognition. PR 36, 567–584 (2003)
27. Zhang, Z., Xu, G.: A unified theory of uncalibrated stereo for both perspective and affine cameras. Journal of Mathematical Imaging and Vision 9, 213–229 (1998)
28. Hodges, S., Richards, R.: Uncalibrated stereo vision for pcb drilling. In: IEEE Colloquium on Application of Machine Vision (1995)
29. Park, J.S., Chung, M.J.: Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. TRA 19, 250–258 (2003)
30. Shimizu, Y., Sato, J.: Visual navigation of uncalibrated mobile robots from uncalibrated stereo pointers. In: Intl. Conf. on Pattern Recognition, pp. 346–349 (2000)
31. Cipolla, R., Hadfield, P.A., Hollinghurst, N.J.: Uncalibrated stereo vision with pointing for a man-machine interface. In: MVA, pp. 163–166 (1994)
32. Simond, N., Rives, P.: Trajectography of an uncalibrated stereo rig in urban environments. Intelligent Robots and Systems 4, 3381–3386 (2004)
33. Fusiello, A., Irsara, L.: Quasi-euclidean uncalibrated epipolar rectification. In: IEEE International Conference on Pattern Recognition (2008)
34. Bleyer, M., Gelautz, M.: Temporally consistent disparity maps from uncalibrated stereo videos. In: Image and Signal Processing and Analysis (2009)
35. Cheng, C.M., Lai, S.H., Su, S.H.: Self image rectification for uncalibrated stereo video with varying camera motions and zooming effects. In: MVA, pp. 21–24 (2009)
36. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: International Conference on Computer Vision (2007)
37. Min, D., Sohn, K.: Edge-preserving simultaneous joint motion-disparity estimation. In: IEEE International Conference on Pattern Recognition, pp. 74–77 (2006)
38. Zhang, Y., Kambhamettu, C.: On 3-D scene flow and structure recovery from multiview image sequences. Systems, Man, and Cybernetics 33, 592–606 (2003)
39. Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. In: International Conference on Computer Vision, pp. 722–729 (1999)
40. Trinh, H., McAllester, D.: Structure and motion from road-driving stereo sequences. In: 3D Information Extraction for Video Analysis and Mining (2009)
41. Pollefeys, M., Gool, L.V., Vergauwen, M., Cornelis, K., Verbiest, F., Tops, J.: Video-to-3D. In: ISPRS Commission V Symposium (2002)

3D Modeling of Haussmannian Facades

Chun Liu and André Gagalowicz

INRIA Rocquencourt, France

Abstract. Urban modeling has attracted many attentions since Google and Microsoft have launched their 3D geo softwares. However, in order to achieve photo-realistic results at the same level as for the latest interactive video games for high-end applications, less effort has been made to automate urban objects recognition and reconstruction. This paper consists of the automation of image-based Haussmannian facade recognition and reconstruction. The input image is firstly rectified and segmented in order to obtain a rectangular and less distorted facade image extracted from urban scenes. Then based upon various visual features and architectural knowledge, different facade elements which include windows, doors and balconies are detected including positions and also measured sizes. Combined with the depth information computed from 3D range data, the facade geometries and textures can be produced. Finally, an analysis-synthesis approach is used to reconstruct the full 3D facade representation. The processing pipeline developed for this research has also been verified and tested on various Parisian facades and confirmed the desired recognition and reconstruction results.

1 Introduction

Large scale 3D building modeling has attracted much attention recently in broad domains. With the advance of both hardware and software, building modeling can be applied to various applications including urban planning, cultural heritage preservation, entertainment and tourism.

With the rapid developments in imaging and sensing technologies, it is possible to obtain large urban models efficiently and cheaply. Aerial images can be used to estimate building volumes and to identify roofs and building footprints. Ground-based laser scanners can be used to unveil building facade surface in 3D. Ground taken facade image is used as the primary source for facade modeling analysis and provides facade 2D structures and textures.

However, current facade modeling has been limited to simpler and regular facades with perfect repetitions. Another limitation is that clean facade wall and less occlusions is required. Therefore for complicated and more realistic facades such as those of historical European cities with complex articulations of projecting and retreating objects, materials and textures, it is much harder and more challenging to automate the facade modeling from real photos.

This paper is dedicated to the modeling of 3D Haussmannian facades for urban reconstruction of Paris which is one of the most complex historical European

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 13–27, 2011.

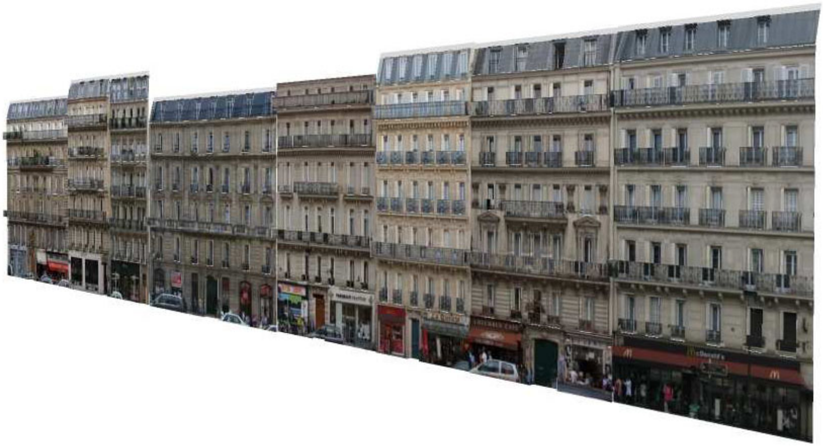


Fig. 1. Facades in Soufflot Street on the right side

cities. In Paris, Haussmannian buildings are widely spread and they are regarded as icons of neo-classic Paris. They present rather regular and consistent elements that are more suitable for analysis automation. Haussmannian buildings consist of multiple highly similar floors, significant repetition of architectural elements and well defined dimensions which are constrained to the street width. These characteristics meet the construction laws requirement and aesthetic perceptions. Therefore Haussmannian buildings exhibit a high degree of consistency. Together with building volumes and roof modeling, a full Paris urban model has to be obtained.

In our work, the facade geometries are recovered by a single view facade image analysis and depth estimation from terrestrial 3D laser data. Assuming all the facade elements are in parallelepipedic shapes, we are able to reconstruct the full 3D visual representation of facades.

The work presented here has been realized in the framework of the competitivity network TerraNumerica funded by the French government through ANR, its funding agency. The project consisted of a group of 18 partners including 10 companies and 8 research laboratories coordinated by THALES. The primary target was to create a 3D compact representation of the whole Paris urban area which would be suitable for various applications (tourism, flood control, urbanism, games...). The budget of this project reached 15 M euros and covered 3D compact representation of buildings as well as vegetation using procedural methods.

In this project we were supposed to dispose of rectified, segmented, georeferenced 2D Haussmannian facades, registered with 3D laser scanner data (work performed by some of our partners). As they were not made available during the period of our research, we had to perform everything mainly by hand beforehand. The work devoted to us by our consortium was to autmatically reconstruct in 3D these Haussmannnian facades from their inputs. This is the core of this paper.

1.1 Related Work

3D modeling from images has been a popular research topic in recent years. Great efforts have been made for 3D scene understanding by using state-of-the-art machine learning techniques [1]. Having multiple views, it is possible to reconstruct more delicate 3D building models [2]. Yet, most facade modeling researches thus far have been done on single view facade images. This is because obtaining multiple-view images are often impossible in urban environments. In addition near planar facade geometries and strong perspective distortions produce camera pose estimation problems.

In building facade recognition, two types of information are used, the visual likelihood and the prior. In most works, windows are considered as the key element in building facades which can help determine many high level information such as floor and tile splits. However, the visual likelihood of windows is rather ambiguous. One cannot model windows visual appearance in all cases. In [6], the author assumes windows are rectangles which could be easily extracted in clean facades without occlusions. In [8], windows are considered as image blob regions which can be characterized by Haar-like features. Other assumptions can be that windows are edge-framed rectangles [10] or are blob regions with different colors contrast from surrounding walls [9]. The prior information used in building facade recognition is perfect alignment of windows horizontally and vertically. This information can be used in two approaches. In [6], [10], the authors are proposing window rectangles and validate the rectangles according to the alignment at the same time. Or one can detect tiling first and then use the alignment of windows to validate windows detection in a top down manner [5], [9].

There are also building reconstruction research works on 3D laser scans to recover building facade surfaces and aerial photos to extract building volumes. However, those methods have not been proven to be useful in recovering the front visual information on the building facades. Therefore, the integration of building facades modeling, building volume extraction and 3D surface scans are highly needed for full urban reconstruction.

1.2 Organization of This Paper

The subsequent sections are organized as follows. Section 2 discusses Haussmannian facade analysis and describes the reconstruction pipeline. Section 3 is devoted to data pre-processings including data acquisition, 2D image rectification as well as 3D range data processing. Section 4 presents the 3D facade geometry extraction from images and range data. Then facade reconstruction is provided in section 5. Finally, results are given in section 6 and the conclusion is drawn in section 7.

2 Haussmannnian Facade and Its Reconstruction Pipeline

2.1 Haussmannian Facade Image Characteristics

Despite the highly organized structure in Haussmannian facades, there are several challenges in vision modeling of facades. Firstly, the Haussmannian facades are not planar as skyscrapers in Manhattan. Instead, Haussmannian facades have overhanging structures such as balconies, which consist of thin metal pieces with multiple holes in front of windows. These balconies create significant difficulties in vision recognition. They often block the view of windows. They cannot be recognized by specific visual features such as color, texture or shape directly. They do not own a volume so they could not be recovered through structure-from-motion (SFM) method. The balconies cannot be either recovered from 3D laser scans because the reflection of laser rays on multiple hole surfaces is random. Secondly, the Haussmannian facades are of large dimensions. Typically the facades are approximately 15 meters in height. According to this dimension, the depth of window retreatments which is 0.5 meter, is negligible. This way, images of Haussmannian facades become planar. Moreover, significant perspective distortions can be seen in facade images. These two problems basically restrict the use of structure-from-motion. Lastly, inhomogeneous window textures also make the direct image comparison impossible even though windows are highly repetitive, and needless to say, light variations and occlusions in urban environment make the vision modeling more complicated.

2.2 Pipeline

Considering all the difficulties in the multiple view-based or the 3D range data-based building reconstruction, we propose to use single view image-based reconstruction instead (see figure 2).

The facade images taken from streets are calibrated, rectified and segmented to get a rectangular and less distorted facade image region as pre-processing the single view image-based reconstruction process. (This was supposed to be performed by our partners in our consortium.) Then we analyse this rectilinear image region to extract the 2D rectangles corresponding to various facade elements such as windows. By assuming all the elements are in parallelepipedic shapes, we can unwrap the facade elements and produce the respective 2D textures. Next, 3D laser scan data is registered onto the 2D images and used to estimate depth for different facade elements. At the end, we use the CGA grammar [4] to reconstruct the facade geometry and map the 2D textures on it to produce a final compact 3D visual representation.

3 Data Pre-processing

Two data types are used to reconstruct the facades, 2D ground-taken facade images with digital single SLR cameras, and terrestrial laser scans. However,

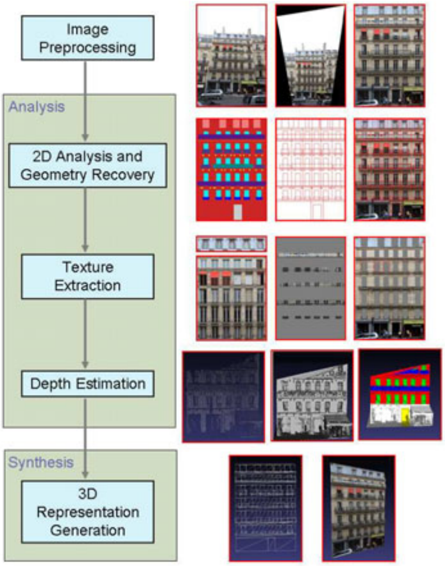


Fig. 2. 3D facade reconstruction processing flowchart

a few challenges have to be solved for the facade reconstruction process. One is due to the fact that there are several imperfections inside the data such as distortions. Second, the data acquired from urban environment consists not only of the facades but also of other urban objects. Therefore, pre-processing steps are needed to correct and clean the data from those imperfections and from environmental object noises.

3.1 Data Acquisition

Considering the light variations in urban areas, facade images should be taken when the building facades are relatively illuminated homogeneously and less occluded by vegetation. This is usually best taken in a cloudy morning, or in the evening, of later Spring or early Autumn. In addition, the image should be taken with wide angle lens so that the facade will be contained in one single image. These restrictive conditions ensure that the maximum details in the facade image can be preserved and different parts of the facade can be relatively easy to differentiate in terms of geometry, color and texture. The 3D laser data was provided by the French national geographic institute (IGN), and it is taken on a Google StreetView-like car from front scanning.

3.2 Pre-processing

In a 2D facade image, the first type of distortion is introduced by the lens. The radial lens distortion is considered as the main lens distortion source, which is a result of the shape of the lens. We use PTLens software [11] to reduce the lens distortion. This software features built-in lens profiles which are used to correct lens distortion. Once the images are calibrated, straight lines in the images become almost straight so that it is much easier to detect parallelism and intersections of lines for perspective rectification. The second type of distortion is the perspective distortion from the imaging process. In order to rectify images, we detect the line segments and estimate the horizontal and vertically vanishing points from which a 2D planar homography is further computed. Lastly, we manually extract the facade from undistorted image for 2D analysis.

Since the 3D scan data is continuous along the street, we manually segment the data with the help of the 2D images in order to extract individual facade data. Each facade data is re-scaled, oriented and positioned in a unique Cartesian coordinate system.

4 Facade Geometry and Texture Extraction

For a complete 3D building visual representation, building 3D geometry and texture have to be are mapped together. The 3D building geometry is obtained from ground-taken 2D images and 3D laser scans. Assuming all facade elements have parallelepipedic shapes, building textures can be produced by unwrapping the meshes into 2D facade images and extracting corresponding image regions.

4.1 2D Facade Analysis

Facade images contain comprehensive informations and are used as main data source for facade analysis. In 2D facade analysis, we use a top-down approach starting from tiling to the recognition of the different facade elements. At the end of the analysis, all facade elements are detected and parameterized as 2D rectangles on the facade plane.

Tiling. The most important feature of a facade is its tiling. Tiling describes how the facade is segmented horizontally and vertically. Each tile corresponds to a room of the apartments attached to the facades (except for the ground floor).

Tiling is also a high level semantic facade feature which cannot be computed directly although it is very helpful for facade analysis. We use window detections to determine the tiling. The hue information is selected as it occured that it best differentiates the windows from the facade walls. Then the detected window rectangles are validated to recover a rigid 2D lattice structure that indicates perfect horizontal and vertical window alignment, which is important in Haussmannian building typology.

The facade image is firstly segmented assuming the intensity of hue image follows two Gaussians distributions. One corresponds to facade walls and the other represents all non facade wall elements. The parameters of the two Gaussians are estimated by using EM (Estimation-Maximization) algorithm. Then the hue image is segmented in the Markov Random Field framework [3] and formulated as a pixel labeling problem. We use the estimated Gaussian mixture to compute a labeling cost. Suppose the facade wall color distribution is $N(\mu_{wall}, \sigma_{wall})$ and the other elements color distribution is $N(\mu_{others}, \sigma_{others})$. The cost of labeling wall pixel (color value v) as others is $N(v - \mu_{others}, \sigma_{others})$. Likewise, the cost of labeling pixel from others as wall, is $N(v - \mu_{wall}, \sigma_{wall})$. We compute the labeling difference between each pixel and its neighbourhood to maintain labeling consistency. It is computed as the sum of absolute label differences between each pixel and its four connecting neighbouring pixels (top, bottom, left and right). The minimization of the total energy is accomplished by using Belief Propagation. In the segmentation result, all elements other than facade wall are almost labeled as 0, including the roof and various shops on the ground floor.

From the segmentation, roofs are separated on the top and the window rectangles are extracted. Then window rectangles are validated by a 2D rigid structure. Subsequently connections are made between closest window rectangles and only window rectangles with more than two connections are left. The result from this connecting and validation is an incomplete window lattice structure. By assuming the perfect alignment of windows, missing window rectangles are recovered. Additionally, Haussmannian typology is checked to ensure no window penetrating into ground floors and roofs. Then the floor split is decided by window rectangle bottoms, and the tile split is done by the separation line between window rectangles. See figure 4.1 for the detailed process of window rectangle detection for tiling.

Normal Window Extraction. With tiling information, window detection is now limited to tiles which are defined by floor and tile separation lines. In addition, window counts are determined because only one window is allowed in every tile. Consequently, the window detection in 2D facade analysis is reduced to window dimension and position estimation.

Windows are parametrized as 2D rectangles inside each window tile. Because in the Haussmannian facades, windows are aligned on the floor bottom, we only need to fix three positions of window rectangles which are the top border and

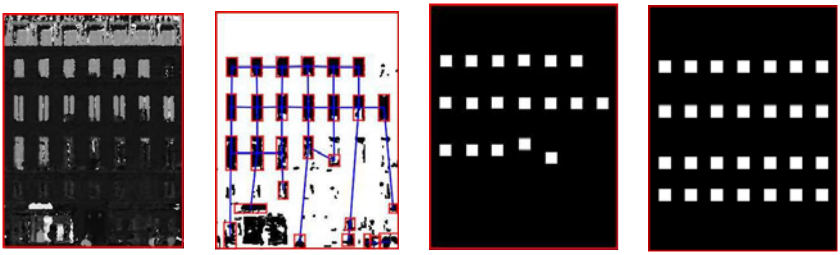
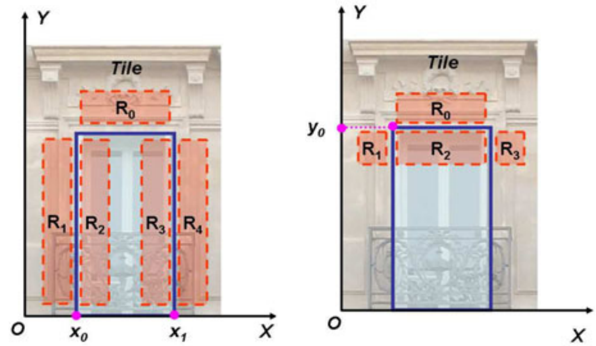

(a) Hue Image (b) Windows Rectangles Connection (c) Detected Windows Rectangles (d) Final Lattice Completion

Fig. 3. Hue-based tiling



(a) vertical border determination (b) top border determination

Fig. 4. Window border determination

two vertical borders. To decide the values of these three positions, we define two optimization energy functions by using histograms and edges. Equation 1 is used to detect the two vertical borders (see figure 4(a)) and equation 2 is used to fix the top border position (see figure 4(b)).

$$D(x_0, x_1) = \frac{d_{ssd}(R_1, R_2) + d_{ssd}(R_3, R_4)}{d_{ssd}(R_1, R_4)} \times \frac{d_{hist}(R_1, R_2) + d_{hist}(R_3, R_4)}{d_{hist}(R_1, R_4) + d_{hist}(R_1, R_0) + d_{hist}(R_4, R_0)} \qquad (1)$$

$$D(y_0) = \frac{d_{hist}(R_0, R_2) + d_{hist}(R_1, R_2) + d_{hist}(R_2, R_3)}{d_{hist}(R_1, R_0) + d_{hist}(R_3, R_0)} \times \frac{d_{ncc}(R_0, R_1) + d_{ncc}(R_0, R_3) + d_0}{d_{ncc}(R_0, R_2) + d_{ncc}(R_1, R_2) + d_{ncc}(R_3, R_2) + d_1} \qquad (2)$$

The detailed procedure for window detection is described below.

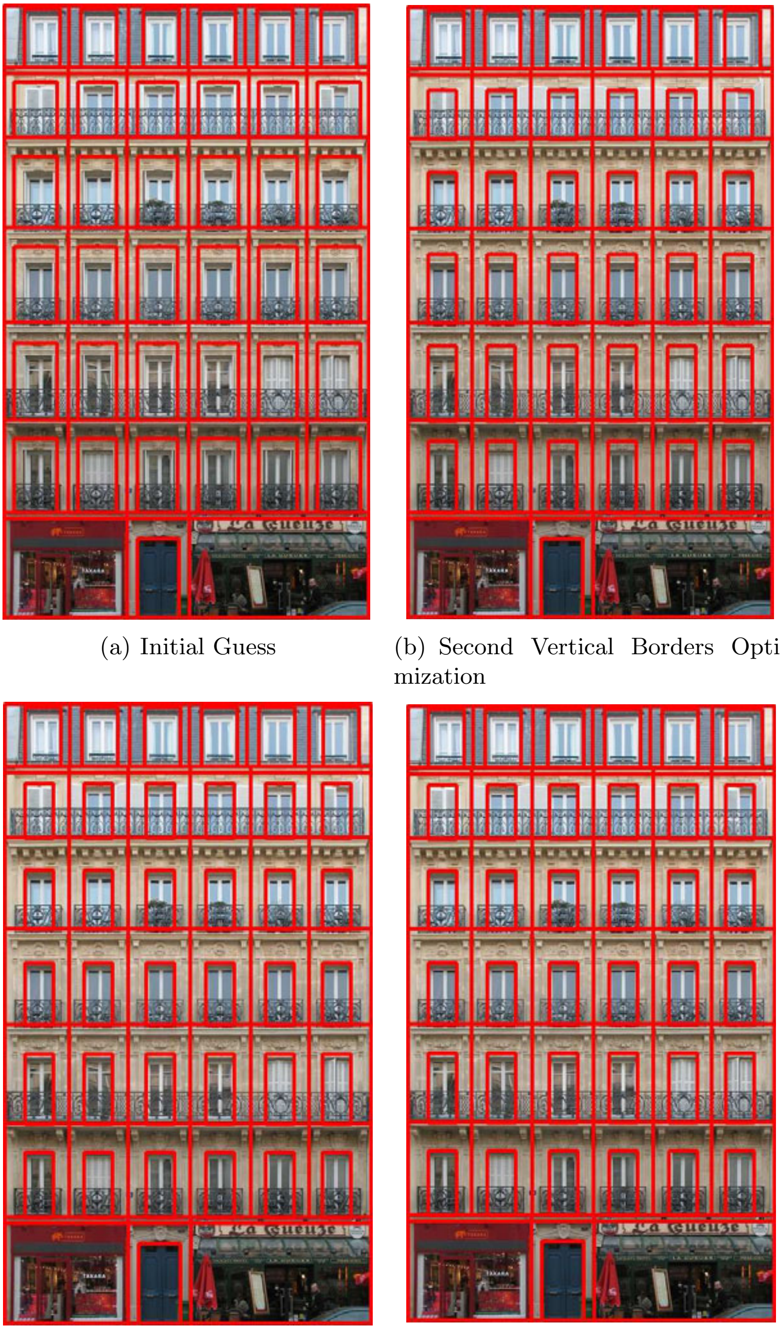
1. First, the initial position of the window rectangle is placed in the middle of the window tile, with the width set to be 2/3 of the total tile width and the height to be 4/5 of the total tile height. The window rectangle is aligned with the tile at the bottom border (see figure 5(a)).
2. Then, the top, left and right borders are refined by looking at the neighborhood for the strongest edges. Some of the window rectangles will be aligned to the real border.
3. Next, synchronization process is introduced to validate the detection. In the synchronization, all window rectangles are re-sized to have the same dimensions and re-aligned horizontally and vertically by averaging the positions and dimensions. This way, we can eliminate the least possible window rectangle positions and dimensions.

4. Furthermore, keeping the dimension constant, rectangles are moved from the present position horizontally to reach an optimal location using the contextual model. This move forces the image content inside the rectangle to be maximally different from the outside. After this step, the synchronization is applied again to keep the window rectangles in the same dimensions and aligned.
5. Similarly, another step finalizing the position and width is done by modifying separately the vertical borders of each window rectangle by a certain amount to reach local width optimization with the same contextual model. This way, the dimension and position is once again synchronized (see figure 5(b)).
6. Lastly, the optimal height is determined by using the same contextual model (see figure 5(c)). In addition, the tiling can be refined further by taking the middle positions between windows as the new vertical splits according to all the recognized window positions and dimensions.

Dormer Window Detection. Dormer windows are detected through edge detection and successive profiling. By using high-pass filter, we can highlight dormer window regions as well as small textures on the roof surface. A high-pass filter is defined by sliding a four-by-four window on the roof image and taking the variance. The contrast of this filtered image can be further enhanced by histogram equalization. Consequently, we can use image profiles (see figure 6) to detect dormer windows using their positions and dimensions. On the x profile, we can see pulse like patterns which correspond to the dormer windows. On the y profile, large values in the beginning and the end indicate the horizontal borders of dormer windows. This way, we can use signal processing to extract dormer windows.

Balcony Detection. In the wrought iron guard region, image pixels are switching from the iron guard to another content behind it. As a result, the image contains high spatial frequencies. In contrast, the luminance and texture change are rather small in the facade wall regions so that the facade wall image regions contain low spatial frequencies. Thus, we can consider that on each image floor two types of image regions exist with different spatial frequencies. Then the rectangular wrought iron can be extracted by segmenting the image floor by spatial frequencies.

The process starts from binary image segmentation using the spatial frequency energy. The result from the segmentation separates the potential balcony regions from facade walls. Next those regions are extracted as rectangles by profiling. As the segmentation is sensitive to a single feature, several problems exist on those rectangular regions. Balcony detection can be broken due to dark parts on windows (loss of image details). As a consequence, balcony dimension can be wrong. Those errors are corrected in further validation process which invokes the structural priors. In this validation process, all wrought iron regions are checked according to various empirical rules. At the end, the rectangular wrought iron regions are detected in place.



(a) Initial Guess

(b) Second Vertical Borders Optimization

(c) Top Border Optimization

(d) Window Detection with Refined Tiling

Fig. 5. Iterative window extraction

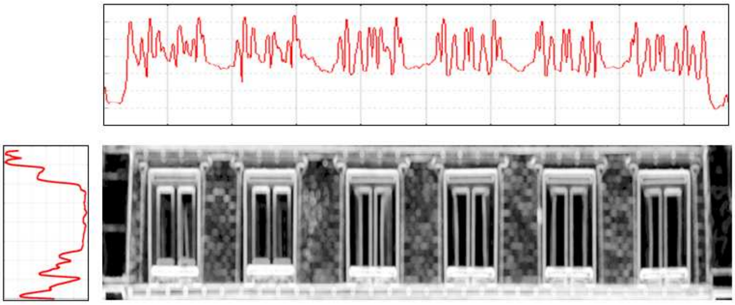


Fig. 6. Edge profiling

Door Detection. Door is detected by using both edge and color information. In the beginning, Canny edge detection is used to detect vertical edges to produce door rectangle candidates. By knowing that the door having a width of one or two tile widths and door position is only close to the middle, or to the left and to the right of the ground floor, we can filter out a lot of improper hypotheses. Then we use color histogram intersection to select the valid door region from rectangle candidates. A previously established small door database is used. Because doors in Haussmannian facades are painted in limited number schemes due to construction regulation, this small database was sufficient.

4.2 3D Depth Estimation

From the 2D analysis, all facade elements are detected as 2D rectangles. In order to obtain the full 3D geometry, the depth information is required. Therefore, we need to use ground-taken laser scans to estimate depth for the various facade elements.

We first manually register the 3D scans with the facade images through a raster depth map image produced from Delaunay triangulation on 3D point clouds. After the registration, different facade object 2D masks are generated to segment the 3D point clouds in the X-Y plane. Thus the depth is estimated by using median values in the Z direction.

4.3 Shape Prior and Facade 3D Geometry

We assume various facade elements have a parallelepipedic shape. For example, the window 3D geometry is modeled as a parallelepipedron without the front face. Thus, by combining the 2D rectangles and the depth information, the full 3D facade geometry is produced by assembling individual terminal elements such as windows, walls, and others together.

4.4 Texture Extraction

To finalize the building facade visual representation, textures are needed to be mapped to the various facade elements. Because Haussmannian facades are not piece-wise planar due to the balconies, we need to process the facade image to get unoccluded facade images for facade walls and windows. We assume the facade walls and windows are symmetric vertically, thus we can copy the unoccluded image region above the balconies and tile them over the occluded regions. This way we achieve a simple version of in-painting to produce photo-realistic textures. Because we assume parallelepipedic shapes for various terminal facade elements, we can then project the geometries from 3D to 2D on the processed facade images and compute UV texture coordinates. Thus, all facade elements are texture-mapped.

5 Facade Reconstruction

A facade 3D visual representation is a semantically and hierarchically organized data sets including facade geometry and texture coordinates. Such representation is valuable for various semantic building information processing. To obtain such visual representation, we use CGA [4] repeat split grammars to reconstruct repetitive building facades and adapt this synthetic building facades to real dimensions through an Analysis-Synthesis approach.

5.1 CGA Grammar-Based Reconstruction

We implemented CGA grammar introduced by Pascal Müller et al. in 2006 [4]. In this grammar modeling tool, an initial building shape is progressively split into sub-spaces through shape derivation process using different split production rules. At the end of the derivation, a hierarchical shape tree is produced where each node represents an oriented and scaled shape occupying a small space with a semantic name. All the terminal leaves are terminal shapes which are replaced with terminal elements meshes (windows, doors, balconies, etc.). By extracting all the terminal nodes and assemble geometries together, a 3D building model is produced.

5.2 Reconstruction by Analysis-Synthesis

We automatically generate facade description from facade analysis results by using repetitive split rules so that a regular synthetic building facade is produced which we call generic facade model. The use of the repetitive splits rules is for reducing the number of specific split rules. Then we can adapt this generated building representation in memory for changing the repetitive split to specific split (non repetititve) to produce the final building model which we call specific model. By using this Analysis-Synthesis approach, we achieve both building information compression with repetitive splits and the accurate dimension recovery with specific splits.

The facade adaptation is done in two steps. Firstly, we project the 3D facade mesh in 2D and establish characteristic point correspondences between this projected 2D geometry and 2D analysis. By comparing their location differences, we can compute RBF (Radial Basis Function) transformation so that shapes in the

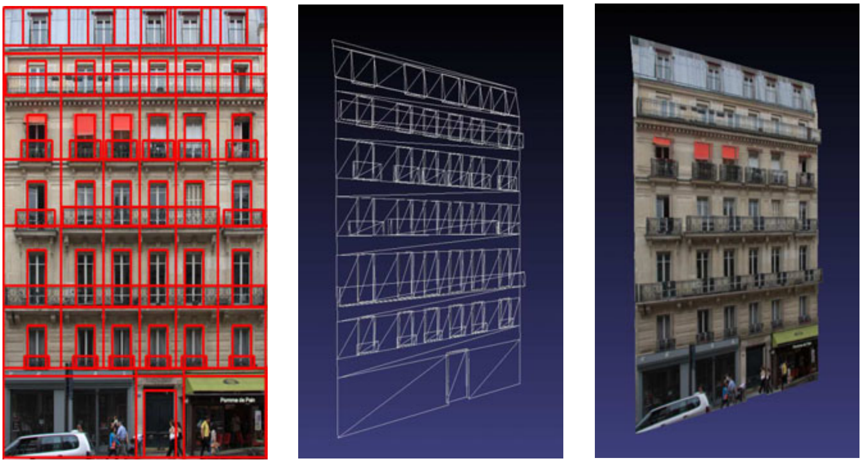

(a) 2D Analysis Result (b) Wireframe (c) Textured Model

Fig. 7. 3D facade synthesis

shape tree are changed on their split rules in the memory. A second synthesis is then invoked in memory but only on the affected shapes hence a modified building 3D model is obtained. In the second step, we need to adjust all the overlay shapes such as balconies which are not attached to any split rules from original facade plane. By completing this two-step adaptation, a generic facade model is transformed into a specific facade model. The final 3D facade model is shown in figure 7.

6 Results and Conclusion

We applied our proposed processing pipeline on all fifteen facades of the Soufflot street in Paris. The right side facade 3D model is presented in figure 1. And the left side 2D analysis is shown in figure 8. For each facade, the processing time is around 40 minutes starting from image calibration, through perspective rectification, facade extraction, 2D analysis, depth inference and texture processing until the grammar-based reconstruction. In order to evaluate the tiling and window detection, we manually selected 100 facades from various boulevards, avenues and streets in Paris. 64 of 100 facades have been successfully automatically processed with all windows detected in places.

We also evaluated different processing steps with various metrics. For example, the 2D detection of windows can be accessed by computing the overlapping rate between manually annotated ground truth and detection instances. Due to the limitation on the numbers of pages published on this paper, we will present them during the oral presentation.

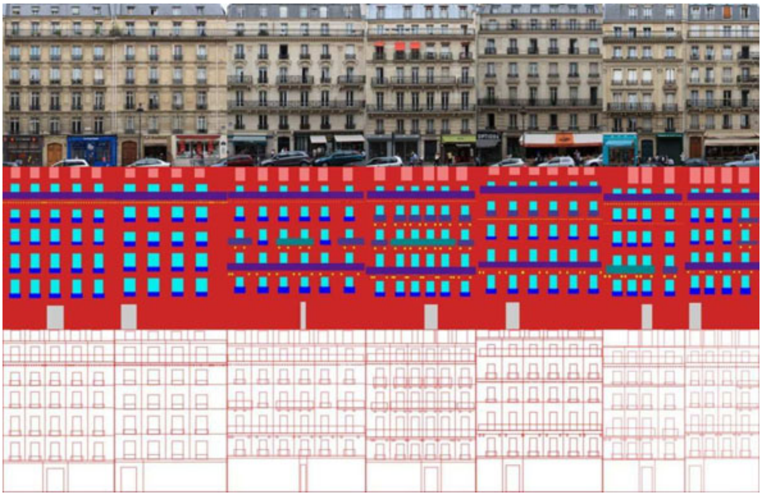


Fig. 8. Facade analysis on left handside Soufflot Street

The main contributions of this paper are three-folds. The first contribution is that we have proven that a fully automatic image-based 3D building modeling is feasible for large scale urban reconstructions. The second contribution is that we have succeeded to produce both semantic analysis and synthesis. The last contribution is that we have used Analysis-Synthesis approach to achieve both the flexibility of grammar-based reconstruction and reconstruction accuracy. Of course, this work is a starting point for large scale urban reconstruction. There are still many challenging problems ahead. For facades with large lighting variations and significant occlusions, we still have not robustly extracted the building geometry. In addition, our algorithm is implemented by using Python which runs longer than pure C++ optimized code. We expect to continue our work in these two directions.

Acknowledgement. The author wish to thank Romain Janil for his great work on the characterization of Haussmannian buildings. Also we are grateful for the invaluable guide from Cédric Guiard and the technical helps from HongPing Yan and Emile Lombardo. The work represented here in the paper has been taken as part of the Cap Digital Terra Numerica competitivity network under the support of INRIA Rocquencourt.

References

1. Hoiem, D., Efros, A.A., Hebert, M.: Geometric Context from a Single Image. In: International Conference of Computer Vision, pp. 654–661 (2005)
2. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. ACM Trans. Graph. 25(3), 835–846 (2006)

3. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. IEEE Trans. Pattern Anal. Mach. Intell. 30(6), 1068–1080 (2008)
4. Müller, P., Wonka, P., et al.: Procedural Modeling of Buildings. In: ACM SIGGRAPH 2006, vol. 25, pp. 614–623 (2006)
5. Müller, P., Zeng, G., et al.: Image-based Procedural Modeling of facades. In: ACM SIGGRAPH 2007, pp. 181–184. IEEE Press, New York (2007)
6. Korah, T., Rasmussen, C.: 2D Lattice Extraction from Structured Environments. In: ICIP (2), pp. 61–64 (2007)
7. Hays, J., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering Texture Regularity as a Higher-Order Correspondence Problem. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 522–535. Springer, Heidelberg (2006)
8. Šochman, J.: Evaluation of the AdaBoost IPM. Technical Report TN-eTRIMS-CMP-01-2007 (2007)
9. Chun, L., Gagalowicz, A.: Image-based Modeling of Haussmannian Facades. The International Journal of Virtual Reality 9 (2009)
10. Tyleček, R., Šára, R.: A Weak Structure Model for Regular Pattern Recognition Applied to Facade Images. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 450–463. Springer, Heidelberg (2011)
11. Niemann, T.: PTLens, http://epaperpress.com/ptlens/

Lung Cancer Detection from Thoracic CT Scans Using 3-D Deformable Models Based on Statistical Anatomical Analysis

Hotaka Takizawa and Shigeyuki Ishii

University of Tsukuba, 305-8573, Japan
takizawa@cs.tsukuba.ac.jp
http://www.pr.cs.tsukuba.ac.jp

Abstract. In the present paper, we propose a novel recognition method of pulmonary nodules (possible lung cancers) in thoracic CT scans. Pulmonary nodules and blood vessels are represented by 3-D deformable spherical and cylindrical models. The validity of these object models are evaluated by the probability distributions that reflect the results of the statistical anatomical analysis of blood vessel trees in human lungs. The fidelity of the object models to CT scans are evaluated by five similarity measurements based on the differences in intensity distributions between the CT scans and templates produced from the object models. Through these evaluations, the posterior probabilities of hypotheses that the object models appear in the CT scans are calculated by use of the Bayes theorem. The nodule recognition is performed by the maximum a posterior estimation. Experimental results obtained by applying the proposed method to actual CT scans are shown.

Keywords: Detection of lung cancers, Thoracic CT scans, Computer-aided diagnosis, Statistical anatomical analysis, 3-D deformable object models.

1 Introduction

Lung cancer is the most common cause of death among all cancers worldwide [1]. To cope with this serious situation, mass screening for lung cancer was widely performed by simple X-ray films with sputum cytological tests. However, it is known that the accuracy of this method is not sufficient for early detection of lung cancer [2, 3]. Therefore, a lung cancer screening system by computed tomography (CT) for mass screening is proposed [4]. This system improves the accuracy of the cancer detection considerably [5], but has one problem, that is, the number of the images is increased to over dozens of slice sections per patient from 1 X-ray film. It is difficult for a radiologist to interpret all the images in a limited time. In order to make the system more practical, it is necessary to build a computer-aided diagnosis (CAD) system that automatically detects abnormal regions suspected to comprise pulmonary nodules that are the

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 28–39, 2011.

major radiographic indicators of lung cancers, and informs a radiologist of their positions in CT scans as a second opinion.

Extensive research has been dedicated to automated detection of pulmonary nodules in thoracic CT scans [6]. Morphological [7] image filters [8–10] are conventional approaches. In the work[11], a multiple-thresholding technique was used to detect nodules that had peaked intensity distribution. Hessian-based image filters [12, 13] individually enhanced blob- and ridge-shaped regions that corresponded to nodules and blood vessels, respectively. These methods were often used for initial detection of nodules, and were intentionally adjusted to minimize the number of misdetection. Consequently, they yielded many false candidates called false positives (FP) that corresponded to normal pulmonary structures such as blood vessels.

In order to reduce false positives, feature-based discrimination methods between nodules and false positives have been also developed [14–17]. Kawata, *et al.* reported a classification method [18] of nodules based on differences in shape indexes, which were computed from two principal curvatures of intensity profiles in images, between nodules and false positives. Suzuki, *et al.* proposed a method [19] that suppressed false positives by using voxel values in regions of interest as input for a novel artificial neural network [20].

Model-based methods are the promising approaches as well.[1] Several works with nodule models were reported. Lee, *et al.* [22] proposed a template-matching method using nodular models with two-dimensional (2-D) Gaussian distribution as reference images to CT scans. The method had an advantage of being able to use the geometrical features of nodules as well as gray level features. Ozekes, *et al.* [23] designed a 3-D prismatic nodule template that was composed of several layered matrices. Farag, *et al.* [24] developed 3-D deformable nodule models such as spherical models of various radii. These methods can make use of the characteristics of the 3-D relation between a suspicious region in a slice section and the other regions in the adjacent slice sections.

In the present paper, we propose a novel recognition method of pulmonary nodules in thoracic CT scans. Pulmonary nodules and blood vessels, which often yield false nodules on CT scans, are represented by 3-D deformable spherical models, cylindrical models and their combination models. The recognition method proposed in this paper is based on [25], but has newly introduced the following two evaluation techniques concerning the object models. First, the validity of the blood vessel models are evaluated by the probability distributions that reflect the results of the statistical anatomical analysis of blood vessel trees in human lungs. Second, the fidelity of the object models to CT scans are evaluated by five similarity measurements based on the differences in intensity distributions between the CT scans and templates produced from the object models. Through these evaluations, the posterior probabilities of hypotheses that the object models appear in the CT scans are calculated by use of the Bayes theorem. The nodule recognition is performed by the maximum a posterior (MAP) estimation.

[1] A considerable work[21] has been also done to detect breast cancers by eliminating false positives due to liner structures in X-ray mammograms.

2 3-D Deformable Object Models

The method[25] represents nodules and blood vessels as 3-D geometrical object models as described below.

A nodule is represented as a sphere model as shown in Figure 1(a). The priori probability of appearance of the nodule model o^N (the superscript "N" is the label that means a nodule) is defined by

$$p(o^N) = P^N \; g(r^N), \tag{1}$$

where P^N is the probability of the set of all possible nodules, r^N is the radius of the nodule model, and $g(r)$ is a Gaussian distribution.

A curved section in a blood vessel tree is represented as a two-connected-cylinder model as shown in Figure 1(b). The priori probability of appearance of the curved blood vessel model o^{B_c} is defined by

$$p(o^{B_c}) = P^{B_c} \; \{g(r_1^{B_c}) \; g(\delta^{B_c}) \; g(\psi^{B_c})\}^{\frac{1}{3}}, \tag{2}$$

where P^{B_c} is the probability of the set of all possible curved blood vessels, $r_i^{B_c}$ is the radius of the i-th cylinder ($i = 1, 2$), δ^{B_c} is a difference in section area between the cylinders, and ψ^{B_c} is an angle between the cylinders. The exponent $\frac{1}{3}$ is introduced for normalizing the number of the Gaussian terms between Equations (1) and (2).

A bifurcation in a blood vessel tree is represented as a three-connected-cylinder model as shown in Figure 1(c). The priori probability of appearance of the bifurcated blood vessel model o^{B_b} is defined by

$$p(o^{B_b}) = P^{B_b} \; \{g(r_1^{B_b}) \; g(\delta_{23}^{B_b}) \; g(\delta_{123}^{B_b}) \; g(\psi_{12}^{B_b}) \; g(\psi_{13}^{B_b}) \; g(\psi_{23}^{B_b})\}^{\frac{1}{6}}, \tag{3}$$

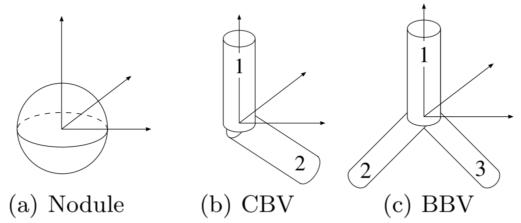
where P^{B_b} is the probability of the set of all possible bifurcated blood vessels, $\delta_{23}^{B_b}$ is the difference in section area between the two child cylinders, $\delta_{123}^{B_b}$ is the difference in section area between the parent and child cylinders, that is $\delta_{123}^{B_b} = \pi \cdot \{(r_1^{B_b})^2 - (r_2^{B_b})^2 - (r_3^{B_b})^2\}$, and $\psi_{ij}^{B_b}$ is the angle between the i-th and j-th cylinders. The exponent $\frac{1}{6}$ is also introduced for the normalization.

2.1 Combination Models of Nodules and Blood Vessels

Two types of object model are newly introduced:

1. a combination model o^{NB_c} that consists of a nodule model and a curved blood vessel model, and
2. another combination model o^{NB_b} that consists of a nodule model and a bifurcated blood vessel model.

Figure 2 shows the procedure of generating the combination models. First, the optimal nodule, curved and bifurcated blood vessel models are individually searched for in the same way as [25]. The optimal combination models o^{NB_c} and o^{NB_b} are generated by combining the optimal models and then are stored



(a) Nodule (b) CBV (c) BBV

Fig. 1. 3-D object models that represent nodules, curved blood vessels (CBV) and bifurcated blood vessels (BBV), respectively

in the *combination models* list that is depicted at the bottom of Figure 2. The priori probabilities of appearance of each combination model are calculated by

$$p(o^{NB_c}) = P^{NB_c} \left\{ g(r^N)\ g(r_{r1}^{B_c})\ g(\delta^{B_c})\ g(\psi^{B_c}) \right\}^{\frac{1}{4}}, \tag{4}$$

$$p(o^{NB_b}) = P^{NB_b} \left\{ g(r^N)\ g(r_{r1}^{B_b})\ g(\delta_{23}^{B_b})\ g(\delta_{123}^{B_b}) \right. \\ \left. g(\psi_{12}^{B_b})\ g(\psi_{13}^{B_b})\ g(\psi_{23}^{B_b}) \right\}^{\frac{1}{7}}. \tag{5}$$

Some of the generated combination models have the singular arrangement of objects. The bottom right model indicated by the "*" mark in Figure 2 is an example. The nodule model is almost embedded in the blood vessel model. Such a combination model can be matched to a normal blood vessel region in CT and thus it can be mistakenly determined to be a nodule. To avoid such mistakes, the combination models that include such embedded models are eliminated from the combination model list.

2.2 Statistical Anatomical Modeling of Blood Vessels in Lungs

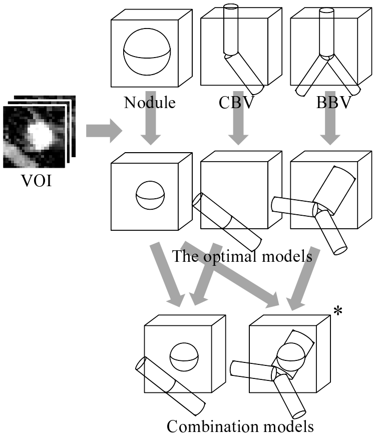


Fig. 2. The procedure of generating the combination models

The Gaussian distribution terms $g(r_1^B)$ in Equations (2), (3), (4) and (5) implicitly contain the mean radius $\mu_{r_1^B}$ and the standard deviation σ_{r1}^B. Generally, blood vessels are thick in the central areas of lungs, and they become thinner as they approach the peripheral areas. $\mu_{r_1^B}$ and σ_{r1}^B should be defined based on such anatomical relations. In this paper, 3-D matrices, namely *distribution models*, are introduced that are composed of the statistical values obtained by measuring blood vessels at each position in lungs. Here, the distribution models of means and standard deviations of blood vessel radii are constructed.

Distribution Model of Means of Blood Vessel Radii. Figure 3 shows the top level block diagram of the construction procedure of the distribution model of means of blood vessel radii. First, we extract blood vessel regions from i-th CT scan V^i. Their radii are measured by applying the distance transform process to the extracted regions, and their centerlines are extracted by the thinning process[26]. Each pixel value on the centerline in the processed images represents the radius of the blood vessel at its position in a lung. Let R^i denote the image obtained from V^i. Next, we settle a boundary box so as to circumscribe the lung region in V^i, and divide it into $L \times M \times N$ smaller boxes that are called "cells". The l-, m-, n-th cell in V^i is denoted by $C^i_{l,m,n}$. The j-th radius value on the centerline in $C^i_{l,m,n}$ is denoted by $u^{i,j}_{l,m,n} \in U^i_{l,m,n}$. Several cells include tissues other than blood vessels such as diaphragms. They can be erroneously recognized as too thick or thin blood vessels. Estimation of the mean radius would suffer from such cells. In order to minimize their effects, we calculate the mean radius of each cell as follows:

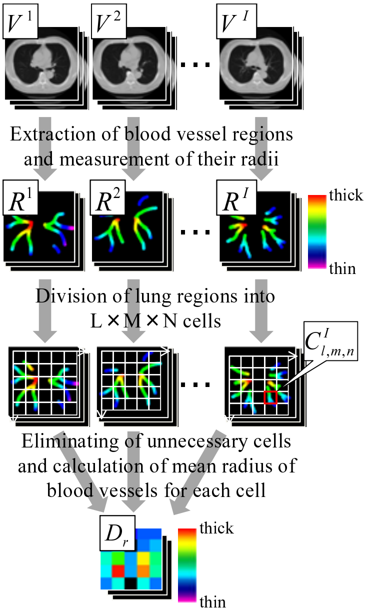

V^1
V^2
V^I
Extraction of blood vessel regions and measurement of their radii
R^1
R^2
R^I
thick
thin
Division of lung regions into L × M × N cells
$C^I_{l,m,n}$
Eliminating of unnecessary cells and calculation of mean radius of blood vessels for each cell
D_r
thick
thin

Fig. 3. Construction procedure of the distribution model of means of blood vessel radii

$$\overline{u^i_{l,m,n}} = \frac{\sum\limits_{U^i_{l,m,n}} u^{i,j}_{l,m,n}}{\#\{U^i_{l,m,n}\}} \tag{6}$$

where $\#\{U^i_{l,m,n}\}$ is the number of radius values. We sort the cells at each position, i.e. $C^1_{l,m,n}$, $C^2_{l,m,n}$,..., $C^I_{l,m,n}$, by the mean radius values, and eliminate the cells that have the N_c largest and smallest radius values. Let $D_r(l,m,n)$ denote a value at the position l, m and n in the matrix of the distribution model. $D_r(l,m,n)$ is calculated by

$$D_r(l,m,n) = \begin{cases} \dfrac{\sum\limits_{i,j} u^{i,j}_{l,m,n}}{N^u_{l,m,n}} & \left(\dfrac{N^c_{l,m,n}}{I} \le T_p\right) \\ 0 & (otherwise), \end{cases} \tag{7}$$

where T_p is a threshold value, $N^u_{l,m,n}$ and $N^c_{l,m,n}$ are the numbers of the remaining radius values and cells, respectively.

Distribution Model of Standard Deviations of Blood Vessel Radii. As is the case with the mean distribution model, the standard deviation at the position l, m and n in the matrix of the distribution model is calculated by

$$D_{sd}(l,m,n) = \\ \begin{cases} \sqrt{\dfrac{\sum\limits_{i,j} \left\{u^{i,j}_{l,m,n} - D_r(l,m,n)\right\}^2}{N^u_{l,m,n} - 1}} & \left(\dfrac{N^c_{l,m,n}}{I} \le T_p\right) \\ 0 & (otherwise). \end{cases} \tag{8}$$

3 Observation Models

The fidelity of an object model to a VOI in an observed CT image is evaluated by using a similarity measurement between the CT VOI and a template produced from the object model. The method[25] uses the following normalized correlation coefficient (NCC) :

$$\gamma_c(v_c, v_t) = \frac{\sum\limits_{x,y,z} \Big(v_c(x,y,z) - \bar{v}_c\Big)\Big(v_t(x,y,z) - \bar{v}_t\Big)}{\sqrt{\sum\limits_{x,y,z} \Big(v_c(x,y,z) - \bar{v}_c\Big)^2} \sqrt{\sum\limits_{x,y,z} \Big(v_t(x,y,z) - \bar{v}_t\Big)^2}}, \tag{9}$$

where $v_c(x,y,z)$ and $v_t(x,y,z)$ are the voxel value at x,y,z in the CT VOI and the template, respectively. $\bar{v}_c$ and $\bar{v}_t$ are the mean voxel values of the CT VOI and the template, respectively.

Using $\gamma_c(v_c, v_t)$, the likelihood function is defined by

$$p_c(v_c|o) = \frac{\gamma_c(v_c, v_t) + 1}{2}. \tag{10}$$

In this paper, the following four likelihood functions are newly introduced.

3.1 Sum of Absolute Distance

The likelihood function based on the sum of absolute distances (SAD) is defined by

$$p_s(v_c|o) = -\frac{\gamma_s(v_c, v_t)}{\kappa_s}, \tag{11}$$

where

$$\gamma_s(v_c, v_t) = \sum_{x,y,z} \Big| v_c(x, y, z) - v_t(x, y, z) \Big|, \tag{12}$$

and κ_s is set to be a maximum value of $\gamma_s(v_c, v_t)$.

3.2 Mutual Information and Normalized Mutual Information

From a CT VOI and a template, the histograms of pixel values $h_c(v)$ and $h_t(v)$ are obtained, respectively, and their entropy values H_c and H_t are calculated, respectively. From the simultaneous histogram obtained from the CT VOI and the template $h_{ct}(v_c, v_t)$, the joint entropy value H_{ct} is also calculated.

The mutual information (MI) and normalized mutual information (NMI) are calculated by

$$\gamma_m(v_c, v_t) = H_c + H_p - H_{cp}, \tag{13}$$

$$\gamma_n(v_c, v_t) = \frac{H_c + H_p}{H_{cp}}, \tag{14}$$

respectively. From these values, their likelihoods are calculated by

$$p_m(v_c|o) = \frac{\gamma_m(v_c, v_t)}{\kappa_m}, \tag{15}$$

$$p_n(v_c|o) = \frac{\gamma_n(v_c, v_t)}{\kappa_n}, \tag{16}$$

respectively. κ_m and κ_n are set to be maximum values of $\gamma_m(v_c, v_t)$ and $\gamma_n(v_c, v_t)$, respectively.

3.3 L0-norm

The likelihood function based on the L0-norm (L0) is defined by

$$p_l(v_c|o) = \frac{\gamma_l(v_c, v_t)}{N_l}, \tag{17}$$

where

$$\gamma_l(v_c, v_t) = \sum_{x,y,z} \delta \Big| v_c(x, y, z) - v_t(x, y, z) \Big|, \tag{18}$$

$$\delta(x) = \begin{cases} 1 & (x < T_l) \\ 0 & (otherwise). \end{cases} \tag{19}$$

N_l is the number of pixels in the CT VOI and T_l is a threshold.

4 Recognition of Nodules and Blood Vessels Based on MAP Estimation

Using the Bayes theorem, the posterior probability of an object model o given a CT VOI v_c is defined by

$$p(o|v_c) = \alpha \; p(v_c|o) \; p(o), \tag{20}$$

where $\alpha = [p(v_c)]^{-1}$. $p(v_c|o)$ is the likelihood obtained from Equations (10), (11), (15), (16) or (17). $p(o)$ is a priori probability obtained from Equations (1), (2), (3), (4) or (5).

The following evaluation value:

$$\rho(v_c) = \frac{\max\limits_{\tau \in \{N, NB_c, NB_b\}} p(o^\tau|v_c)}{\max\limits_{\tau \in \{B_c, B_b\}} p(o^\tau|v_c)} \tag{21}$$

is calculated. The CT VOI v_c is determined to contain a nodule if the evaluation value is larger than a threshold, and vice versa.

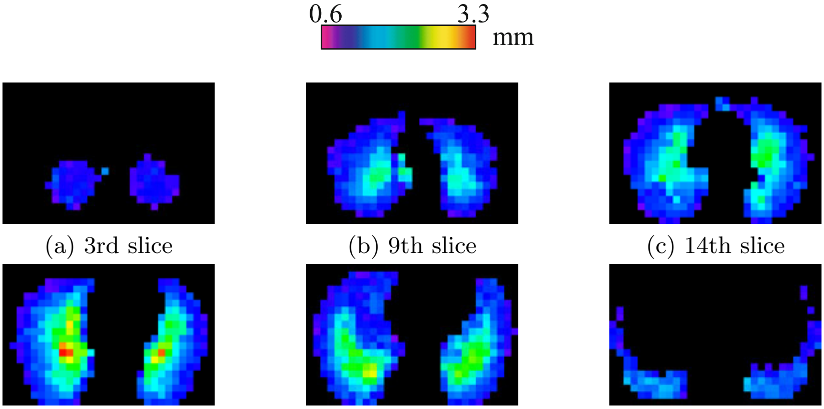
5 Experimental Results

5.1 Construction of Distribution Models of Pulmonary Blood Vessels

We construct the distribution models of pulmonary blood vessels from 48 healthy thoracic CT scans. The CT scans contain from 31 to 44 slice cross sections. The parameters of our improvements are determined as follows. The division numbers L, M and N are 30, 20 and 31, respectively. The distribution models of means and standard deviations of blood vessel radii are shown in Figures 4 and 5, respectively. Figure 4(c) and (d) show that thick blood vessels are distributed in the central area in lungs, and thin blood vessels are distributed in the peripheral area. The distribution shown in Figure 4(e) is more important. Thick blood vessels do not necessarily gather around the center of a lung. They are on the dorsal area of a lung.

5.2 Performance Comparison

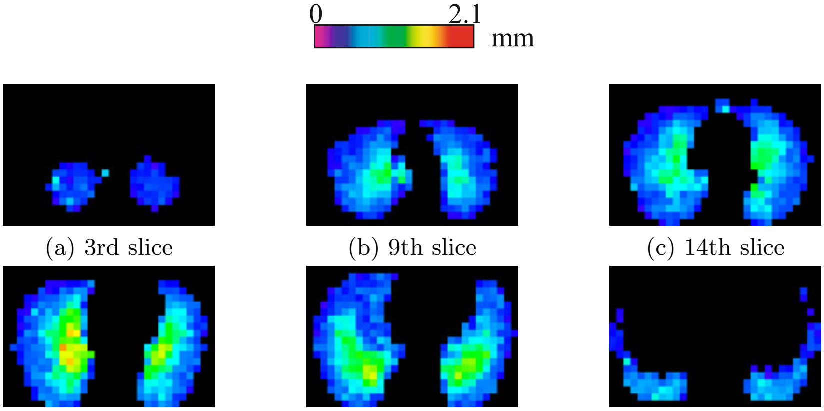
We compare the performance of the likelihood functions in Section 3. By applying our initial detection method[27] to actual 98 CT scans including 98 pulmonary nodules identified by a radiologist, 96 nodules are detected with 56.4 FPs per scan (two false negatives occur). The VOIs are extracted which include the initial detected nodule candidates inside, and are fed into the proposed method as the CT VOI v_c in Equation (20). The VOI sizes are 15 × 15 pixels in the x-y slice planes and 3 slices in the z (so-called *body*) axes. Figure 6 shows the free-response receiver operating characteristic (FROC) curves that represent the relation between the true positive ratio and the number of false positives per scan. The



(a) 3rd slice (b) 9th slice (c) 14th slice

(d) 19th slice (e) 24th slice (f) 30th slice

Fig. 4. The distribution model of means of blood vessel radii



(a) 3rd slice (b) 9th slice (c) 14th slice

(d) 19th slice (e) 24th slice (f) 30th slice

Fig. 5. The distribution model of standard deviation of blood vessel radii

curves indicated by NCC, SAD, MI, NMI and L0 represent the performance of the likelihood functions of the Equations (10), (11), (15), (16) and (17), respectively, in Section 3.

At the true positive ratio of 100%, the minimum (best) number of false positives is 19.2 per scan, that is obtained by SAD, and the second minimum number is 22.3 by NCC. However, at the true positive ratio of 80%, the minimum FP number is 3.3 by NCC and the second minimum FP number is 3.5 by NMI. The FP number of the SAD is 4.5 at this true positive ratio. The performance order of the five likelihood functions varies depending on the aimed true positive ratio. At the sensitivity of 100%, the specificity of the results of SAD, which is the best likelihood function at this sensitivity, against that of the initial detection is

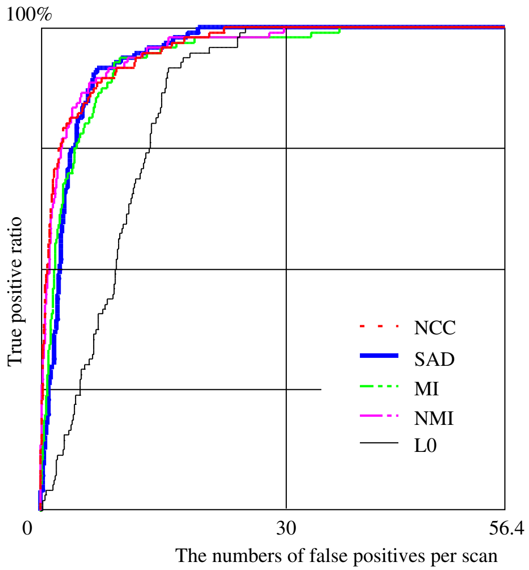


Fig. 6. The FROC curves

about 34.0% (= 19.2/56.4). The FP number of 19.2 per scan is almost equivalent to that of 0.62 per slice and is less than the so-called practical upper limit 2 per image[28].

6 Conclusion

In the present paper, we propose a novel recognition method of pulmonary nodules in thoracic CT scans. Pulmonary nodules and blood vessels are represented by 3-D deformable spherical and cylindrical models. The validity of the object models are evaluated by the probability distributions that reflect the results of the statistical anatomical analysis of blood vessel trees in human lungs. The fidelity of the object models to CT scans are evaluated by five similarity measurements. Through these evaluations, the posterior probabilities of hypotheses that the object models appear in the CT scans are calculated by use of the Bayes theorem. The nodule recognition is performed by the MAP estimation. Comparison of performance of the five similarity measurements are performed.

In this paper, the distribution models are defined with respect only to the means and standard deviations of blood vessel radii. Ones of our future works for improving the recognition accuracy are to define the distribution models of nodules and to use other statistics such as medians.

References

1. Weir, H.K.: Annual report to the nation on the status of cancer, 1975-2000. Journal National Cancer Institute 95(17), 1276–1299 (2003)
2. Tanaka, T., Yuta, K., Kobayashi, Y.: A study of false-negative case in mass-screening of lung cancer. Jay.J.Thor.Med. 43, 832–838 (1984)
3. Oda, J., Akita, S., Shimada, K.: A study of false-negative case in comparative reading of mass-screening of lung cancer. Lung Cancer 29, 271–278 (1989)
4. Yamamoto, S., Tanaka, I., Senda, M., Tateno, Y., Iinuma, T., Matsumoto, T., Matsumoto, M.: Image processing for computer-aided diagnosis of lung cancer by CT(LSCT). Systems and Computers in Japan 25(2), 67–80 (1994)
5. Henschke, C.I., McCauley, D.I., Yankelevitz, D.F., Naidich, D.P., McGuinness, G., Miettinen, O.S., Libby, D.M., Pasmantier, M.W., Koizumi, J., Altorki, N.K., Smith, J.P.: Early lung cancer action project: overall design and findings from baseline screening. Lancet 354(9173), 99–105 (1999)
6. van Ginneken, B.: Computer-aided diagnosis in thoracic computed tomography. Imaging Decisions 12(3), 11–22 (2009)
7. Haralick, R.M., Sternberg, S.R., Xinhua, Z.: Imaging analysis using mathematical morphology. IEEE Trans. on Pattern Anal. and Machine Intell. 9(4), 532–550 (1987)
8. Yamamoto, S., Matsumoto, M., Tateno, Y., Iinuma, T., Matshmoto, T.: Quoit filter: A new filter based on mathematical morphology to extract the isolated shadow, and its application to automatic detection of lung cancer in x-ray ct. In: Proc. 13th Int. Conf. Pattern Recognition II, pp. 3–7 (1996)
9. Okumura, T., Miwa, T., Kako, J., Yamamoto, S., Matsumoto, M., Tateno, Y., Iinuma, T., Matshmoto, T.: Variable n-quoit filter applied for automatic detection of lung cancer by x-ray ct. In: Computer Assisted Radiology and Surgery(CAR 1998), pp. 242–247 (1998)
10. Kostis, W.J., Reeves, A.P., Yankelevitz, D.F., Henschke, C.I.: Three-dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical ct images. IEEE Transactions on Medical Imaging 22(10), 1259–1274 (2003)
11. Giger, M.L., Bae, K.T., MacMahon, H.: Computerized detection of pulmonary nodules in ct images. Investigative Radiology 29(4), 459–465 (1994)
12. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kiknis, R.: Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. Medical Image Analysis 2(2), 143–168 (1998)
13. Li, Q., Doi, K.: New selective nodule enhancement filter and its application for significant improvement of nodule detection on computed tomography. In: Proc. SPIE, vol. 5370 (2004)
14. McNitt-Gray, M.F., Wyckoff, N., Hart, E.M., Sayre, J.W., Goldin, J.G., Aberle, D.R.: Computer-aided techniques to characterize solitary pulmonary nodules imaged on ct. In: Computer-Aided Diagnosis in Medical Imaging, pp. 101–106. Elsevier, Amsterdam (1999)
15. Armato III, S.G., Giger, M.L., Moran, C.J., Doi, K., MacMahon, H.: Computerized detection of lung nodules on ct scans. RadioGraphics 19(5), 1303–1311 (1999)
16. Arimura, H., Katsuragawa, S., Suzuki, K., Li, F., Shiraishi, J., Sone, S., Doi, K.: Computerized scheme for automated detection of lung nodules in low-dose ct images for lung cancer screening. Academic Radiology 11(6), 617–629 (2004)

17. Matsumotoa, S., Kundelb, H.L., Geeb, J.C., Gefterb, W.B., Hatabu, H.: Pulmonary nodule detection in ct images with quantized convergence index filter. Medical Image Analysis 10(3), 343–352 (2006)
18. Kawata, Y., Niki, N., Ohmatsu, H., Kakinuma, R., Eguchi, K., Kaneko, M., Moriyama, N.: Quantitative surface characterization of pulmonary nodules based on thin-section ct images. IEEE Transaction Nuclear Science 45, 2132–2138 (1998)
19. Suzuki, K., Armato, S.G., Li, F., Sone, S., Doi, K.: Massive training artificial neural network (mtann) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography. Medical Physics 30(7), 1602–1617 (2003)
20. Suzuki, K., Horiba, I., Sugie, N.: Neural edge enhancer for supervised edge enhancement from noisy images. IEEE Transaction on Pattern Analysis and Machine Intelligence 25(12), 1582–1596 (2003)
21. Bator, M., Chmielewski, L.J.: Elimination of linear structures as an attempt to improve the specificity of cancerous mass detection in mammograms. Advances in Soft Computing 45, 596–603 (2007)
22. Lee, Y., Hara, T., Fujita, H., Itoh, S., Ishigaki, T.: Automated detection of pulmonary nodules in helical ct images based on an improved template-matching technique. IEEE Transactions on Medical Imaging 20(7), 595–604 (2001)
23. Ozekes, S., Saman, O., Ucan, O.N.: Nodule detection in a lung region that's segmented with using genetic cellular neural networks and 3d template matching with fuzzy rule based thresholding. Korean Journal of Radiology 9(1), 1–9 (2008)
24. Farag, A.A., El-Baz, A., Gimel'farb, G., Falk, R.: Detection and recognition of lung abnormalities using deformable templates. In: Proceedings of 17th International Conference on Pattern Recognition 2004, vol. 3, pp. 738–741 (2004)
25. Takizawa, H., Yamamoto, S., Shiina, T.: Recognition of pulmonary nodules in thoracic ct scans using 3-d deformable object models of different classes. Algorithms, Molecular Diversity Preservation International (MDPI) 3(2), 125–144 (2010)
26. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Pearson Prentice Hall, London (2008)
27. Takizawa, H., Nishizako, H.: Lung cancer detection from x-ray ct scans using discriminant filters and view-based support vector machine. The Journal of the Institute of Image Electronics Engineers of Japan 40(1), 59–66 (2011)
28. Zwiggelaar, R., Parr, T.C., Schumm, J.E., Hutt, I.W., Taylor, C.J., Astley, S.M., Boggis, C.: Model-based detection of spiculated lesions in mammograms. Medical Image Analysis 3(1), 39–62 (1999)

An Evaluation on Estimators for Stochastic and Heuristic Based Cost Functions Using the Epipolar-Constraint

Anton Feldmann[1], Lars Krüger[1], and Franz Kummert[2]

[1] Daimler AG,
Advanced Research,
Willhelm-Runge-Straße 11
D-89081 Ulm Germany
{anton.feldmann,lars.krueger}@daimler.com,
[2] University Bielefeld, Faculty of Technology,
Universitätsstraße 25,
D-33615 Bielefeld, Germany
franz@techfak.uni-bielefeld.de

Abstract. Camera calibration is a process of optimizing the camera parameters. This paper describes an evaluation of different stochastic and heuristic estimators for cost function minimization used in camera calibration.

The optimization algorithm is a standard gradient walk on the epipolar-constraint. The results show estimators work similar on the given set of correspondence. Correspondences selected to a given distribution over the frame gives better calibration results, especially the results on the yaw angle estimation show more robust results. In this paper the distribution will uniformly distributed over the frame using binning [1, 2].

The data used in this paper shows binning does lower the error behavior in most calibrations. The $\mathcal{L}^1$-norm and $\mathcal{L}^2$-norm using binning does not reach an error with respect to the ground truth higher 4 *pix*. The calibrations rejecting binning shows an impulse on the 970 calibration. To avoid this impulse binning is used.

Binning influences the calibration result more as the choice of the right m-estimator or the right norm.

1 Introduction

Advanced driver assistance systems (ADAS) use stereo camera systems for object detection, distance measuring or 3D reconstruction. A stereo system can de-calibrate over time due to thermal or physical interaction. To counteract the de-calibration of the camera, self-calibration is crucial.

Camera calibration is the process of determining geometric camera parameters. Usually camera calibration uses calibration targets for calibration [3–6]. Self-calibration uses no calibration targets for calibration and uses image data instead [1, 2, 5–10]. In general, the camera system optimization is based on the pinhole camera model (Fig. 1).

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 40–50, 2011.

Camera calibration can be separated in two parts. On the one hand the set of *intrinsic* camera parameter determining the projection of the camera coordinate system into the image coordinate and on the other hand the set of *extrinsic* camera parameter relates to the camera orientation and the camera position. In general, both parameter sets are obtained using non-linear equation optimization.

Non-linear optimization requires a good initial guess for the camera parameters. Estimation of camera parameters has been studied in the last three decades. Hence, various articles have been proposed for the direct estimation of the camera parameter. The most robust methods are the M-estimator, the Least Median of Squares (LMedS) method, and the Random Sample Consensus (RANSAC) method, but all suffer either from outliers or Gaussian noise in image data [11, 12]. Most described calibration methods use calibration targets [2–5, 8–10, 13].

This paper presents an evaluation on different estimation methods including the $\mathcal{L}^1$-norm, the $\mathcal{L}^2$-norm, the Blake-Zisserman, and the Corruption Gaussian estimator [9, P. 618] on image data for self-calibration and investigate the effect of equally distributed correspondences on the self-calibration. [3, 9] gives the statement that the choice of the right M-estimator or norm leads to a robust and accurate calibration. The evaluation in this paper shows that this statement can not be hold.

In Sec. 2 the state of the art is outlined. In Sec. 3 the tested estimators are described. Sec. 4 presents the results on the evaluation and this paper ends with a conclusion on the evaluation in Sec. 5.

2 Essential Matrix Estimation

The projective camera geometry is based on the pinhole camera model. In the pinhole camera model the projection rays intersects in a single point C, the camera center. The focal length defines the distance between the image plane and the camera center C. The principle point $m_0 = (u_0, v_0)$ is the intersection of the ray from C into the image plane (Fig. 1).

The epipolar-constraint describes the projection from one camera coordinate system into the other camera coordinate system for a stereo camera system [8–10].

The intrinsic camera parameters are archived in the camera calibration matrix [9, p. 163]

$$K = \begin{pmatrix} f & s & u_0 \\ 0 & \varphi \cdot f & v_0 \\ 0 & 0 & 1 \end{pmatrix} ,$$

with s the skew parameter indicating a skew pixel geometry and φ a scalar describing the aspect ratio. To estimate the items in the set of intrinsic camera parameter an initial camera calibration is used. In this paper we use the bundle adjustment method for the calibration of the set of intrinsic camera parameter [1, 4, 5, 13].

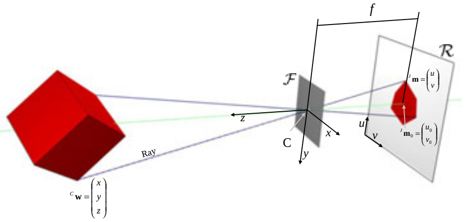

f
$\mathcal{R}$
$^{J}\mathbf{m} = \begin{pmatrix} u \\ v \end{pmatrix}$
$\mathcal{F}$
z
C
x
y
u
v
$^{J}\mathbf{m}_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$
Ray
$^{C}\mathbf{w} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

Fig. 1. The pinhole camera model [14]

The mapping of correspondences from the right into the left image and vice versa is described with

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

with F the *fundamental matrix* and x' the image pixel in the left image and x the corresponding pixel in the right image [9, p. 245]. x' and x are homogeneous and represented in the pixel-coordinate-system. The fundamental matrix is the natural extension of the *essential matrix* [15]

$$E = K'^T F K. \tag{2}$$

[16, 17] shows stability analysis on Eq. (1) and leads to the separation of intrinsic and extrinsic camera parameter Eq. (2). This separation of the fundamental matrix leads with Eq. (1) to Eq. (3)

$$x'^T F x = x'^T (K'^{-1})^T E K^{-1} x = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T E \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}^T t_\times R \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0, \tag{3}$$

with an essential matrix $E = t_x R$, R the rotation matrix and t_x the skew symmetric matrix with the translation vector for components. The methods in this paper uses the algorithm described in [18] for correspondence detection.

In general, correspondences are usually not equally distributed on the image. Binning is used to achieve near equally distributed correspondences [1, 2]

After correspondence mapping from one image into the other, the distance of the x coordinate between the correspondences $d(x', x) = \|x' - x\|_2$ is called *disparity.* Camera self-calibration using the epipolar-constraint with a stereo rig as described in [2, 7–10].

3 Epipolar Constraint Optimization Using Stochastic or Heuristic Estimators

In this section we present the evaluated methods: the $\mathcal{L}^1$-norm estimator, the $\mathcal{L}^2$-norm estimator, the Blake-Zisserman estimator, and the Corrupted Gaussian estimator.

The optimization problem described with Eq. (3) searches for the shortest length between the error of Eq. (3) and its kernel

$$ker(x'Ex) \equiv 0 \Longleftrightarrow 0 = x'Ex.$$

To find the minimum distance between the error and the kernel of Eq. (3), the presented algorithm use Shor's algorithm with a standard newton gradient method [19]. Shor's algorithm assures a fast optimization process. For outlier detection we use the estimator with the cost function $C(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$.

A small residuum $\xi_i = x_i' E x_i$ indicates the inlier and a large residuum indicates the outlier. The inliers are used for calibration [9].

The equation

$$\mathcal{L}_p(\xi) = (\int_{\mathbb{R}^n} \|\xi\|^p dt)^{\frac{1}{p}} =: \|\xi\|_p \tag{4}$$

is called the *Lebesque*-Norm or p-Norm. The *least mean square* optimization methods uses usually $C(\xi) = \sum_{i=1}^n \|\xi_i\|_2$ for regression [20]. However, the cost function $C(\xi) = \sum_{i=1}^n \|\xi_i\|_1$ is not as outlier sensitive as $C(\xi) = \sum_{i=1}^n \|\xi_i\|_2$ [21].

The Corrupted Gaussian estimator is defined by

$$C(\xi) = \sum_{i=1}^{n} -\ln(\alpha e^{-\xi_i^2} + (1-\alpha)e^{\frac{(\frac{-\xi_i}{\omega})^2}{\omega}}) \tag{5}$$

with $\alpha \in [0, 1]$ the expected factor of the inlier and ω the ratio of the standard deviation. Corrupted Gaussian estimation expects that the outliers have a Gauss distribution with a large standard deviation.

Using Eq. (5) with $\alpha \approx 1$ leads to the Blake-Zisserman estimators

$$C(\xi) = \sum_{i=1}^{n} -\ln(e^{-\xi_i^2} + \epsilon), \tag{6}$$

with $\epsilon << 1$ a small expected error in the image data.

4 Experimental Investigations

A stereo camera system takes a sequence for testing. The experiments includes 1435 stereo image pairs in about 30 sec while driving. The sequence is taken with the camera system shown in Fig. 2.

Fig. 3 - Fig. 10 presents the experimental results. The correspondences are rectified and normalized using the algorithm described in [18].

Fig. 3 shows a calibration using the $\mathcal{L}^1$-norm. Each row represents the error with respect to the ground truth of the three angles. In the second row in Fig. 3 the yaw angle is presented with an error value between $[0, 7]$ *pix* while the other angles are limited to 1 *pix*. The biggest error rise in calibration 970. This impulse is also seen in the third row at the same frame.

Fig. 4 shows the calibration using the $\mathcal{L}^2$-norm. The behavior of figure Fig. 3 and Fig. 4 are similar on the given dataset. The impulse at calibration 970 does appear in the calibration using the $\mathcal{L}^2$-norm and has the same value the $\mathcal{L}^1$-norm shows.

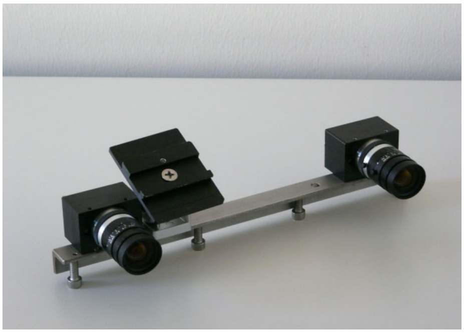


Fig. 2. The stereo-camera-system

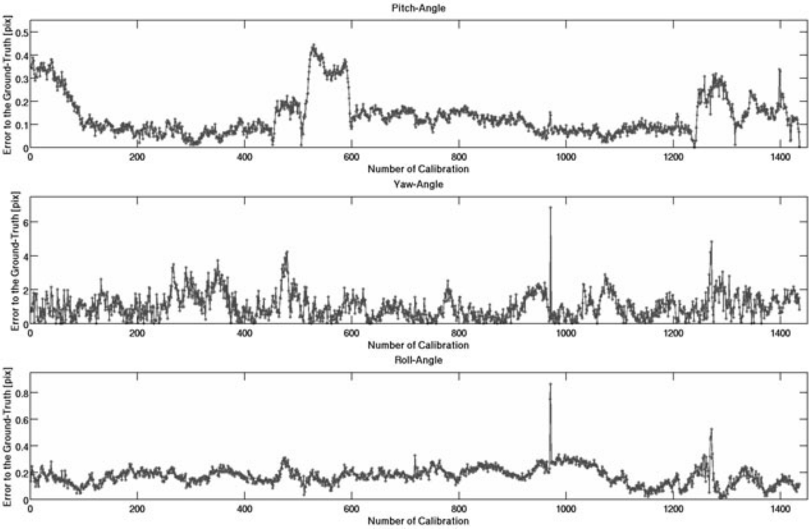


Fig. 3. Stereo-calibration using the $\mathcal{L}^1$-norm the estimator for optimization. The first row shows the pitch angle error with respect to the ground truth, and the second row the yaw angle error with respect to the ground truth. The roll angle is given in the last row.

The best behavior is illustrated in Fig. 5. The error value for the calibration of the yaw angle is less then 6 *pix*. The Corrupted Gaussian estimator uses a value to give a smooth change from the set of data and outlier. The Blake-Zisserman estimator has a hard limit for a data outlier selector. The results using the Blake-Zisserman estimator is presented in Fig 6.

The results on the stereo calibration using the Blake-Zisserman estimators gives at calibration number 970 an unexpected impulse as much higher the other calibrations does. The pitch angle has a error of 2416 *pix*, the yaw angle has a error of 70 *pix* and the roll angle gives a error of 9 *pix*. For calibration number

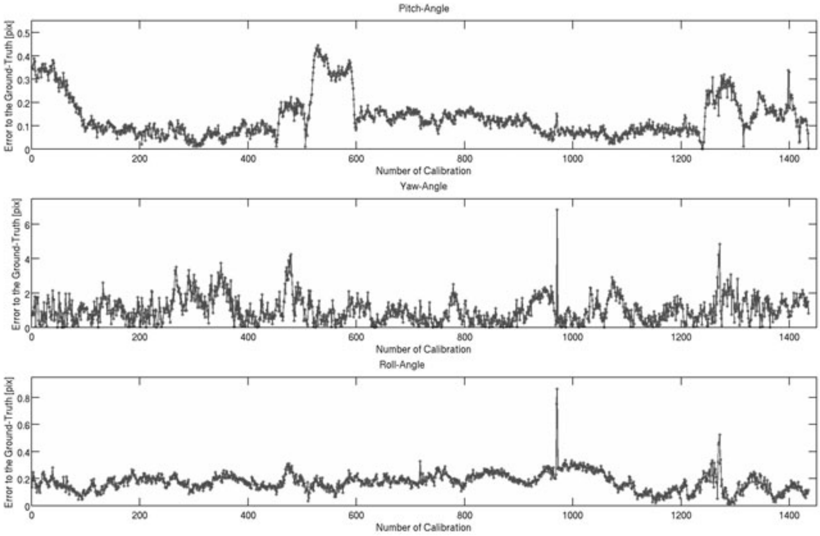


Fig. 4. Stereo-calibration using the $\mathcal{L}^2$-norm the estimator for optimization. The first row shows the pitch angle error with respect to the ground truth, and the second row the yaw angle error with respect to the ground truth. The roll angle is given in the last row.

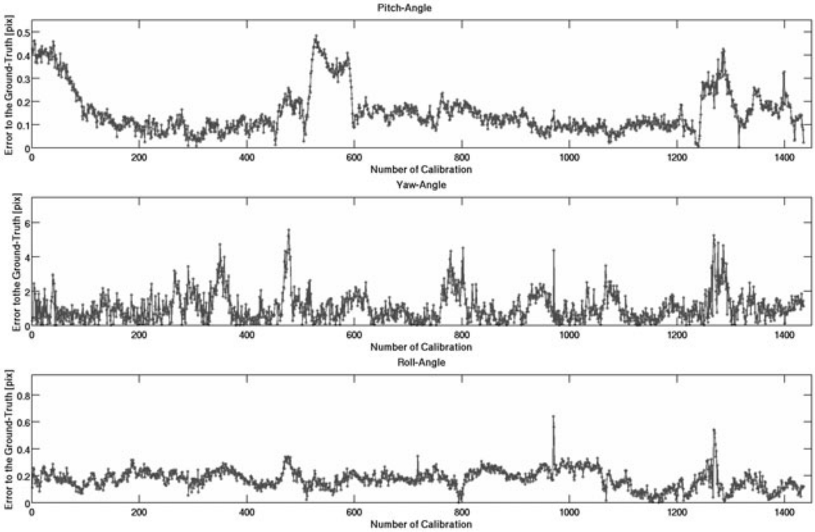


Fig. 5. Stereo-calibration using the Corrupted Gaussian estimator for optimization. The first row shows the pitch angle error with respect to the ground truth, and the second row the yaw angle error with respect to the ground truth. The roll angle is given in the last row.

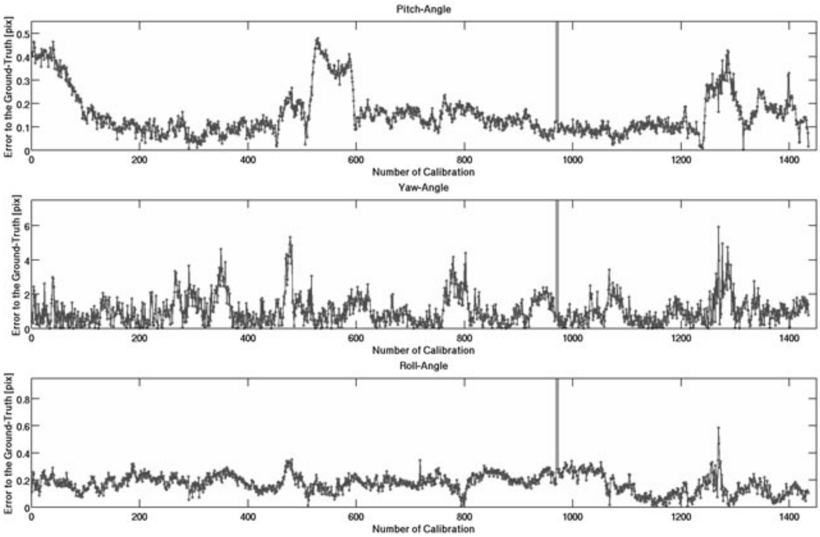


Fig. 6. Stereo-calibration using the Blake-Zisserman estimator for optimization. The first row shows the pitch-angle error with respect to the ground truth, and the second row the yaw angle error with respect to the ground truth. The roll angle is given in the last row.

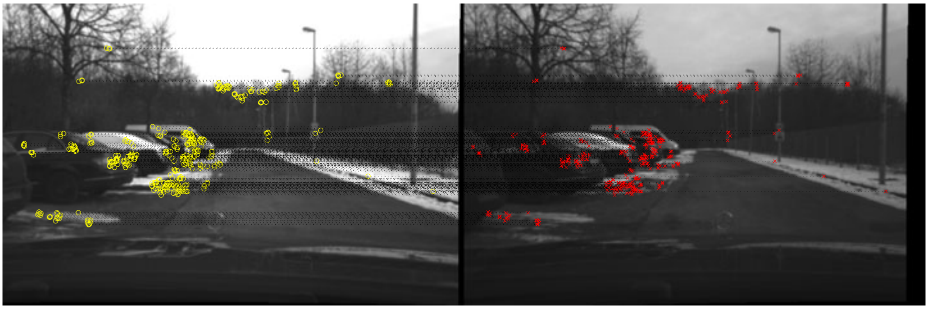


Fig. 7. The calibration with number 970 has just correspondences in the left lower corner. This leads to the hugh errors in the angle for calibration number 970.

970 see Fig. 7. In Fig. 7 the correspondences are centered in more or less a corner of the frame. This impulse in a stereo camera system for an ADAS can have fatal effect on the car behavior.

Binning is used to counteract this effect [1, 2]. Binning splits the image into four partitions, and binning force to correspondence set to select correspondences from the images until the bins are filled to the bin limit. Results from the evaluation using binning are presented in Fig. 8 - Fig. 10

Fig. 8 shows (in contrast to Fig. 3) an error value in the range of $[0, 4]$ pix. The unexpected impulse behavior is avoided.

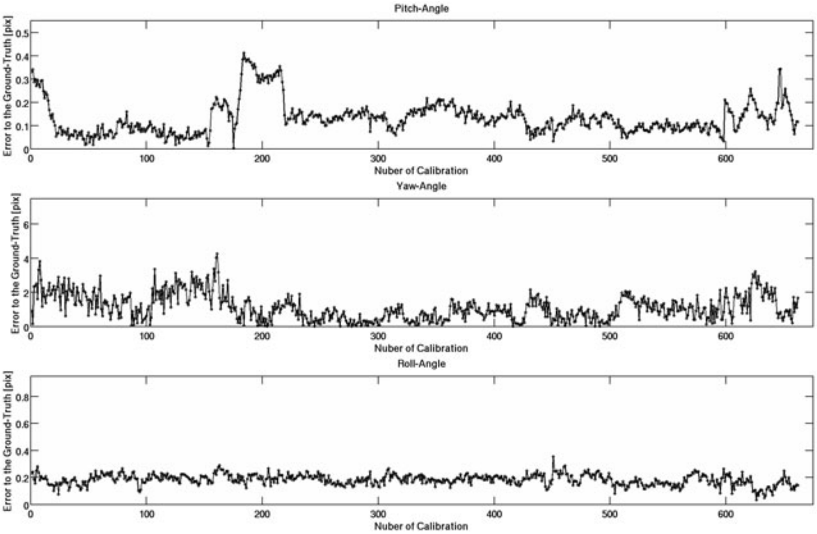


Fig. 8. Stereo-calibration using the $\mathcal{L}^1$-norm estimator with the binning-approach. The first row shows the pitch-angle error with respect to the ground truth, and the second row the yaw-angle error with respect to the ground truth. The roll-angle is illustrated in the last row.

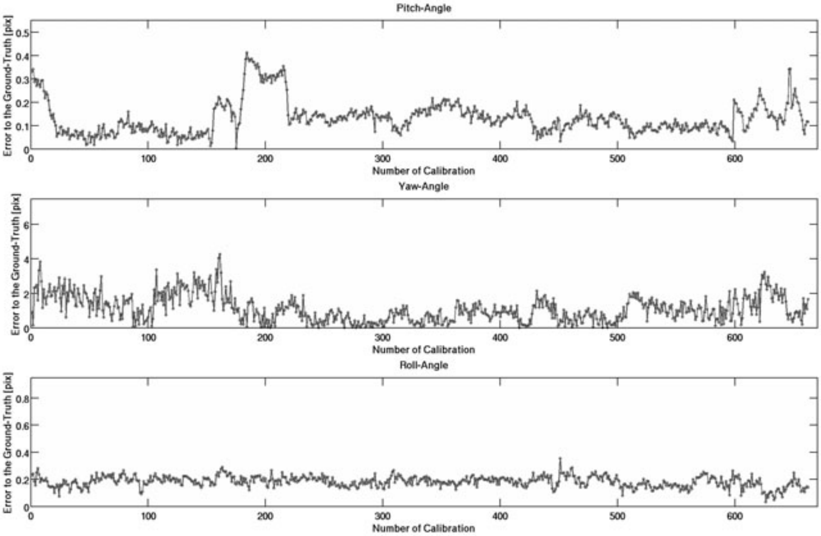


Fig. 9. stereo-calibration using the $\mathcal{L}^2$-norm estimator with the binning-approach. The first row shows the pitch-angle error with respect to the ground truth, and the second row the yaw-angle error with respect to the ground truth. The roll-angle is illustrated in the last row.

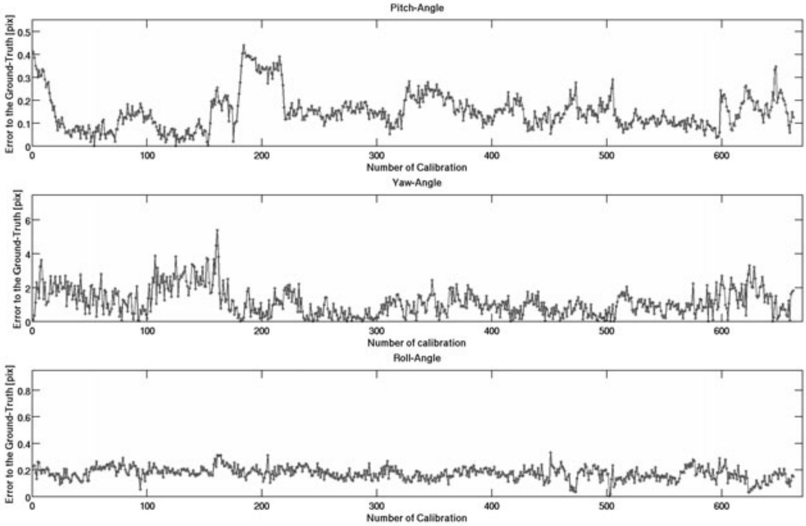


Fig. 10. Stereo-calibration using the Corrupted Gaussian estimator with the binning-approach. The first row shows the pitch-angle error with respect to the ground truth, and the second row the yaw-angle error with respect to the ground truth. The roll-angle is illustrated in the last row.

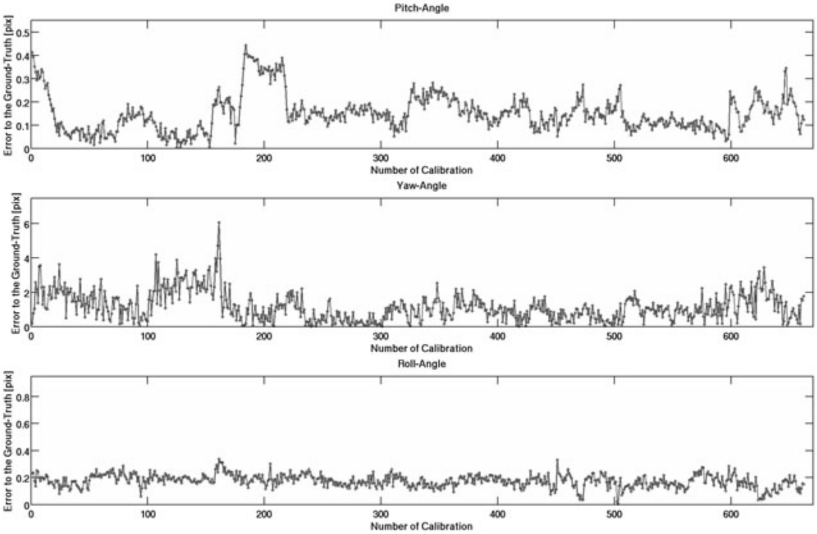


Fig. 11. Stereo-calibration using the Blake-Zisserman estimator with the binning-approach. The first row shows the pitch-angle error with respect to the ground truth, and the second row the yaw-angle error with respect to the ground truth. The roll-angle is illustrated in the last row.

Fig. 9 shows a similar error behavior on the correspondences as Fig. 8. This was expected, because this behavior was observed in the calibration without binning.

Fig. 10 illustrates that using binning avoids the impulse seen in Fig. 5 at calibration 970 and leads to a more accurate calibration.

The Corrupted Gaussian estimator is slightly more accurate as the Blake-Zisserman estimator (see. Fig. 5 in contrast to Fig. 6 and Fig. 10 in contrast to Fig. 11). A calibration using the $\mathcal{L}^1$-norm or $\mathcal{L}^2$-norm estimator is a robust method using binning or rejecting binning.

Selecting correspondences use binning until a bin limit is reached gives better results after the calibration step. The yaw- angle is the angle with the largest error and the hardest angle to estimate, albeit if binning is used the error of the yaw-angle shrinks while the number of calibration shrinks. Selecting correspondences to reach the bin limit often use a certain number of frames. In general, a calibration is done after a series of frames using binning. Therefor using no binning before calibration can give a calibration after each frame.

5 Conclusion

If the evaluated estimators use binning then the error with respect to the ground truth is at leased smaller over all calibrations. This paper gives a hint of a relation between the yaw angle and the rotation angle. The number of correspondences has an influence on the calibration is shown using binning. The influence on the number of correspondences to the calibration quality is shown in [1].

A robust estimators use the $\mathcal{L}^1$-norm or the $\mathcal{L}^2$- norm. The assumption that the error is Gaussian distributed is not correct in the scenario illustrated in this paper. The Corrupted Gaussian and the Black-Zisserman estimator does use this constraint on the error in the data. The large error values shows the error in the data is not Gaussian distributed.

This paper shows that the use of binning is more important then the choice of the right M-estimator or the right norm.

Further work has to be done in the robust essential matrix and fundamental matrix estimation using probabilistic heuristics for long range approaches. The selected correspondences in the image using the algorithm in [18] gives a random selected set. A dynamic binning algorithm based on the equally distribution is necessary to force an equally distributed set of correspondences in an image. The correspondence distribution has to be checked against existing distributions in stochastic and regression analysis.

References

1. Feldmann, A., Krüger, L., Kummert, F.: Quality measure in epipolar geometry for vehicle mounted stereo camera systems. In: Luhmann, T., Müller, C. (eds.) Photogrammetrie Laserscanning Optische 3D-Messtechnik (Beiträge der Oldenburger 3D-Tage 2011). Wichmann Verlag, Heidelberg (to appear, 2011)
2. Dang, T., Hoffmann, C.: Stereo calibration in vehicles. In: IEEE Intelligent Vehicles Symposium, vol. 4, pp. 268–273 (2004)

3. Luhmann, T., Robson, S., Reeves, C., Wainwright, P., Kyle, S.: Close range Photogrametry:Principles, Methods and Applications, vol. 1. Whittles Publishing (2006)
4. Pollefeys, M.: Self-calibration and metric 3D reconstruction from uncalibrated image sequences. PhD thesis, K.U.Leuven (1999)
5. Krüger, L., Emmert, V., Feldmann, A., Lindner, F.: Evaluating the accuracy of camera calibration for driver assistance systems. In: Luhmann, T., Müller, C. (eds.) Photogrammetrie Laserscanning Optische 3D-Messtechnik Paper Describes an Evaluation of Different Stochastic and Heuristic est (Beiträge der Oldenburger 3D-Tage 2011). Wichmann Verlag, Heidelberg (to appear, 2011)
6. Krüger, L.: Model Based Object Classification and Localisation in Multiocular Images. PhD thesis, University of Bielefeld (2007)
7. Dang, T.: Kontinuierliche Selbstkalibrierung von Stereokameras. PhD thesis, Institut für Mess- und Regelungstechnik mit Maschinenlaboratorium, MRT (2007)
8. Wöhler, C.: 3D Computer Vision. Efficient Methods and Applications. Springer, Heidelberg (2009)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
10. Faugeras, O., Luong, Q.T., Papadopoulou, T.: The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications. MIT Press, Cambridge (2001)
11. Xu, G., Zhang, Z.: Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach. Kluwer Academic Publishers, Norwell (1996)
12. Torr, P.H.S., Murray, D.W.: Outlier detection and motion segmentation. In: SPIE, vol. 2059, pp. 432–443 (1995)
13. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment – A modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
14. Barrois, B.: Analyse der Position, Orientierung und Bewegung von rigiden und artikulierten Objekten aus Stereobildsequenzen. PhD thesis, University Bielefeld (2010)
15. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135 (1981)
16. Luong, Q.T.: Matrice fondamentale et auto-calibration en vision par ordinateur. PhD thesis, Universite de Paris-Sud, Orsay (1992)
17. Luong, Q.T., Faugeras, O.D.: The fundamental matrix: Theory, algorithms, and stability analysis. International Journal of Computer Vision 17, 43–75 (1996)
18. Stein, F.J.: Efficient computation of optical flow using the census transform. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 79–86. Springer, Heidelberg (2004)
19. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26, 1484–1509 (1997)
20. Draper, N.R., Smith, H.: Applied Regression Analysis (Wiley Series in Probability and Statistics), 3rd edn. Wiley Interscience, Hoboken (1998)
21. Klenke, A.: Wahrscheinlichkeitstheorie. Springer, Heidelberg (2006)

Facial Movement Based Recognition

Alexander Davies[1], Carl Henrik Ek[2], Colin Dalton[3], and Neill Campbell[4]

[1] University of Bristol
davies@cs.bris.ac.uk
[2] Royal Institute of Technology
chek@csc.kth.se
[3] University of Bristol
colin.dalton@bristol.ac.uk
[4] University of Bristol
neill.campbell@bristol.ac.uk

Abstract. The modelling and understanding of the facial dynamics of individuals is crucial to achieving higher levels of realistic facial animation. We address the recognition of individuals through modelling the facial motions of several subjects. Modelling facial motion comes with numerous challenges including accurate and robust tracking of facial movement, high dimensional data processing and non-linear spatial-temporal structural motion. We present a novel framework which addresses these problems through the use of video-specific Active Appearance Models (AAM) and Gaussian Process Latent Variable Models (GP-LVM). Our experiments and results qualitatively and quantitatively demonstrate the framework's ability to successfully differentiate individuals by temporally modelling appearance invariant facial motion. Thus supporting the proposition that a facial activity model may assist in the areas of motion retargeting, motion synthesis and experimental psychology.

1 Introduction

Realising increasingly believable human facial animation is a key concept for members of the computer science community and has been a desirable goal since the start of facial animation. Human Computer Interaction (HCI)[1], medical imagery and forensics [2] are just a few areas which greatly benefit from the application of realistic facial animation. Furthermore, achieving convincing human facial animation has been viewed as a major milestone for the special effects industry; a fundamental step in connecting the audience with computer generated content, whether it is virtual supporting characters in a live-action film or a complete virtual cast in a videogame.

Realistic facial animation is a formidable challenge to both the academic fields of computer vision and graphics as well as the entertainment and video game industries. A major contributing factor to the difficulty of creating convincing facial animation is our innate perceptions of the visual mechanics of the human face. When presented with more realistic virtual characters it is this intrinsic model of human facial motion that becomes a hindrance in our acceptance of

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 51–62, 2011.

the character. This phenomenon, aptly named the "uncanny valley" [3] has been set as the main hurdle whom all involved in research in realistic facial animation attempt to overcome.

In this paper we investigate whether it is possible for a system to distinguish an individual from others based on facial movement alone. To this end we present a framework for capturing and modelling the facial dynamics of multiple individuals. Our framework is capable of classifying individuals based on facial motion without any appearance based cues and has the facilities to generate synthetic facial motion. Well established techniques, such as Active Appearance Models (AAM) for markerless feature tracking of video data and the Gaussian Process Latent Variable Model (GP-LVM) for non-linear dimensionality reduction are incorporated into our framework. AAMs and GP-LVMs are techniques that have been explored extensively in their respective fields of computer vision and dimensionality reduction and have shown great promise in the modelling of human motion data. Our novel contribution to this field lies is the modelling of the facial dynamics rather than static images using the GP-LVM. In addition, we are confident that modelling the facial dynamics of individuals will result in practical applications in the areas of motion retargeting and motion synthesis as well as have a positive impact in the field of experimental psychology.

The remainder of this paper is organised as follows: Section 2 consists of a brief overview of facial animation and a summary of related work. We describe our framework in more detail in Sect. 3, discuss our results in Sect. 4 and present our conclusions in Sect. 5.

2 Background and Related Work

Over the last ten years the discriminative power of facial dynamics has become more recognised in the computer vision community with increased research in applying facial dynamics to facial recognition. It has been demonstrated that using dynamic features to compliment appearance features can improve facial recognition two-fold [4]. A disadvantage of analysing still images is that they are normally taken during the apex of an expression which is atypical of realistic facial motion; a complex combination of expressions at varying intensities. The goals regarding facial recognition and analysis can roughly be grouped into one of two categories. The first group focusing on recognising known expressions at different intensities across different individuals whilst the second group are concerned with discriminating individuals based on facial movement regardless of which expressions they are performing. Our work falls into the latter category.

There have been numerous approaches to facial movement analysis each employing different dynamic features and techniques. Texture derived features such as the similarities between Haar-like features [5], Extended Volume Local Binary Pattern (EVLBP) features [6], dense optical flow [4] and a histogram of fused Gabor wavelet representations [7] have been used to reduce a video sequence to a lower dimensional temporal vector. Conversely, 3D parametric models [8, 9] have been fitted to video sequences to produce similar temporal vectors.

Once dynamic features of facial movements are extracted, machine learning techniques are employed to infer structure from the features for classification, regression or generative purposes. Techniques such as Support Vector Machines (SVM) [5, 8], Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Hidden Markov Models (HMM)[6] can be used to classify one individual or expression from another. However caution should be exercised when combining dynamic features with appearance features as a wrong set of dynamic features may hinder the task of recognition and classification instead of aiding it [6]. Consequently, selecting a relevant set of dynamic features can be cast as a feature selection problem [9]. One technique for selecting the features which have the highest discriminative power is Boosting as shown in [6, 5, 7]. Our contribution fits into this ideological model by using AAMs for feature extraction, a GP-LVM coupled with K-Means clustering to reveal the higher level structure of the facial data followed by an HMM to classify one individual from another.

Our case for using AAMs for feature extraction has been built on the ability of AAMs to robustly track non-rigid deformation of facial movements as well as large out of plane head rotations. Since being introduced to the Computer Science community [10], AAMs have been extended to boost robustness and flexibility through several proposed fitting methods [11]. More efficient algorithms have also been proposed such as the Inverse Compositional (IC) method [12] which achieved significant performance gains over typical gradient descent algorithms. Boosting and Support Vector Regression (SVR) have also been utilised to create non-linear fitting algorithms [11]. AAMs have also been extended to use 3D motion estimation to constrain the examples generated to ones only plausible in 3D, drastically reducing fitting times [13].

The GP-LVM [14] is a non-linear probabilistic dimensionality reduction technique using Gaussian Processes (GP). Our motivation for using the GP-LVM is that the GP-LVM and its various flavours have been used for unsupervised learning of human motion data [15–17], outperforming other dimensionality reduction techniques [18]. Furthermore, GP-LVMs have been used in combination with SVMs for facial expression classification [19]. More importantly, the GP-LVM is a probabilistic model capable of generating the observational probabilities which are instrumental to the modelling of facial dynamics in our framework. The work in [20] shares similar attributes to our own, using AAMs to parameterise facial motion from video. However whilst the focus of [20] is on creating an audio-visual mapping using an SGP-LVM, ours is on discovering a mapping between individuals and their differences in facial motion.

To summarise, our novel contribution is that instead of attempting to recognise and classify facial expressions in individuals we have attempted to distinguish individuals performing similar and dissimilar facial actions. We are attempting to model the style or facial dynamics which are *individual specific* by discretisation of the latent space and incorporating dynamic features for purposes of classification and synthesis of realistic facial motion. Additionally, whereas previous works have been tested using established expression databases where the expressions may have been posed [21] or triggered by stimuli [22]; we have attempted to apply our framework to video data of subjects undergoing natural facial actions without constraints to global head movement.

3 AAM and GP-LVM Framework

Our framework models the facial dynamics of individuals through an HMM. In order to build the HMM our framework requires the probabilities of the observations given a particular latent state and the transitional probabilities from one state to another. The observation probabilities are generated using the GP-LVM whilst the observations themselves are features extracted using a face tracker built on AAMs. Due to the inclusion of the GP-LVM, our framework has the advantage over other classification frameworks by having the added functionality of filtering the data and synthesising new data.

3.1 Obtaining Tracking Data Using AAMs

For the successful tracking of facial motion we would need an AAM trained on a wide range of facial expressions. Initially, our training set of images was based on the Facial Action Coding System (FACS) Action Units (AU) [23]. Our choice stemmed from the fact that FACS has been the most authoritative attempt at comprehensively measuring the facial movements of the human face. With 46 muscle-based AUs, it is possible to create a wide range of facial expressions using the combinations of just two or three AUs. We created three AAMs for a single individual. One training set consisted of images of single AUs, the second set consisted of common combinations of AUs and the final set contained both the single and combined AUs. In practice we found that these training sets performed poorly. The AU images, although atomic in terms of expression, may be non-orthogonal in the AAM space attributing to the poor performance of the AAM.

Consequently, to build an AAM tracker which tracked our subjects as accurately as possible, we opted to create video clip specific AAMs. Our empirical experience has shown that the fitting methods we used [12, 10] were unable to accurately track features in video with models trained from multiple subjects or the same subjects in different video clips. Ultimately our system applies the Ramer-Douglas-Peuker algorithm [24] to the input video data. These key-frame images returned by the algorithm would then be labelled semi-automatically for use as the training set. Specifically, a subset of the training images would be labelled manually to create a small initialising AAM. This AAM would then be used to fit the label points onto the remaining key-frame images, accelerating the labelling process. Since we were using AAMs trained on individual video sequences we decided to just use the shape data from the AAM, shown in Fig. 1, as input to the GP-LVM.

3.2 Normalising Shape Data for GP-LVM Use

Before the tracking data is used to create a GP-LVM the data needs to be normalised for translation, scale and shape to remove variance due to appearance.

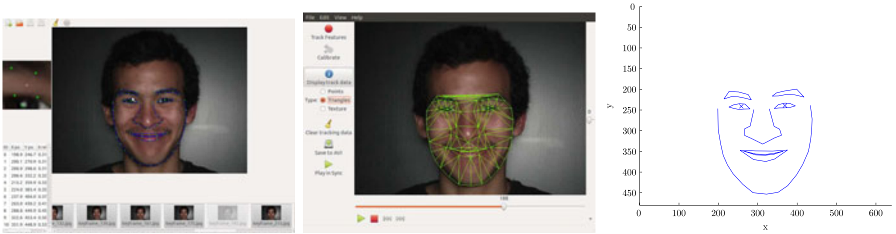


Fig. 1. The tracking software: The utility used to mark the key-frames (left), the AAM tracker on a single frame (middle) and the bottom panel shows the same data output in Matlab (right)

This step was vital in ensuring that the GP-LVM learned the facial motions of the individuals and not their appearances. For scale and translation normalisation, we apply the following: for each frame i in the video sequence with points $j = 1 \dots P$ the centre of the marker points $(p_i^{\mathrm{c_x}}, p_i^{\mathrm{c_y}})$ and the scaling factor s_i are calculated as shown in (1) to (3).

$$p_i^{\mathrm{c_x}} = \frac{1}{P}\sum_{j=1}^{P} p_{ij}^{\mathrm{x}}, \quad p_i^{\mathrm{c_y}} = \frac{1}{P}\sum_{j=1}^{P} p_{ij}^{\mathrm{y}} \quad . \tag{1}$$

$$s_i = \max(\|\mathbf{p}_{ij} - \mathbf{p}_i^{\mathrm{c}}\|) \ \text{ where,} \tag{2}$$

$$\mathbf{p}_{ij} = (p_{ij}^{\mathrm{x}}, p_{ij}^{\mathrm{y}})^{\mathrm{T}} \quad , \mathbf{p}_i^{\mathrm{c}} = (p_i^{\mathrm{c_x}}, p_i^{\mathrm{c_y}})^{\mathrm{T}} \quad . \tag{3}$$

Equation (4) shows how the normalised points $(p_{ij}^{\mathrm{n_x}}, p_{ij}^{\mathrm{n_y}})$ are calculated once the centre and scale values are known.

$$p_{ij}^{\mathrm{n_x}} = \frac{1}{s_i}(p_{ij}^{\mathrm{x}} - p_i^{\mathrm{c_x}}) \quad , p_{ij}^{\mathrm{n_y}} = \frac{1}{s_i}(p_{ij}^{\mathrm{y}} - p_i^{\mathrm{c_y}}) \quad . \tag{4}$$

After the scale and translation information has been removed from the data the next step is to remove shape information of the individual. In our tracker we appended the the *mean* shape generated by the AAM to the tracking data which would then be used to subtract from the normalised data to get the displacement information.

3.3 Decoupling Global Head Motion

One of our video data sets containing Natural Expressions (NER) of students performing authentic unconstrained facial motions contains a large degree of global head motion. Since the main focus of our research is determining the discriminative power of using facial motions as features, it was important to segregate rigid global head motion from non-rigid facial deformations and preventing one set of motions from influencing the other. More importantly, in practice, we have observed that global head motion accounts for the majority of the variation in the model, factoring out facial expression motions during the creation of the GP-LVM if left unfiltered from the data.

Compared to extracting global head movement from 3D tracking data which involves calculating a linear transformation, decoupling global head movement from 2D tracking data is a non-trivial process. One approach would be to treat the face or the set of points as a 3D plane, then using homographies, recover the head pose to subsequently determine the global head movement [25]. These methods, though successful in recovering pose information, using homogrphies to subtract the global movement from the 2D tracking data are not sufficient for large out of plane rotations as they rely on the assumption that the face is a plane.

Our selected approach was to use the structure from motion algorithm developed in [26]. Once the 3D rigid structure was recovered, the data was then orthographically projected onto 2D space. Subsequently, the translations between the facial points in each frame of the rigid motion sequence and a manually selected frontal pose frame was calculated. These translations could then be applied to the original 2D tracking data to remove the global head movement.

3.4 Classification of Individual Facial Dynamics Using GP-LVM

Creating a GP-LVM from the normalised data reduced the dimensionality of the data from 160 dimensions to two. Once the latent space is generated we then segment the space into clusters using K-Means++ clustering [27]. Once the clusters are set, the means of these clusters are taken to represent the states in the HMM. The probabilities of these states can be interpreted as *the probability that the observed data generated came from the cluster k*. Or more formally, $p(y_i|C_k)$ where C_k is the mean position of the cluster k.

More interestingly however with regards to the HMM was the process of modelling the transitional probabilities from one state to another. There are a number of ways to go about generating these probabilities. One method would be to assume transitional smoothness between states and model a Gaussian over the transitional probabilities. Another method may be to actually model these probabilities using a multi-modal Gaussian. However the initial method that we selected was to use a non-parametric histogram approach where the transition probabilities would be calculated by recording the actual states which the tracking data of individuals would pass through. These values would then be normalised to get the transitional probabilities.

Once both the probabilities for the states and the transitions were found, we could then calculate the Viterbi path through the HMM state trellis, i.e. the most likely path given the states and transition probabilities.

In practice when classifying individuals we get a set of candidate paths per individual i.e. one path per combination of transitional model and individual. To calculate the probability of each candidate path we have to take the multiplicative sum (or sum of the logs) of the state probabilities and transition probabilities for each observation in the path. The path with the highest probability is then assigned to that individual.

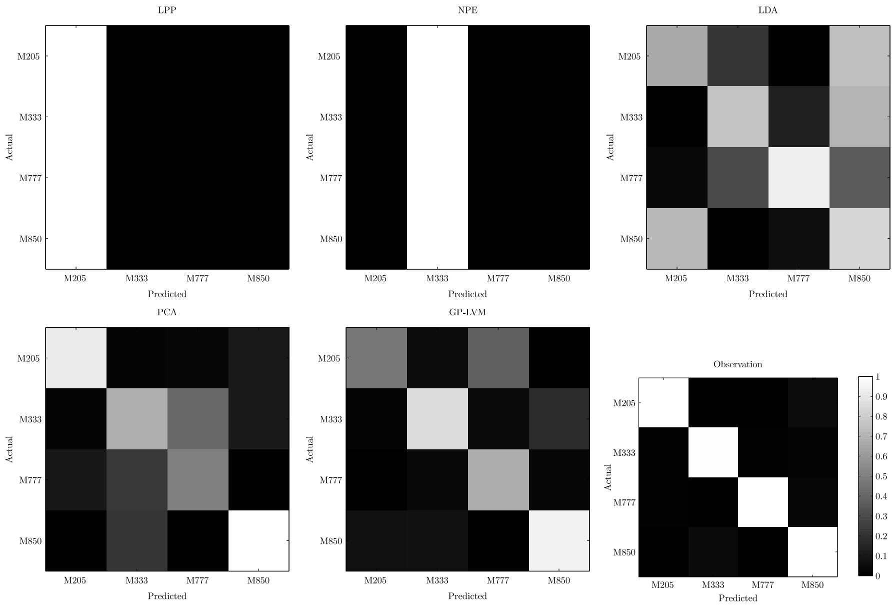


Fig. 2. Confusion matrices for classification on test points in lower dimensional space. From top-left to bottom right. LPP, NPE, LDA, PCA, GP-LVM and Observational space.

Table 1. Classification percentages for various reduced dimensional spaces

	M205	M333	M777	M850
Observation Space	99.1803	93.8776	98.9691	100.0000
Embedded Space				
LPP	100	0	0	0
NPE	0	100	0	0
LDA	37.7049	49.6599	61.8557	51.7007
PCA	86.8852	53.7415	62.8866	80.9524
GP-LVM	55.7377	89.7959	72.1649	91.1565

4 Experiments and Results

For our Experiments we used two video sets. A Simple Smile Set (SSS) which consisted of four individuals (M205, M333, M777 and M850) performing a simple artificial smile. Each subject was asked to perform the smile multiple times as consistently as possible and as close to an example video as possible. A second video set consisting of Natural Expressions (NER) was acquired from a psychology experiment. These video clips consist of greater numbers of individuals watching a set of prepared video clips and periodically being asked a set of questions as well as being asked to perform a set of tasks. These stimuli were designed to provoke natural actions and emotional responses with unconstrained head movement. Although filmed using different cameras in different locations all subjects were filmed at 1080p resolution with 25FPS in an environment with controlled background and lighting conditions. We split the SSS data set into a training set and a testing set by grouping the first and last smiles of each individual as the training set and using the remaining three middle smiles as the test set. The number of frames in the SSS was 1224 with 711 frames used for the training set and the remaining 513 frames used for the test set.

4.1 Baseline Test

The purpose of the baseline test was to compare the classification strength of existing linear dimensionality reduction methods against the GP-LVM. In this experiment we only wanted to see how well classification performed without the use of temporal information, treating each frame as a static image. Using the SSS training set, we generated two dimensional latent spaces using Locality Preserving Projection (LPP), Neighbourhood Preserving Embedding (NPE), LDA, PCA and GP-LVM. We used Nearest Neighbours (NN) to classify the test set. We also used observation space as a control. Figure 2 and Table 1 show that the GP-LVM outperforms the linear dimensionality reduction techniques. It is worth noting that the high classification accuracy rate achieved using just the observation space shows that the data is very dense and that it is possible to find very similar examples in the training set to match the test set using NN. However, classifying data in the observation space is of little use to us since additional tasks such as synthesis of data, generalising to more varied test cases and filtering of data would not be possible.

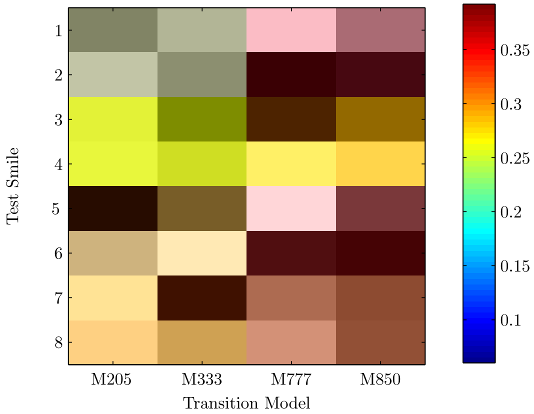


Fig. 3. Normalised negative log probabilities of paths generated using different transition models. The lower values indicate a higher probability.

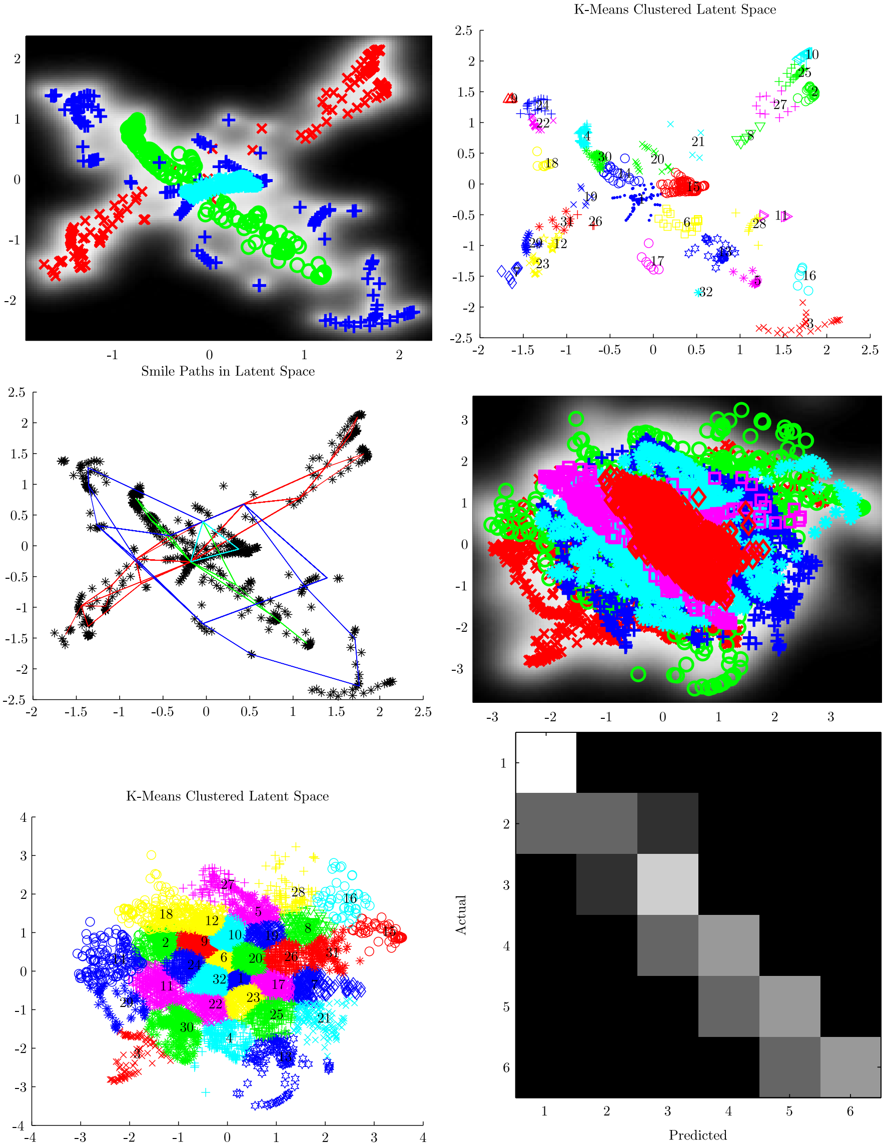


Fig. 4. From top-left to bottom-right: Results from the smile experiment(1-3) and the emotional talking experiment(4-6). The latent space of four individuals (1), the same latent space clustered using K-Means clustering (2) and the paths through the cluster centres of the training data in latent space (3). The latent space generated from the six individuals (4), the clustered latent space (5) and the confusion matrix of the classification process (6).

4.2 Smile Experiment

The main aim of this particular experiment was to ascertain whether it was possible to use the GP-LVM and HMM framework described in Sect. 3 to discriminate different individuals performing the same action. The plots in Fig. 4 reveal that the paths generated from the discretised latent space using the Viterbi algorithm are able to maintain the temporal structure of their respective individuals within the training data. Furthermore, Fig. 3 confirms that it is possible to use the log probabilities of paths generated from the test data to classify individuals successfully, showing a one hundred percent accuracy rate over our test data with $k = 32$ clusters. Figure 3 demonstrates that a path generated from a test smile and transitional model which came from the same individual will have the highest log probability.

4.3 Emotional Talking Experiment

The Emotional Talking Experiment is an extension of the Smile Experiment. Instead of trying to differentiate individuals who are performing a simple identical action, this experiment tested whether it was possible for the framework to separate individuals performing a range of natural actions from one another. Clips were taken from the NER set of six individuals talking about a subject which made them angry. The data was split into a training set and a test set of equal length, 8250 frames each. Classification was done in the same manner as the Smile Experiment. The results in Fig. 4 demonstrate that although there is a performance penalty with more complex data, it is still possible to separate and classify individuals.

5 Conclusion and Future Work

In this paper we have presented a novel implementation for modelling facial dynamics in a probabilistic framework using a GP-LVM and HMM on tracking data generated using AAMs. The results from our experiments, in particular with the NER set have demonstrated the discriminative power of combining temporal information with discrete states to identify individuals in a complex latent space that would otherwise be very difficult to do so with non-temporal techniques. Our initial findings have supported our beliefs that it is possible to distinguish individuals using only facial movement as dynamic features and have encouraged us to continue our investigation within this topic.

Currently we acknowledge that we are working with a small data set of individuals and for future work we will attempt to generalise our model to a larger data set with a wider range of subjects performing a more extensive set of actions. Additionally, in our current experiments although we have decoupled global head movement from the tracking data, we have yet to investigate how using global head movement alone for the purposes of identifying individuals compares with the use of facial expressions.

We have argued the importance of dynamics in facial data in our framework and have used a non-parametric dynamic model to calculate the transitional probabilities in our HMM trellis. It will be of great interest to investigate how different models for calculating transitional probabilities affect the discriminatory power of our current framework. Furthermore there is also scope to explore how efficient the GP-LVM is at synthesising individual specific facial motions. Although it is beyond the scope of this paper, in future we may consider augmenting our feature extraction processes by introducing high-speed video to capture more subtle facial motion.

Acknowledgements. We would like to express our thanks to Dr. Ian Penton-Voak and Dr. Ian Stephen from the Department of Experimental Psychology at the University of Bristol for kindly providing us the video data of individuals making up the NER set.

References

1. Tang, H., Fu, Y., Tu, J., Huang, T.S., Hasegawa-Johnson, M.: EAVA: A 3D Emotive Audio-Visual Avatar. In: WACV 2008, pp. 1–6 (2008)
2. Kähler, K., Haber, J., Seidel, H.P.: Reanimating the Dead: Reconstruction of Expressive Faces from Skull Data. In: SIGGRAPH 2003, pp. 554–561 (2003)
3. MacDorman, K.F., Green, R.D., Ho, C.C., Koch, C.T.: Too Real for Comfort? Uncanny Responses to Computer Generated Faces. Computers in Human Behavior 25, 695–710 (2009)
4. Ye, N., Sim, T.: Combining Facial Appearance and Dynamics for Face Recognition. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 133–140. Springer, Heidelberg (2009)
5. Yang, P., Liu, Q., Metaxas, D.: Dynamic Soft Encoded Patterns for Facial Event Analysis. Computer Vision and Image Understanding 115, 456–465 (2011)
6. Hadid, A., Pietikäinen, M., Li, S.Z.: Learning personal specific facial dynamics for face recognition from videos. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 1–15. Springer, Heidelberg (2007)
7. Fan, X., Sun, Y., Yin, B., Guo, X.: Gabor-based Dynamic Representation for Human Fatigue Monitoring in Facial Image Sequences. Pattern Recognition Letters 31, 234–243 (2010)
8. Raducanu, B., Dornaika, F.: Dynamic vs. Static recognition of facial expressions. In: Aarts, E., Crowley, J.L., de Ruyter, B., Gerhäuser, H., Pflaum, A., Schmidt, J., Wichert, R. (eds.) AmI 2008. LNCS, vol. 5355, pp. 13–25. Springer, Heidelberg (2008)
9. Dornaika, F., Lazkano, E., Sierra, B.: Improving Dynamic Facial Expression Recognition with Feature Subset Selection. Pattern Recognition Letters 32, 740–748 (2011)
10. 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)
11. Asthana, A., Saragih, J., Wagner, M., Goecke, R.: Evaluating AAM Fitting Methods for Facial Expression Recognition. In: ACII 2009, pp. 1–8 (2009)
12. Matthews, I., Baker, S.: Active Appearance Models Revisited. International Journal of Computer Vision 60, 135–164 (2004)

13. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time Combined 2D+3D Active Appearance Models. In: CVPR 2004, vol. 2, pp. 535–542 (2004)
14. Lawrence, N., Hyvärinen, A.: Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. Journal of Machine Learning Research 6, 1783–1816 (2005)
15. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based Inverse Kinematics. In: SIGGRAPH 2004, pp. 522–531 (2004)
16. Lawrence, N.D., Quiñonero Candela, J.: Local Distance Preservation in the GP-LVM through Back Constraints. In: ICML 2006, pp. 513–520 (2006)
17. Ek, C.H., Torr, P., Lawrence, N.D.: Gaussian Process Latent Variable Models for Human Pose Estimation. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds.) MLMI 2007. LNCS, vol. 4892, pp. 132–143. Springer, Heidelberg (2008)
18. Quirion, S., Duchesne, C., Laurendeau, D., Marchand, M.: Comparing GPLVM Approaches for Dimensionality Reduction in Character Animation. WSCG 16, 41–48 (2008)
19. Huang, M., Wang, Z., Ying, Z.: A Novel Method of Facial Expression Recognition Based on GPLVM Plus SVM. In: ICSP 2010, pp. 916–919 (2010)
20. Deena, S., Galata, A.: Speech-driven facial animation using a shared gaussian process latent variable model. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.-X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5875, pp. 89–100. Springer, Heidelberg (2009)
21. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive Database for Facial Expression Analysis. In: AFGR 2000, pp. 46–53 (2000)
22. Wallhoff, F.: Facial Expressions and Emotion Database (2006)
23. Ekman, P., Friesen, W.V., Hager, J.C.: Facial Action Coding System. [CD-ROM] (2002)
24. Ramer, U.: An Iterative Procedure for the Polygonal Approximation of Plane Curves. Computer Graphics and Image Processing 1, 244–256 (1972)
25. Zhu, Z., Ji, Q.: Real Time 3D Face Pose Tracking From an Uncalibrated Camera. In: CVPR 2004, vol. 73 (2004)
26. Akhter, I., Sheikh, Y.A., Khan, S., Kanade, T.: Nonrigid Structure from Motion in Trajectory Space. Neural Information Processing Systems (2008)
27. Arthur, D., Vassilvitskii, S.: K-Means++: The Advantages of Careful Seeding. In: SODA 2007, pp. 1027–1035 (2007)

Towards Temporally-Coherent Video Matting

Xue Bai, Jue Wang, and David Simons

Adobe Systems, Seattle, WA 98103, USA
{xubai,juewang,dsimons}@adobe.com

Abstract. Extracting temporally-coherent alpha mattes in video is an important but challenging problem in post-production. Previous video matting systems are highly sensitive to initial conditions and image noise, thus cannot reliably produce stable alpha mattes without temporal jitter. In this paper we propose an improved video matting system which contains two new components: (1) an accurate trimap propagation mechanism for setting up the initial matting conditions in a temporally-coherent way; and (2) a temporal matte filter which can improve the temporal coherence of the mattes while maintaining the matte structures on individual frames. Experimental results show that compared with previous methods, the two new components lead to alpha mattes with better temporal coherence.

1 Introduction

Video matting refers to the problem of separating a hairy or fuzzy foreground object from the background (either static or dynamic) by determining partial pixel coverage around the object boundary on each video frame. Mathematically, a boundary pixel I_p^t on frame t is modelled as a convex combination of a foreground color F_p^t and a background color B_p^t:

$$I_p^t = \alpha_p^t F_p^t + (1 - \alpha_p^t) B_p^t, \tag{1}$$

where $\alpha_p^t \in [0, 1]$ is the alpha matte that matting systems seek to solve for. Once estimated, the alpha matte can be used to create new composites, or as a layer mask for applying object-specific effects in the video. It is thus a key component in the video post-production pipeline.

The requirement for accurate video matting is two-fold. First, it requires *spatial accuracy*, which means that the alpha mattes extracted on individual frames should accurately represent the object to be extracted. Second, it demands *temporal coherence*, meaning that playing the extracted mattes in normal video speed should not result in noticeable temporal jitter. In fact, temporal coherence is often more important as the human visualization system (HVS) is more sensitive to temporal inconsistency when watching a video [1].

Due to the importance of temporal coherence, directly applying image matting approaches frame-by-frame for video is undesirable. Previous video matting approaches combine image matting methods with special temporal coherence

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 63–74, 2011.

treatments to try to meet both goals. However, as we will demonstrate later, these approaches still cannot reliably generate high quality, temporally-coherent alpha mattes even in relatively simple cases. There are many contributing factors, and we argue that the two main reasons are:

1. Overlooking the sensitivity of matting algorithms to initial conditions. Matting algorithms usually require an input trimap to start with, and modern approaches are very sensitive to how the trimap is defined on each frame. Changing the trimap even slightly may result in a large change in the final matte. Temporally-coherent video matting thus requires temporally-coherent trimap generation as the first step. Unfortunately the importance of generating coherent trimaps has been largely ignored in previous approaches.
2. The lack of a temporal filter that can improve the temporal coherence while maintaining spatial accuracy of the alpha mattes. Previous matting approaches try to impose some temporal coherence constraints when solving for the alpha matte on each frame. However due to image noise and other factors, the mattes generated in this way still contain temporal jitter.

In this paper we propose a new video matting system which contains two new components that explicitly handle the two problems mentioned above. Specifically, we propose a temporally-coherent trimap propagation method, which allows accurate trimap to be propagated from one frame to the next in a coherent way. We show that this leads to greatly-improved matte stability, as the initial matting conditions on adjacent frames are consistent. We further propose a novel temporal matte filter which, unlike other low-pass filters, can greatly improve the temporal coherence of the final alpha mattes. We combine these two new components with existing matting techniques to form an efficient video matting system. Experimental results suggest that our system can generate more temporally-coherent results than previous methods.

2 Related Work

Image matting techniques have been significantly advanced in recent years, and have been incorporated into various commercial products such as Adobe Photoshop. Leading techniques include the matting Laplacian [2], the shared sampling method [3] and learning-based approaches [4]. We refer readers to Wang and Cohen's comprehensive survey [5] on recent image matting techniques.

Various approaches have been proposed to extend image matting methods to video. These systems usually contain a binary segmentation module which allows the user to interactively generate a binary mask for the foreground object first, then create a trimap on each frame for alpha matting. The Bayesian video matting system [6] assumes the static background can be reconstructed from the input sequence, thus a background subtraction method can be used for automatic trimap generation. Once the trimaps are generated, the Bayesian matting [7] method is used in each frame to generate the final matte. The video object cut-and-paste system [8] and the 3D cutout system [9] both use graph

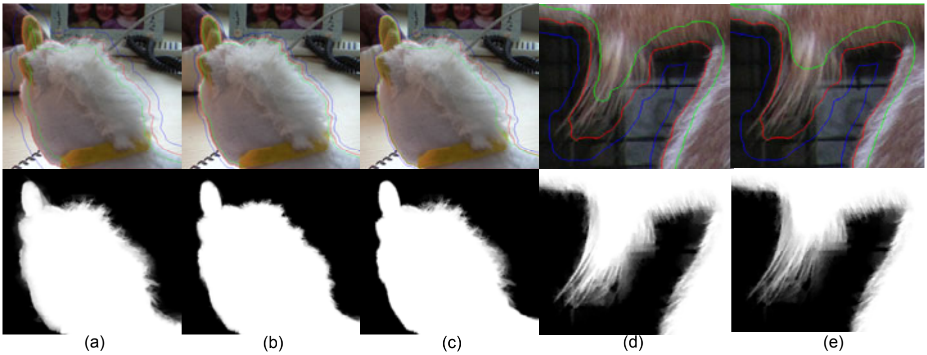


Fig. 1. Illustration on how the accuracy of the input trimap affects the final matte. (a) Top: Uniform bandwidth matting band generated by eroding (green) and dilating (blue) the binary segmentation boundary (red). Bottom: the corresponding matte. (b) A narrow matting band using a small bandwidth and the resulting matte. (c) Adaptive trimap generated by our system and the resulting matte. (d) An adaptive trimap and its resulting matte on another example. (e) Slightly modified trimap from (d) and the resulting matte.

cut optimization for binary segmentation. The binary masks are then eroded and dilated evenly to create the trimaps for matting. The recently proposed Video SnapCut system employs a more efficient binary segmentation tool based on localized classifiers [10]. It also uses a modified matting Laplacian formulation with an added temporal coherence term as the matting solution. Lee et al. [11] propose a 3D matting approach by treating video data as a spatio-temporal cube and extending the Robust matting approach [12] from 2D to 3D.

3 Limitations of Previous Approaches

Before introducing our techniques, we first analyze the limitations of previous video matting approaches, and explain why their resulting mattes are not temporally-coherent.

3.1 Inaccurate and Inconsistent Trimaps

Since the trimap is treated as a hard constraint for matting, matting algorithms are sensitive to even small changes to the trimap. Given a binary mask B^t, previous video matting approaches [8,9,10] usually create a uniform bandwidth unknown region by eroding and dilating B^t for a fixed number of pixels, as shown in Figure 1. We call the unknown region of the trimap the *matting band*. However, this uniform matting band is not accurate enough for objects with various lengths of hair around the boundary. An example is shown in Figure 1a-c. If the bandwidth is too large (Figure 1a), or too small (Figure 1b), the results of using the matting Laplacian method [2] contain various artifacts. A more accurate alpha matte could be achieved by using an adaptive matting band shown in Figure 1c, which is wider where the hair is longer, and is narrower where the boundary is nearly solid. Previous video matting systems are incapable of generating such trimaps.

However, using adaptive trimap alone is insufficient to guarantee the temporal coherence of the alpha mattes. For the same part of the object, if the local bandwidth is not consistent across frames, then the local mattes may have temporal jitter. An example is shown in Figure 1d-e. For the same part of the object, if we just change the inner boundary of the matting band a little bit, it could lead to a significant change in the final alpha matte. This suggests that in order to achieve temporally-coherent alpha matte, we not only need accurate adaptive trimap generation, but also need to make sure that the local bandwidth of the matting band to be temporally-consistent.

3.2 Weak Temporal Constraints

Previous video matting approaches usually contain a temporal coherence term in the matting formulation. For instance, the recently proposed video SnapCut system minimizes the following matting energy:

$$E(\alpha^t) = \arg\min_{\alpha^t} \sum_x \left[\lambda_x^T (\alpha_x^t - \hat{\alpha}_x^{t-1})^2\right] + E^S(\alpha^t), \tag{2}$$

where $E^S(\alpha^t)$ is the regular image matting energy which only involves frame t, and $\hat{\alpha}_x^{t-1}$ is the temporal prior which is essentially the matte computed on the previous frame and then warped by the optical flow. However, computing optical flow around the object boundary is often problematic, not to mention pixels that have partial foreground coverage. As the result, the warped alpha matte is often inaccurate to serve as a pixel-wise prior for the current frame. Furthermore, the temporal prior can only partially affect the final matte during the optimization process, thus the temporal coherence of the final mattes can still be poor, if the other terms in Equation 2 dominate the solution. We will show examples to illustrate this problem in Section 4.3.

4 Our Approach

4.1 System Framework

We develop a new video matting system that explicitly addresses the limitations of previous approaches. The system flow chart is shown in Figure 2. Given an input sequence, we first apply the Video SnapCut system [10] to interactively generate a binary mask for the target object on each frame. We then ask the user to use brush tools to specify accurate trimaps on some keyframes. The user-specified trimaps are then parameterized and propagated from keyframes to all other frames, using the method described in Section 4.2. Mattes are then computed given the trimaps, using the Robust Matting approach [12]. If the

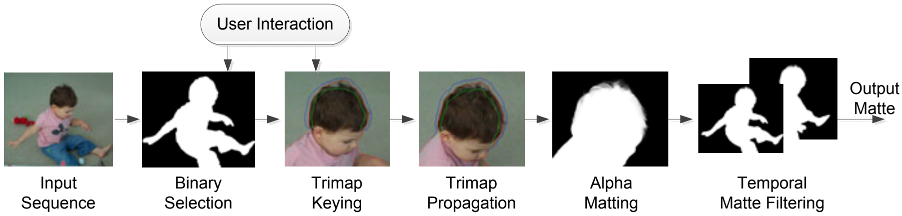


Fig. 2. The work flow of the proposed video matting system

mattes on some frames contain errors, the user can additionally modify the trimaps on those frames and the system will automatically propagate the user edits to neighboring frames to improve the mattes. Once the initial mattes are computed, we apply a temporal matte filter on them to improve their temporal coherence, as we will describe in Section 4.3.

4.2 Adaptive Trimap Propagation

Our adaptive trimap propagation method is illustrated in Figure 3. Given the binary segmentation boundary (red line in Figure 3a), we first ask the user to carefully specify the hair region of the object on a keyframe, as shown as light-yellow in Figure 3a. For unspecified parts of the object, a very tight trimap is generated by fattening the binary segmentation boundary. We then compute a matte under the current trimap, as shown in Figure 3b. The user iterates until a satisfying matte is achieved on the keyframe.

To parameterize the trimap, given any point p_i on the binary contour, we can define a local window W_i and compute a local inner and outer radius d_i^F and d_i^B, which together can cover all fractional alpha pixels in the local window, as shown in Figure 3b. By sampling along the object contour uniformly, we connect a set of control points $\{p_i, d_i^F, d_i^B\}$,$i = 1, ..., M$, as shown in Figure 3c.

To propagate the parameterized trimap shape, we first use the optical flow field computed between the current (frame t) and the next frame (frame $t+1$) to push the set of control points to new locations $\{p'_i, d_i^F, d_i^B\}$, based on the object motion. Due to optical flow errors and topology changes, the moved control points p'_i may not be on the binary segmentation boundary on frame $t+1$. To assign the radius values to points on the object boundary, we employ a thin-plate interpolation method [13]. Specifically, we compute the interpolation function as

$$f(x, y) = c_0 + c_x x + c_y y + \sum_{i=1}^{M} c_i \phi(\|(x, y) - p'_i\|), \tag{3}$$

where $\phi(r) = r^2 \log r$ is the thin plate spline function, and the coefficients $c_0, c_x, c_y, c_1, ..., c_M$ are solved by minimizing smoothing thin plate spline energy function (see [13] for details). We then uniformly sample a new set of control points around the object boundary on frame $t+1$ as $\{q_j\}$,$j = 1, ..., M$. The inner

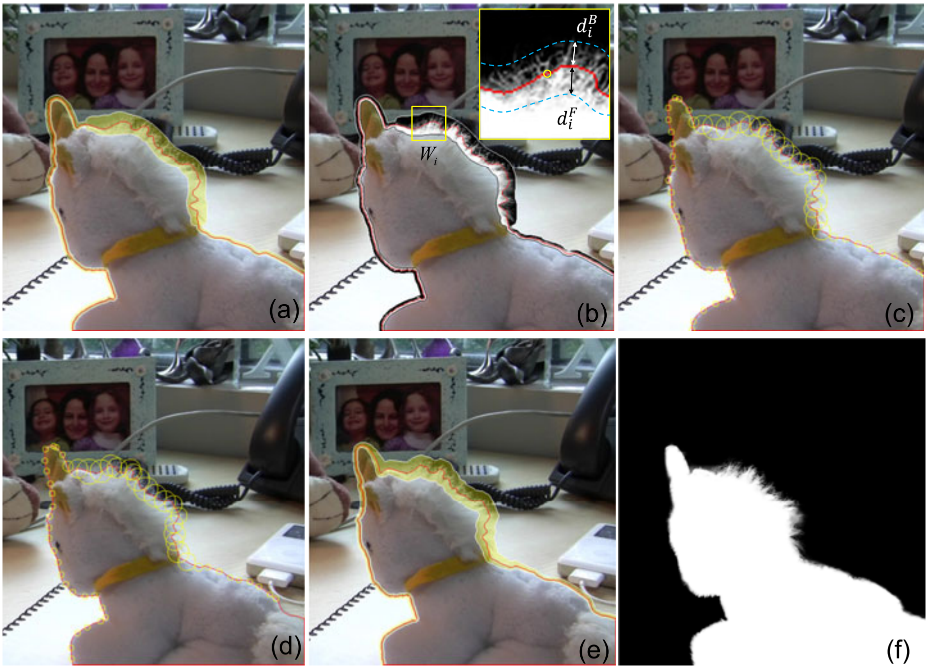


Fig. 3. Illustration of the adaptive trimap propagation. (a) User-specified trimap on a keyframe. (b) Automatically computed local trimap radii based on the matte. (c) Compute a set of control points around the object. (d). Propagate local radii to the next frame. (e). Rasterized trimap on the next frame. (f). Computed matte on the next frame based on the propagated trimap.

trimap radius at each new control point is $d^F(q_j) = f(q_j)$. The outer trimap radius $d^B(q_j)$ is computed in a similar fashion. Once the local radii for all control points are computed (Figure 3d), we rasterize the trimap and compute the matte on the new frame, as shown in Figure 3e-f.

The above process describes how we propagate the trimap from frame t to $t+1$. To further propagate the trimap, we need to update the radius values $d^F(q_j)$ and $d^B(q_j)$, based on the newly-computed alpha matte on frame $t + 1$. One straightforward idea is to update all radius values based on the alpha matte, as we do on the keyframe. However, in practice we found that this solution quickly leads to deteriorated trimaps, since the computed alpha matte may contain errors, and these errors will in turn introduce more errors in the trimap. We thus have to update the radius values conservatively, i.e., only update the local trimap radius when we have high confidence on the local alpha matte.

Our key observation is that the matte quality is directly related to local foreground and background color distributions. If the colors are distinct, the matte is usually accurate. On the contrary, matte error is usually introduced when the foreground and background color distributions overlap. For every control point

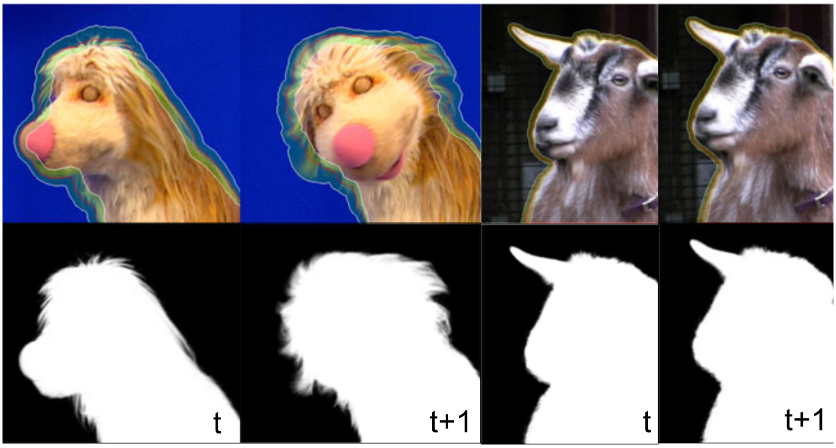


Fig. 4. Adaptive trimap update on two examples using Equation 5

q_j we thus compute a local matte confidence f_j, based on the local color distributions. Specifically, inside the local window W_j centered at q_j, we sample a group of foreground and background colors based on the computed alpha matte as F_k and B_k,$k = 1, ..., S$. We then estimate a foreground Gaussian mixture Model (GMM) based on F_ks, and a background GMM based on B_ks, denoted as G^F and G^B, respectively. The matte confidence f_j is computed as

$$f_j = \frac{1}{S}\sum_{k=1}^{S}\left(1 - \frac{G^F(B_k) + G^B(F_k)}{2}\right), \tag{4}$$

where $G^F(x)$ is the probability measured by the GMM model G^F given a color x. Since we feed background colors B_k to the foreground GMM G^F, if the foreground and background colors are well-separable, both $G^F(B_k)$ and $G^B(F_k)$ should be small, thus the matte confidence f_j is high. Once f_j is computed, the local trimap radius is updated as

$$d^{F/B}_{final}(q_j) = (1 - f_j)d^{F/B}(q_j) + f_j\widehat{d}^{F/B}(q_j), \tag{5}$$

where $d(q_j)$ is the propagated radius computed using Equation 3, and $\widehat{d}(q_j)$ is the new radius computed from the matte. When the local matte confidence is low, we basically freeze the trimap radius update so that the trimap can stay stable when the matte cannot be trusted to avoid divergence. Two examples are shown in Figure 4. For the left example where the foreground and background color distributions are separable and the matte confidence is high, our system allows the trimap to be freely updated based on the estimated mattes. For the right example where the color distributions are mixed, our system freezes the trimap radius update to allow stability.

4.3 The Temporal Matte Filter

As we discussed earlier, due to various contributing factors such as scene color changes and image noise, mattes generated on individual frames often contain a certain degree of temporal jitter. In this section we introduce a novel temporal matte filter which can help reduce the jitter and improve the temporal coherence of the final mattes.

Level Set based Matte Interpolation. Our temporal filter is based on the level set parametrization of the grayscale alpha matte. We first describe how to interpolate between two matte images, i.e., given two alpha mattes α^1 and α^2, generate an in-between matte $\alpha^* = f_I(\alpha^1, \alpha^2, \beta)$, where $\beta \in [0, 1]$ is the interpolation coefficient. Directly applying pixel-wise interpolation will not work since the two mattes are not aligned due to object or camera movement. We instead use a level set based interpolation approach. As shown in Figure 5a-d, given two input mattes, we first parameterize them using level set curves, where each curve is defined as the boundary of the iso-level region: $h_i = \partial M_i, M_i = \{x|\alpha(x) > i/K\}$,$i = 0, ..., K-1$. Note that each curve has a signed normal pointing to the descendent direction of the alpha matte.

Given two level sets $\{h_i^1\}$ and $\{h_i^2\}$ (Figure 5b,d), we compute an interpolated level set (Figure 5f) by interpolating each pair of corresponding curves h_i^1 and h_i^2, using the distance-transform-based method shown in Figure 5e. We first apply a signed distance transform on h_i^1 and h_i^2, denoted as D_1 and D_2. We then average the two distance transform fields to create a new distance field $D^* = (D_1 + D_2)/2$ (assuming $\beta = 0.5$), and then threshold it to an average binary shape M^* as the interpolation result. The contour of M^* is the interpolated curve h_i^*.

Once we have the interpolated level set $\{h_i^*\}$, as shown in Figure 5g, for a pixel x on the image plane, we first identify its shortest distances to the nearest two level set curves h_i^* and h_{i-1}^* as $d_i(x)$ and $d_{i-1}(x)$. The alpha value of x is then interpolated as:

$$\alpha(x) = \gamma \cdot \frac{i}{K} + (1-\gamma) \cdot \frac{i-1}{K}, \text{ where } \gamma = \frac{d_{i-1}(x)}{d_{i-1}(x) + d_i(x)}. \qquad (6)$$

By applying Equation 6 to all pixels between the inner and outer level set curves h_K^* and h_0^*, and assigning other pixels outside these two curves to be 1 or 0, we can reconstruct an interpolated matte as shown in Figure 5h. As we can see the interpolated matte maintains the same alpha profile with the two input mattes, despite that the two input mattes have a large shape difference.

Temporal Matte Filtering. We use the matte interpolation method described above for temporal matte filtering. Given mattes on three adjacent frames α^{t-1},α^t and α^{t+1}, our temporal filter is defined as:

$$\alpha_{new}^t = f_I\left(\alpha^t, f_I(\alpha^{t-1}, \alpha^{t+1}, 0.5), 0.5\right), \qquad (7)$$

where f_I is the level-set-based matte interpolation procedure described above. Essentially the temporal filter is a weighted interpolation of three mattes using

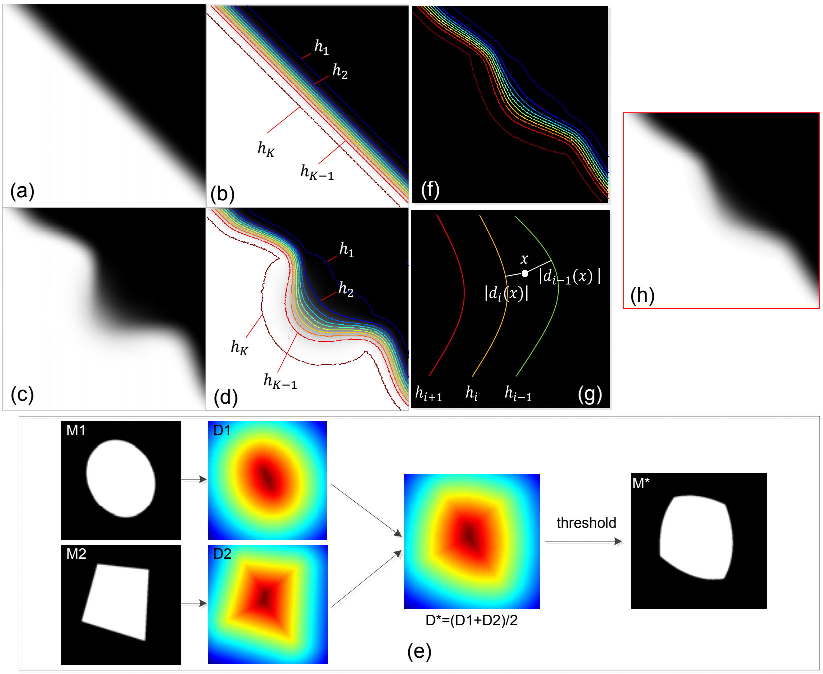


Fig. 5. Illustration of the temporal filter. (a-b) The first input matte and its level set curves. (c-d) The second input matte and its level set curves. (e) Distance transform based curve interpolation. (f) Interpolated level set curves. (g) The alpha value of any pixel is determined by interpolation between nearest level set curves. (h) Final interpolated alpha matte.

the same level set interpolation procedure, and the weights for the three frames are $\{0.25, 0.5, 0.25\}$. The filter could be applied iteratively over a chunk of frames to achieve stronger temporal smoothing effect.

Figure 6 compares the proposed temporal filter with other temporal smoothing approaches. Given three input mattes, the simplest solution is to apply a pixel-wise temporal Gaussian filter on them. However, as shown in the figure, since the matte structures are not well-aligned, pixel-wise interpolation results in a significantly blurred foreground edge. Another approach is to treat α^{t-1} and α^{t+1} as priors, and re-solve α_t using Equation 2. However, this method also produces blurry results and destroys the underlying matte structure. On the contrary, our temporal filter is able to effectively reduce the temporal jitter while maintaining the original matte structure.

5 Results and Comparisons

To demonstrate the effectiveness of the system, we applied it on a variety of examples and compared the results with those generated by other approaches. Figure 7 shows a few representative examples. For each example, we compare

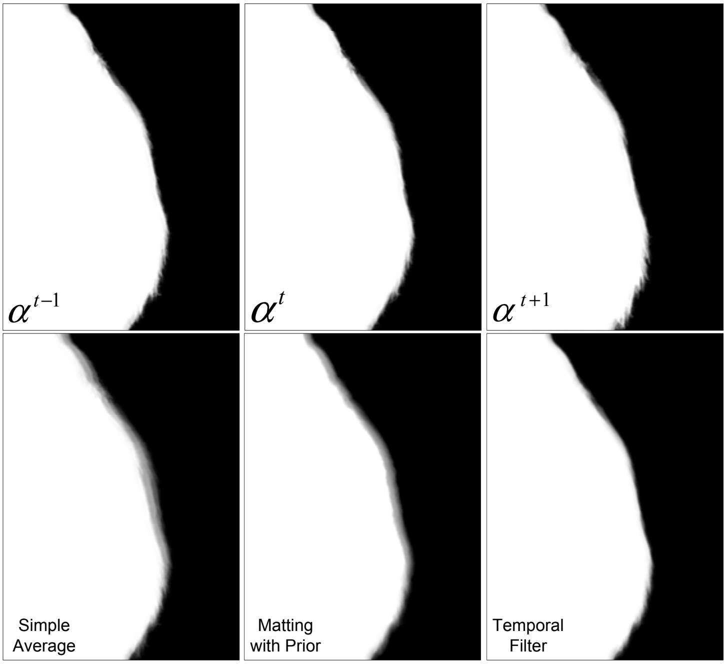


Fig. 6. Compare our temporal filter with other methods. Top: input mattes on three consecutive frames. Bottom: smoothed α^t by three methods: simple pixel-wise averaging; solving Equation 2 using both α^{t-1} and α^{t+1} as priors; our temporal filter.

results generated by the following methods: commercial software Keylight from The Foundry[1], matting with uniform trimap, matting with adaptive trimap without temporal filtering, and matting with both adaptive trimap and temporal filtering. Due to the limited space, we only show detailed comparison on the "girl" example. Since this example contains moving and textured background, traditional blue/green screen keying method such as Keylight cannot really work, as shown in Figure 7a. Figure 7b-c show the uniform trimap approach and its resulting matte. The trimap is too narrow for the hair region, thus the matte in the hair region is less soft than desired. The trimap is also too wide for lower body, which introduces a lot of noise in the matte. Using the adaptive trimap generation method proposed in this paper, we can get substantially better mattes as shown in Figure 7d-e. This matte is further improved after the temporal filtering process, as shown in Figure 7f.

The benefit of the temporal filtering can only be seen when playing videos at the normal speed. To demonstrate the effectiveness of the temporal filter, we create a supplemental video which contains detailed comparisons on all three examples with and without applying the temporal filter. The video can be downloaded at http://www.juew.org/mirage11/videoMatting.mp4.

[1] http://www.thefoundry.co.uk/products/keylight/

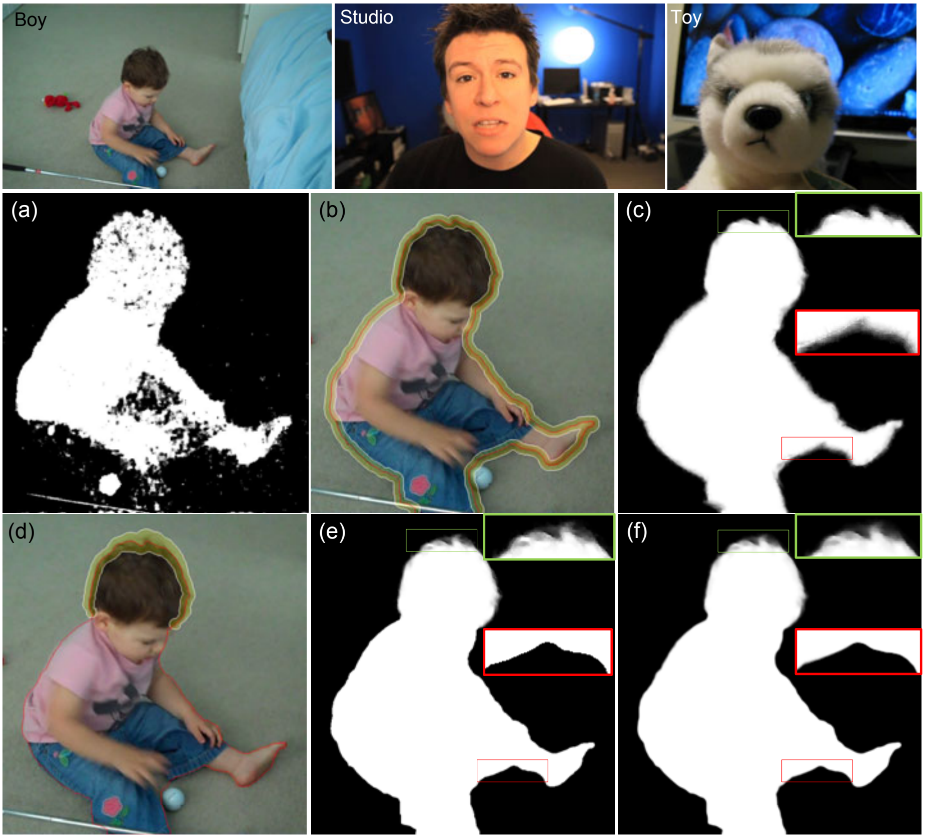


Fig. 7. Examples and comparisons. Top row: examples used for comparison. Bottom: comparisons on the "girl" example. (a) Matte generated by Keylight. (b) Uniform trimap. (c) Alpha matte generated using trimap (b). (d) Our variable bandwidth trimap. (e) Alpha matte generated using trimap (d). (f) Final alpha matte after temporal smoothing.

6 Conclusion

We propose a new video matting system which contains two new components: an adaptive trimap propagation procedure and a temporal matte filter. The adaptive trimap propagation method allows an accurate and consistent trimap to be generated on each frame, which sets up a temporally-coherent initial condition for the matte solver. The temporal matte filter can further improve the temporal-coherence of the alpha mattes while maintaining the matte structure on each frame. Combining these new components with previous image matting methods, our system can generate high quality alpha mattes with better temporal-coherence than previous approaches.

Despite the progress we made in this paper, video matting still remains a challenging problem in difficult cases. When the background contains strong textures and similar colors to the foreground, extracting high quality matte on each single frame may not be possible using existing image matting approaches. Another limitation of the system is that the temporal filter is efficient on improving the temporal stability of the mattes, but it also has a tendency to smooth out small scale, fine matte structures. Future research has to address these problems in order to develop a better video matting system.

References

1. Villegas, P., Marichal, X.: Perceptually-weighted evaluation criteria for segmentation masks in video sequences. IEEE Trans. Image Processing 13, 1092–1103 (2004)
2. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. on Pattern Analysis and Machine Intelligence 30, 228–242 (2008)
3. Gastal, E.S.L., Oliveira, M.M.: Shared sampling for real-time alpha matting. Computer Graphics Forum 29(2), 575–584 (2010); Proceedings of Eurographics
4. Zheng, Y., Kambhamettu, C.: Learning based digital matting. In: Proc. IEEE International Conference on Computer Vision (2009)
5. Wang, J., Cohen, M.: Image and video matting: A survey. Foundations and Trends in Computer Graphics and Vision 3, 97–175 (2007)
6. Chuang, Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. In: Proceedings of ACM SIGGRAPH, pp. 243–248 (2002)
7. Chuang, Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proc. of IEEE CVPR, pp. 264–271 (2001)
8. Li, Y., Sun, J., Shum, H.: Video object cut and paste. In: Proc. ACM SIGGRAPH, pp. 595–600 (2005)
9. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Interactive video cutout. ACM Trans. Graph. 24, 585–594 (2005)
10. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. 28, 70:1–70:11 (2009)
11. Lee, S.Y., Yoon, J.C., Lee, I.K.: Temporally coherent video matting. Graphical Models 72, 25–33 (2010)
12. Wang, J., Cohen, M.: Optimized color sampling for robust matting. In: Proc. IEEE CVPR (2007)
13. Wahba, G.: Spline models for observational data. In: CBMSNSF Regl. Conf. Ser. Appl. Math., vol. 59 (1990)

Leaf Segmentation and Tracking Using Probabilistic Parametric Active Contours

Jonas De Vylder[1,⋆], Daniel Ochoa[1,2], Wilfried Philips[1],
Laury Chaerle[3], and Dominique Van Der Straeten[3]

[1] Department of Telecommunications and Information Processing,
IBBT - Image Processing and Interpretation, Ghent University
[2] Facultad de Ingenieria en Electricidad y Computacion, Escuela Superior Politecnica del Litoral
[3] Department of Physiology, Laboratory of Functional Plant Biology, Ghent University
jonas.devylder@telin.ugent.be
http://telin.ugent.be/~jdvylder

Abstract. Active contours or snakes are widely used for segmentation and tracking. These techniques require the minimization of an energy function, which is generally a linear combination of a data fit term and a regularization term. This energy function can be adjusted to exploit the intrinsic object and image features. This can be done by changing the weighting parameters of the data fit and regularization term. There is, however, no rule to set these parameters optimally for a given application. This results in trial and error parameter estimation. In this paper, we propose a new active contour framework defined using probability theory. With this new technique there is no need for ad hoc parameter setting, since it uses probability distributions, which can be learned from a given training dataset.

1 Introduction

With a constantly increasing demand for food it becomes necessarily to optimize agricultural planning, e.g. to plant the best type of plants and use the best fertilizers for a given field with a specific soil, expected weather, etc. This off course assumes it is known what the best plant type is for a specific field. This has led to the development of biological laboratories which quantitatively measure the development of plants under the influence of specific stress factors, e.g. wind, lack of nutrients, etc. A common feature to evaluate the well-being of a plant is to periodically measure the average leaf temperature using a thermal camera.

The reliable measuring of average leaf temperature in thermal images is a time consuming task. It demands skilled technicians who spend time identifying and delineating objects of interest in the image. Although interactive software can ease this work, this approach becomes impractical when the measurements have to be monitored over time for a large variety of plant types. This paper proposes an automated technique to segment leaves and measure its average temperature in thermal images. An interesting approach to segment objects is based on probability theory [1,2]. In this work, a new Bayesian technique is proposed. This new technique combines the Bayesian framework with the popular active contour model, an extensively studied segmentation and tracking framework used in computer vision.

[⋆] Corresponding author.

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 75–85, 2011.

In the active contour framework, an initial contour is moved and deformed in order to minimize a specific energy function. This energy function should be minimal when the contour is delineating the object of interest, e.g. a leaf. Two main groups can be distinguished in the active contour framework: one group representing the active contour explicitly as a parameterized curve and a second group which represents the contour implicitly using level sets. In the first group, also called snakes, the contour generally converges towards edges in the image [3,4,5]. The second group generally has an energy function based on region properties, such as the intensity variance of the enclosed segment [6,7]. These level set approaches have gained a lot of interest since they have some benefits over snakes. They can for example easily change their topology, e.g. splitting a segment into multiple unconnected segments. Recently an active contour model has been proposed with a convex energy function, making it possible to define fast global optimizers [8,9].

Unfortunately do the level set approaches assume certain prior knowledge about the regions defined, e.g. the variance of intensity in the segments should be minimal. These kind of assumptions are unfortunately not always valid and might bias the temperature measurements of leaves. Therefore we will base our new active contour model on the snake approach. This doesn't form a problem since leaves don't divide and therefore we don't need the variable topology of level set active contours. Although snakes don't have global optimizers, several optimizations techniques have been proposed and proven useful for segmentation and tracking. In order not to converge to local optima, one or more regularization terms are incorporated in the energy function. The influence of these regularization terms can be tuned using a set of weighting parameters. This tuning is generally done by trial and error, which is a time consuming and error prone approach. Even after manually tuning, the parameters might not be optimal since the segmentation quality in function of these weighting parameters generally is not a convex function. So only by exploring the full parameter space one can be sure to find the optimal parameters. We propose a new active contour framework based on probability theory. Instead of exhaustively searching optimal weighting parameters, the proposed method uses prior knowledge about the probability of certain features, e.g. edges. It also removes the linear influence of image features, i.e. it is not because the gradient of an edge is twice as large as another edge, that it is twice as likely to be the true border of an object. This is especially important if you are segmenting leaves in noisy images with other objects.

This paper is arranged as follows. The next section provides a detailed description of parametric active contours. Both the classical and gradient vector flow snakes are explained. In section 3 our proposed algorithm is presented. Section 4 elaborates on the results and compares the proposed methods with the classical snakes. Finally, section 5 recapitulates and lists some future research possibilities.

2 Active Contours

2.1 Snakes

The classical snake model proposed by Kass et al. [4], defines the active contour as a parametric curve, $\mathbf{r}(s) = (x(s), y(s))$ with $s \in [0, 1]$, that moves in the spatial domain until the energy functional in Eq. (1) reaches its minimum value.

$$E[\mathbf{r}] = E_{int}(\mathbf{r}(s)) + E_{ext}(\mathbf{r}(s)) \tag{1}$$

$E_{int}[.]$ and $E_{ext}[.]$ represent the internal and external energy, respectively corresponding to a regularization and a data fit term. A common internal energy function that enforces smoothness along the contour is defined as follows:

$$E_{int}[\mathbf{r}(s)] = \frac{1}{2}\int \alpha\left|\frac{\partial \mathbf{r}(s)}{\partial s}\right|^2 + \beta\left|\frac{\partial^2 \mathbf{r}(s)}{\partial ds^2}\right|^2 ds \tag{2}$$

where α and β are weighting parameters. The first term, also known as tension energy, prevents the snake to ”stretch” itself too much, thus avoid being attracted to isolated points. The second term, known as bending energy, prevents the contour of developing sharp angles. More complex internal energy functions, e.g. incorporating prior shape knowledge, have also been reported in literature [10,11].

The external energy is derived from the image, so that the snake will be attracted to features of interest. Given a grey level image $I(x, y)$, a common external energy is defined as:

$$E_{ext}[\mathbf{r}] = \int -\left|\nabla I(\mathbf{r}(s)\right|^2 ds \tag{3a}$$

or

$$E_{ext}[\mathbf{r}] = \int -\left|\nabla\big(G_\sigma(x, y) * I(\mathbf{r}(s))\big)\right|^2 ds \tag{3b}$$

where ∇ is the gradient operator, $G_\sigma(x, y)$ a 2D Gaussian kernel with standard deviation σ and where $*$ is the convolution operator.

2.2 Optimization

Eq. (1) can be minimized using gradient descent by treating $\mathbf{r}(s)$ as a function of time, i.e. $\mathbf{r}(s, t)$. The partial derivative of $\mathbf{r}$ with respect to t is then

$$\frac{d\mathbf{x}(s,t)}{dt} = \alpha\frac{d^2\mathbf{x}(s,t)}{ds^2} - \beta\frac{d^4\mathbf{x}(s,t)}{ds^4} - \frac{\partial E_{ext}}{\partial x} \tag{4a}$$

$$\frac{d\mathbf{y}(s,t)}{dt} = \alpha\frac{d^2\mathbf{y}(s,t)}{ds^2} - \beta\frac{d^4\mathbf{y}(s,t)}{ds^4} - \frac{\partial E_{ext}}{\partial y} \tag{4b}$$

The snake stabilizes, i.e. an optimum is found, when the terms $\frac{d\mathbf{x}(s,t)}{dt}$ and $\frac{d\mathbf{y}(s,t)}{dt}$ vanish.

This gradient descent approach requires a good initialization, close to the object boundary, in order to segment the object. This limitation is caused by the nature of the partial derivatives of the external energy, which differs from the null vector only in the proximity of the object's boundary. As we move away from the boundary these derivatives approach the null vector, or under the influence of noise point towards false optima. This results in a contour which will converge to a local optimum. To overcome this problem, Xu and Prince [12] proposed to replace the partial derivatives by an external force $\mathbf{v}(\mathbf{r}(s)) = (u(\mathbf{r}(s)), v(\mathbf{r}(s)))$. This force is calculated by minimizing the following energy functional:

$$E_{GVF}[u,v] = \iint \mu\Big(\frac{du}{dx}^2 + \frac{du}{dy}^2 + \frac{dv}{dx}^2 + \frac{dv}{dy}^2\Big) + \mid \nabla f \mid^2 \mid \mathbf{v} - \nabla f \mid^2 dxdy \quad (5)$$

where μ is a nonnegative parameter expressing the degree of smoothness of the force field $\mathbf{v}$ and where f is an edge map, e.g. $f(x,y) = \mid \nabla I(x,y) \mid$. The first term of Eq. (5) keeps the field $\mathbf{v}$ smooth, whereas the second term forces the field $\mathbf{v}$ to resemble the original edge map in the neighbourhood of edges. This new external force is called *gradient vector flow* (GVF). For details on the optimization of Eq. (5) , we refer to [12].

3 Probabilistic Active Contours

The active contour framework has already been proven useful for a wide range of applications. However, tuning the weighting parameters of the energy function remains a challenging task. The optimal parameters are a trade-off, where the regularization weights have to be set high enough to overcome the influence of clutter and low enough to accurately detect the true contour of the object. In this section, a new set of active contours is defined. This framework is based on statistical modelling of object features, thus omitting the tuning of the weighting parameters.

3.1 Framework

The goal of our active contour framework is to find the contour, $\mathbf{r}^*(.)$, which is most probable to delineate the object of interest. This can be formalized as finding the contour that maximizes $P[O[\mathbf{r}(.)]]$, where $O[\mathbf{r}(.)]$ is a predicate returning true if the contour delineates the object of interest and returns false if it does not. Let's assume that in order to find such a probable contour we can use a set of features $F(x,y)$ measured in the image, e.g. the edge strength at a specific pixel. The optimal contour can be defined as

$$\mathbf{r}^*(.) = \arg\max_{\mathbf{r}(.)} P\big[O[\mathbf{r}(.)]\big|F(.,.)\big] \quad (6)$$

Using Bayes rule, we can rewrite this as

$$\mathbf{r}^*(.) = \arg\max_{\mathbf{r}(.)} \frac{P\big[F(.,.)\big|O[\mathbf{r}(.)]\big]P\big[O[\mathbf{r}(.)]\big]}{P\big[F(.,.)\big]} \quad (7)$$

$$= \arg\max_{\mathbf{r}(.)} \Bigg(\log \frac{P\big(F(.,.)\big|O[\mathbf{r}(.)]\big)}{P\big(F(.,.)\big)} + \log P\big(O[\mathbf{r}(.)]\big)\Bigg) \quad (8)$$

Equivalent to the snake energy, we can define an internal and external probability, respectively: $P_{int}[\mathbf{r}(.)] = \log P\big(O[\mathbf{r}(.)]\big)$ and $P_{ext}[\mathbf{r}(.)] = \log \frac{P\big(F(.,.)\big|O[\mathbf{r}(.)]\big)}{P\big(F(.,.)\big)}$.

3.2 Internal Probability

The internal probability is completely independent of the image and can be used to incorporate the shape possibility of an object of interest. As an example we will use a simple model proposed in [13], where the likeliness of a contour only depends on the second derivative of the contour.

$$\begin{aligned} P_{int}[\mathbf{r}(.)] &= \log P\big[O[\mathbf{r}(.)]\big] \\ &= \log P\Big(\Big|\frac{\partial^2 \mathbf{r}(s)}{\partial s^2}\Big|\Big] \\ &= \int_0^1 \log P\Big(\Big|\frac{\partial \mathbf{r}^2(t)}{\partial t^2}\Big|\Big) dt \end{aligned} \tag{9}$$

For this last step we assume that the second derivative of $\mathbf{r}(\mathbf{t})$ is independent for every t. This off course assumes that the probability distribution of $P\Big(\Big|\frac{\partial \mathbf{r}(t)}{\partial t^2}\Big|\Big)$ is known. This probability distribution can be learned out of a small training set of ground truth segments. Note that this is just an example of an internal probability model. If the objects of interest has a specific shape or if they have more pronounced local features, e.g. jags, then the internal probability could be formulated using a more complex shape model, such as the models proposed in [11,10].

3.3 External Probability

The external probability depends on the image features that are used to characterize an object. As example we will model the objects of interest as an edge map, e.g. $F(x,y) =\mid \nabla I(x,y) \mid$. If we consider the gradient to be independent for all (x,y), then the external probability can be rewritten as:

$$\begin{aligned} P_{ext}[\mathbf{r}(.)] &= \log \frac{P\big[F(.,.)\big|O[\mathbf{r}(.)]\big]}{P\big[F(.,.)\big]} \\ &= \log \frac{P\big[\mid \nabla I(.,.) \mid \big|O[\mathbf{r}(.)]\big]}{P\big[\mid \nabla I(.,.) \mid\big]} \\ &= \iint\limits_{\Omega^+} \log \frac{P\big(\mid \nabla I(u,w) \mid \ \big|O[\mathbf{r}(.)]\big)}{P\big(\mid \nabla I(u,w) \mid\big)} du\, dw \end{aligned} \tag{10}$$

$$+ \iint\limits_{\Omega^-} \log \frac{P\big(\mid \nabla I(u,w) \mid \ \big|O[\mathbf{r}(.)]\big)}{P\big(\mid \nabla I(u,w) \mid\big)} du\, dw \tag{11}$$

where $\Omega^+ = \{(u,w)\big|\big(\exists k \in [0,1]\big)\big(\mathbf{r}(k) = (u,w)\big)\}$ and $\Omega^- = \{(u,w)\big|\big(\nexists k \in [0,1]\big)\big(\mathbf{r}(k) = (u,w)\big)\}$. For the application of plant monitoring, imaging happens in a strictly controlled environment. Due to this controlled imaging, technicians can avoid clutter and thus minimize edges not coming from leaf contours.

Therefore $\iint\limits_{\Omega^-} \log \frac{P\left(|\nabla I(u,w)| \,\middle|\, O[\mathbf{r}(.)]\right)}{P(|\nabla I(u,w)|)} du\; dw$ will be very small. This allows us to approximate Eq. (11) by discarding this factor, i.e.

$$P_{ext}[\mathbf{r}(.)] = \int_0^1 \log \frac{P(\mid \nabla I(\mathbf{r}(t)) \mid \ \big| O[\mathbf{r}(.)])}{P(\mid \nabla I(\mathbf{r}(t)) \mid)} dt \tag{12}$$

The probabilities used in Eq. (12) can be interpreted as follows:

- $P(\mid \nabla I(\mathbf{r}(t)) \mid \ \big| O[\mathbf{r}(.)])$: the probability that the gradient of a point lying on the contour of a real segment is equal to $\mid \nabla I(\mathbf{r}(s)) \mid$.
- $P(\mid \nabla I(\mathbf{r}(s)) \mid \ \big| \mathbf{r}(s))$: the probability that the gradient of a random point in the image is equal to $\mid \nabla I(\mathbf{r}(s)) \mid$ whether or not this point is on the contour of a real object or not.

The probability distribution of the gradient strength of an object's contour can be estimated from a small training set of images where the objects are manually segmented. First measure the gradient strength at the contours delineating the ground truth segments. Then based on these measurements, calculate the probability distribution, e.g. using a kernel density estimator. The probability distribution of the gradient can be estimated in a similar way, but instead of measuring only the gradient strength at the contours, measure it at each pixel in the training data set. Note that although this example uses the gradient, this framework could also incorporate other image features such as ridges, intensity, output of a feature detector, region-based features, etc. [1,14,6].

3.4 Optimization

Substituting the proposed internal in external probabilities in Eq. (8) results in:

$$\begin{aligned}\mathbf{r}^*(.) &= \underset{\mathbf{r}(.)}{\arg\max}\left(P_{int}\Big[\mathbf{r}(.)\Big] + P_{ext}\Big[\mathbf{r}(.)\Big)\right] \\ &= \underset{\mathbf{r}(.)}{\arg\max} \int_0^1 \log \frac{P(\mid \nabla I(\mathbf{r}(s)) \mid \ \big| O[\mathbf{r}(.)])}{P(\mid \nabla I(\mathbf{r}(s)) \mid)} + \log P\left(\left|\frac{\partial \mathbf{r}^2(s)}{\partial s^2}\right|\right) ds\end{aligned} \tag{13}$$

This optimization can be solved using gradient descent by treating $\mathbf{r}(s)$ as a function of time, i.e. $\mathbf{r}(s,t)$. The partial derivative of $\mathbf{r}$ with respect to t is then

$$\frac{dx(s,t)}{dt} = \frac{d \log P\left(\left|\frac{\partial \mathbf{r}^2(s,t)}{\partial s^2}\right|\right)}{ds} + \frac{\partial \log \frac{P\left(|\nabla I(\mathbf{r}(s,t))| \,\middle|\, O[\mathbf{r}(.,t)]=\text{true}\right)}{P(|\nabla I(\mathbf{r}(s,t))|)}}{\partial x} \tag{14a}$$

$$\frac{dy(s,t)}{dt} = \frac{d \log P\left(\left|\frac{\partial \mathbf{r}^2(s,t)}{\partial s^2}\right|\right)}{ds} + \frac{\partial \log \frac{P\left(|\nabla I(\mathbf{r}(s,t))| \,\middle|\, O[\mathbf{r}(.,t)]=\text{true}\right)}{P(|\nabla I(\mathbf{r}(s,t))|)}}{\partial y} \tag{14b}$$

In order to use gradient descent, the initial $\mathbf{r}(s)$ should be in the vicinity of the true object. To avoid the probabilistic snake of converging to a local, false optimum, the same optimization technique as used with classical snakes can be used, i.e. optimization using gradient vector flow. This can easily be done by imposing the edge map in Eq. 5 to be $F(x, y) = \log \frac{P\left(|\nabla I(x,y)| \,\middle|\, O(x,y)=\text{true}\right)}{P(|\nabla I(x,y)|)}$, where $O(x, y) = \text{true}$ represents the assumption that (x, y) lies on the contour of a leaf.

4 Results

The proposed method was developed to automate the measurement of average temperature of leaves. Therefore individual leaves should be segmented and tracked over time. The dataset used to validate the proposed method monitors sugar beet seedling plants using a thermal camera. The time-lapse sequences were captured at a time resolution of one image an hour. These thermal images are noisy, low contrast greyscale images. In these time-lapse sequences the 4 leaves of the sugar beet seedlings all move in different directions, at different speeds.

Fig.1 shows an example of leaf segmentation using both the classical active contours as our proposed probabilistic active contours. All active contours are optimized using the gradient vector flow optimization. The gradient vector flow force was calculated using 30 iterations and a smoothing factor μ equal to 0.1. In Fig.1 (a) the initialization of the four different snakes is shown. As can be seen, is this initialization already a good approximation of the real leaf contours. This proper initialization is however not sufficient for the classical active contours to converge to the real leaf boundaries. Two examples of active contour segmentation using different weighting parameters α and β in Eq. (2) can be seen in Fig.1 (b) and (c). The active contours in Figure (b) were optimized using $\frac{1}{6}$ and $\frac{1}{3}$ respectively for weighting parameters α and β. These weighting values are apparently too low to prevent the contours to converge to false local optima. An example of such an incorrect convergence can be seen at the yellow contour, which partially converged towards the border of a wrong leaf. The cause of these segmentation errors is the difference in the gradient strength. Figure (d) shows the absolute value of the gradient in the image. The bigger leaves display a much stronger gradient which attract the contours of the elongated smaller leaves. Note this effect near the stalk of the "yellow" leaf. In Figure (c), the yellow contour converged correctly by increasing the contour weighting parameters, i.e. values $\frac{5}{3}$ and $\frac{2}{3}$ respectively for parameters α and β. However due to these strong smoothness constraints, the green and blue contours lose the real leaf borders near the tip of the leaves. Clearly, a general set of weighting parameter values is difficult to find. Even when such "optimal" parameter combination could be found for one image, it is unlikely that it would work for all the images in the sequence.

We now show results for our proposed method to illustrate that it does not suffer from the parameter selection, nor does it suffer from the linear influence that edge strength has. In order to use the proposed method, the internal and external probabilities have to be modelled first. The prior probabilities used for our method were learned using a single ground truth image. This image originated from a training time lapse sequence that was manually segmented. The distributions were calculated using the kernel density estimator with a normal distribution as kernel. In Fig.1 (e) the external probability at each pixel is shown. The gradient strength at leaf borders varies between leaves, nevertheless

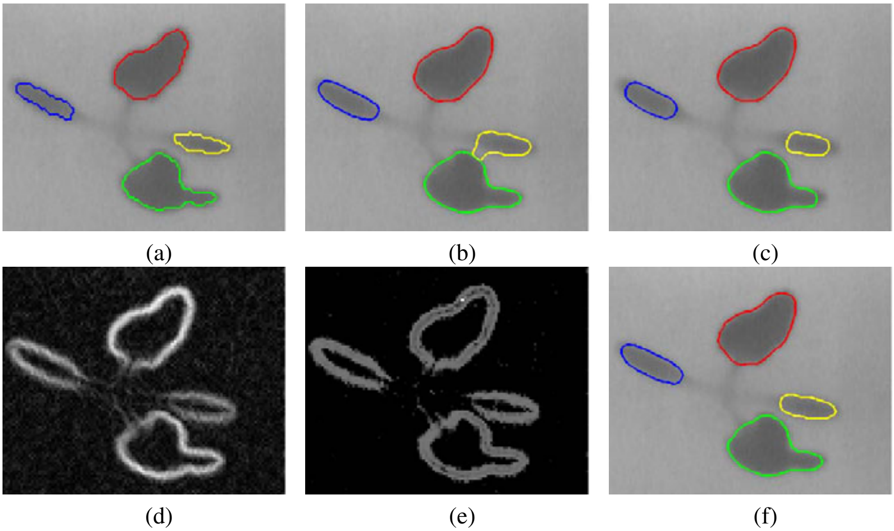


Fig. 1. Examples of segmentation using (probabilistic) active contours. The top row contains in (a) the initialization used for the (probabilistic) active contours, (b) the segmentation result of the classical active contours with $\alpha = \frac{1}{6}, \beta = \frac{1}{3}$, (c) the segmentation result of the classical active contours with $\alpha = 1, \beta = \frac{2}{3}$. The bottom row: (d) the gradient strength of the image, (e) the external probability of each pixel in the image, (f) the segmentation result of the proposed probabilistic active contours.

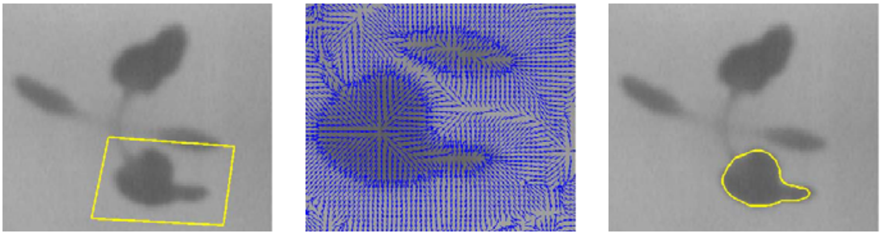
show the different leaves an equally strong probability at their border. This results in a better segmentation result as can be seen in Figure (f).

The previous example started from an almost perfect initialization. This was helpful to illustrate the problems that might occur with classical active contours, but it is rare that such a good initial contour is available. A more realistic example is shown in Fig. 2. The initialization is shown in Figure (a). In Figure (b) a detailed view of the GVF force field is shown. The force field points towards both leaves, as can be expected. The regularization effect of the internal probability however enforces the contour to converge to the correct leaf as is shown in Figure (c).

To quantitatively validate the proposed method, in 56 images have been manually segmented, each containing 4 leaves. The initialization of the active contours was based on these ground truth segments: the segments were dilated using a circular structuring element of size 5, the borders of these dilated segments were then used as initialization. As a validation metric the Dice coefficient is used: consider S the resulting segment from the active contour, i.e. the region enclosed by $\mathbf{r}(s)$, and GT the ground truth segment, then the Dice coefficient between S and GT is defined as:

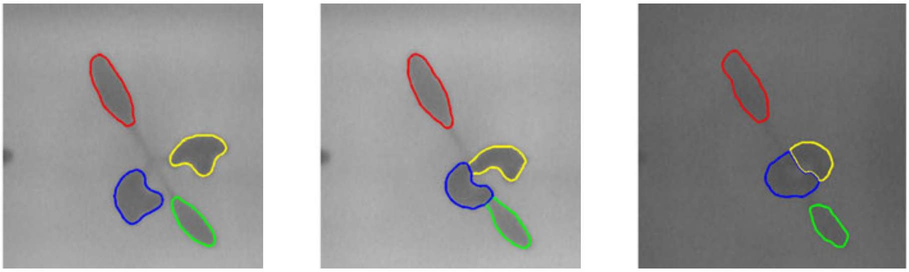
$$d(S, GT) = \frac{2\ \text{Area}(S \wedge GT)}{\text{Area(S)} + \text{Area}(GT)} \tag{15}$$

Here $S \wedge GT$ consist of all pixels which both belong to the detected segment as well as to the ground truth segment. If S and GT are equal, the Dice coefficient is equal to one. The



(a) (b) (c)

Fig. 2. Examples of segmentation using probabilistic active contours. (a) the initialization used for the (probabilistic) active contours, (b) the gradient vector flow of the external probability, i.e. the force used to optimize the external probability of the active contour (c) the segmentation result of the proposed probabilistic active contours.



(a) frame 1 (b) Frame 7 (c) Frame 15

Fig. 3. Example of tracking using probabilistic active contours

Dice coefficient will approach zero if the regions hardly overlap. In order to compare our method with the active contours with the most optimal parameter setting, the image sequence has been segmented with α and β both in the range of $0, \frac{1}{30}, \frac{2}{30}, \frac{3}{30}, ..., 1$. The best combination of α and β resulted in an average Dice coefficient of 0.872. Using these optimal parameters, 24 segments resulted in a Dice coefficient equal to 0, i.e. the segments where completely lost. The proposed probabilistic active contours resulted in an average Dice coefficient of 0.929 with no leaves lost.

As a last example our proposed method was applied for leaf tracking in a time lapse sequence. Since the movement of the leaves does not seem to have a clear motion model, we cannot incorporate prior knowledge about the motion in our tracking methods such as in [15,1]. Instead the result of frame t will be used as an initialization for frame $t+1$, such as done by Tsechpenakis et al. [16]. As can be seen in n Fig. 3 does the proposed method cope with the movement and deformation of the leaves. Even frame 15 where the illumination level diminished due to nightfall, is still segmented correctly. If this illumination change is too strong, the learned probability distributions will not correspond to the image features. Therefore the segmentation results will be less accurate. This can already be seen at Figure (c), where the contours delineating the bigger leaves miss the true border at the center of the plant. Although this error is almost unnoticeable at this frame, there's a risk that it becomes more prominent in subsequent frames which will use these contours as an initialization.

5 Discussion and Conclusion

In this paper a new variant on the active contour framework is defined. Instead of optimizing an energy function it strives to maximize the probability that the contour is on the edge of an object. The proposed method does not need to tune a set of weighting parameters, since it is based on probability theory. This approach however needs a good estimate of the probability distribution functions that are needed for the calculation of the internal and external probability. These probability distributions can be learned from a ground truth training set. This method has been tested for the segmentation and tracking of sugar beet seedling leaves in thermal time lapse sequences. In these tests the proposed technique has been shown to be useful and outperformed classical active contours for the segmentation of multiple objects. To cope with changing light conditions, the learned probability distributions should be updated in order to follow the illumination settings of the image. This could be done using methods similar to background maintenance techniques [17]. We intend to investigate the influence of these methods in future research.

Acknowledgment. The authors are grateful to Erik De Bruyne, SESVanderhave NV, for providing the plant material. Jonas De Vylder is funded by the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT).

References

1. Isard, M., Blake, A.: Active contours. Springer, Heidelberg (1998)
2. Liu, Y.: Automatic 3d form shape matching using the graduated assignement algorithm. Pattern Recognition 38, 1615–1631 (2005)
3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. International Journal of Computer Vision, 321–331 (1988)
4. Tsechpenakis, G., Rapantzikos, K., Tsapatsoulis, N., Kollias, S.: A snake model for object tracking in natural sequences. Signal Processing-Image Communication 19(3), 219–238 (2004)
5. Charmi, M.A., Derrode, S., Ghorbel, F.: Fourier-based geometric shape prior for snakes. Pattern Recognition Letters 29(7), 897–904 (2008)
6. Chan, T., Vese, L.: An active contour model without edges. Scale-Space Theories in Computer Vision 1682, 141–151 (1999)
7. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. Ieee Transactions on Image Processing 10(10), 1467–1475 (2001)
8. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. Siam Journal on Applied Mathematics 66(5), 1632–1648 (2006)
9. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging and Vision 28(2), 151–167 (2007)
10. Charmi, M.A., Derrode, S., Ghorbel, S.: Fourier-based geometric shape prior for snakes. Pattern Recognition Letters 29, 897–904 (2008)
11. Staib, L., Duncan, J.: Boundary finding with parametrcally deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 1061–1075 (1992)

12. Xu, C., Prince, J.: Snakes, shapes and gradient vector flow. IEEE Transactions on Image Processing 7, 359–369 (1998)
13. Xu, C., Yezzi, A., Prince, J.: On the Relationship between Parametric and Geometric Active Contours. In: Proc. of 34th Asilomar Conference on Signals, Systems, and Computers, vol. 34, pp. 483–489 (October 2000)
14. Poon, C.S., Braun, M.: Image segmentation by a deformable contour model incorporating region analysis. Physics in Medicine and Biology 42, 1833–1841 (1997)
15. Goobic, A., Welser, M., Acton, S., Ley, K.: Biomedical application of target tracking in clutter. In: Proc. 35th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 88–92 (2001)
16. Tsechpenakis, G., Rapantizikos, K., Tsapatsoulis, N., Kollias, S.: A snake model for object tracking in natural sequences. Signal Processing: Image Communication 19, 219–238 (2004)
17. Shireen, Khaled, Sumaya: Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. Recent Patents on Computer Science 1, 32–34 (2008)

Gallbladder Segmentation from 2–D Ultrasound Images Using Active Contour Models and Gradient Vector Flow

Marcin Ciecholewski

Institute of Computer Science, Jagiellonian University,
ul. Łojasiewicza 6, 30-348 Kraków, Poland
marcin.ciecholewski@ii.uj.edu.pl

Abstract. Extracting the shape of the gallbladder from an ultrasonography (USG) image is an important step in software supporting medical diagnostics, as it allows superfluous information which is immaterial in the diagnostic process to be eliminated. In this project, several active contour models were used to segment the shape of the gallbladder, both for cases free of lesions, and for those showing specific disease units, namely: lithiasis, polyps, and anatomical changes, such as folds of the gallbladder. The approximate edge of the gallbladder is found by applying one of the active contour models like the membrane and motion equation as well as the gradient vector flow model (GVF-snake). Then, the fragment of the image located outside the identified gallbladder contour is eliminated from the image. The tests carried out showed that the average value of the Dice similarity coefficient for the three active contour models applied reached 81.8%.

1 Introduction

The rapid development of medical information technology contributes a lot to a significant improvement in the visualisation and diagnostic capabilities of computer systems supporting diagnostic processes in medicine. These systems have become important tools helping physicians with difficult diagnostic tasks. Medical image analysis supports not only recognising human internal organs, but also identifying lesions occurring in them. However, for some important organs like the gallbladder there are no ready, practical solutions to help physicians in their work.

The job of extracting the gallbladder structure from USG images is a difficult process because images have uneven backgrounds, as shown in Fig. 1. In addition, there is a large variety of gallbladder shapes in USG images due to individual traits of patients, among other reasons. USG images can also present such diseases as lithiasis, polyps, changes of the organ shape like folds, turns and others which hinder extracting the contour.

In general, literature includes many publications about extracting shapes of organs from USG images. One group of algorithms are these that detect edges

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 86–97, 2011.

in the image [1, 3]. Edges are usually located in areas with a high gradient value on the image, where the values of the grey level clearly change, e.g. from black to white. Edge algorithms yield inexact results when detecting an edge that is dotted and unclear. They are also computationally complex and leave noise which needs to be eliminated later. Another solution is offered by algorithms based on textures. Richard and Keen [11] have developed an algorithm designed for detecting edges in USG images using the classification of pixels corresponding to specific characteristics of textures. Although the algorithm is fully automatic, the authors note that it is computationally complex. The computational complexity of methods based on texture analysis is usually equal to $O(n^4) : W \times H \times r^2$ where: W is the image width, H is its height, and r denotes the length of the ROI side.

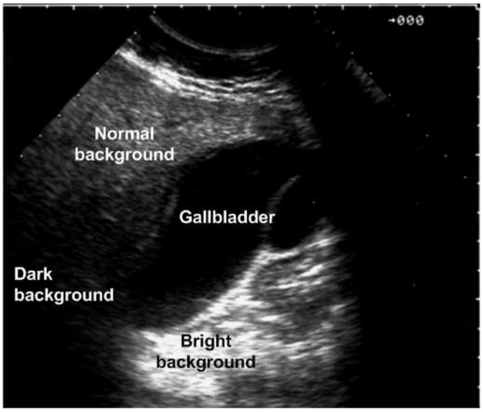


Fig. 1. An example US image of the gallbladder

Algorithms based on deformable models like 2D AAM (the active appearance model) and the active contour (ACM) yield very exact results with relatively low calculation [12, 13]. They are usually semi-automatic methods where the initial contour or the average shape model is initiated by the user. AAM models contain information about the average shape of an object, e.g. the lumbar section of the spine on a digital x-ray image [12] and data describing the most characteristic modifications of this shape observed in the training set. The form of the model may be modified by algorithms which try to fit it to the actual shape while not allowing unnatural deformations to appear. The active contour is a mathematical model of a deformable curve located within a two-dimensional environment of an external field created by the local characteristics of the image. The fitting of the model to the shape of the object is an iterative process just as in the case of AAM. Active contour models have been used for US images to determine the shape of such organs as: the carotid artery [7] and the liver [5]. However, they have not yet been used to support the USG diagnostics of the gallbladder. In this publication, the following active contour models have been used to determine the approximate area of the gallbladder: the membrane and motion equation as well as the gradient vector flow model. Research covered 600 cases from different patients, including USG images without lesions and ones showing lesions like: lithiasis, polyps and changes in the shape of the organ, namely folds of the gallbladder. This article is structured as follows. Section 2 presents methods for detecting the gallbladder contour in USG images. Section 3 describes the method of extracting the shape of the gallbladder from USG images. Section 4 discusses the experiments conducted and the research results. The last section contains a summary and sets out directions of future research.

2 Delineating the Contour in a USG Gallbladder Image

This chapter presents a method of determining the approximate contour of the gallbladder in USG images. The first step towards determining the approximate gallbladder contour in USG images is the normalisation transformation, which makes it possible to improve the contrast of images if the values of image brightness do not cover the entire range of possible values. The histogram normalisation transformation is an elementary operation in digital image processing [6]. The next step is to determine the approximate contour of the gallbladder by applying one of the active contour models like the membrane equation and the motion equation as well as the gradient vector flow model.

2.1 Active Contour Method

In a 2D image analysis context, an active contour is a flat curve which can change its shape dynamically and fit itself to image elements such as edges or borders. The concept of contour shape formation for matching image edges is explained in Fig. 2. The objective of contour movements is to find the best fit, in terms of some cost function, as a trade-off between the contour curvature and the boundary of the image object under analysis. In [8] the potential energy function of the active contour has been proposed to play the role of this cost function. The energy function is given by the following integral equation:

$$E_S = \int_0^{S_{m-1}} [E_i(v(s)) + E_e(v(s)) + E_p(v(s))]ds \tag{1}$$

where the parametric equation $v(s) = (x(s), y(s))$ defines the position of the curve, E_i represents the internal potential energy of the contour, E_e is the energy which models external constraints imposed onto the contour shape, and E_p represents component energies derived from image features, e.g. the image brightness distribution. The notation of the energy function in the discrete format is more convenient in the computer implementation of deformable models:

$$E_S = \sum_{s=0}^{S_{m-1}} [E_i(v(s)) + E_e(v(s)) + E_p(v(s))] \tag{2}$$

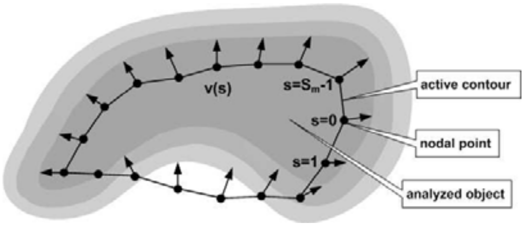


Fig. 2. Building a model of the active contour method. Arrows represent the directions in which nodal points move towards the edge of the analyzed object.

In this case, the energy equation is interpreted as the total of component energies of all nodal points. The symbol s symbol is the index identifying the nodal point.

2.2 Membrane Model

The membrane model used in this project had been proposed in publications [8, 10]. The value of energy E_i expressed in relationship (2) is presented by the following membrane equation:

$$E_i(v(s)) = \tau\left|\frac{dv(s)}{ds}\right|^2 + \rho\left|\frac{d^2v(s)}{ds^2}\right|^2 \tag{3}$$

The values τ and ρ influence the elasticity of the model and represent extensibility and flexibility, respectively. In the case of the active contour, the discrete version of the E_i equation can have the following form:

$$E_i(v(s)) = \tau[v(s+1) - v(s)]^2 + \rho[v(s+1) - 2v(s) + v(s-1)]^2 \tag{4}$$

The value of energy E_e can be defined as:

$$E_e = \beta_1\frac{|v(s) - v_0|^3}{3} + \beta_2 ln|v(s) - v_0| \tag{5}$$

The value of the energy given by equation (5) depends only on the distance between the nodal point and the v_0 point. This means that the vector of its gradient in a given point always has the same direction as the line running through that point and the field source point. The β_1 and β_2 parameters make it possible to select the degree to which the force acts on the nodal points and the distance from the point v_0, at which the minimum of its corresponding energy E_e occurs. The value of the energy E_p with which the image acts can be presented by the following equation: $E_p = -\xi\sqrt{g_x^2 + g_y^2}$ where ξ is a parameter while g_x and g_y are gradients of grey levels of the image along the Ox and Oy directions, respectively. Directional gradients can be calculated using the following equations:

$$\begin{aligned} g_x(x, y) &= g(x+1, y) - g(x-1, y) \\ g_y(x, y) &= g(x, y+1) - g(x, y-1) \end{aligned} \tag{6}$$

where $g(x, y)$ is the value of the grey level at the coordinates (x, y) .

Publication [8] proposes a method for minimizing the active contour functional using Euler equations and the finite difference method. For functional (2), the Euler equations for the coordinates along, respectively, the Ox and Oy, directions can be written in the matrix form:

$$\begin{aligned} A_x + f_x(x,y) &= 0 \\ A_y + f_y(x,y) &= 0 \end{aligned} \tag{7}$$

Matrix A is a pentadiagonal matrix mapping the elasticity of the model, vectors of functions f_x and f_y correspond to components E_p and E_e of equation (2), while the x and y vectors determine the coordinates of individual nodal points. Equations (7) can be solved iteratively. To obtain the appropriate equations in the iterative form [8], the left sides of these equations are equated with the negated derivatives of vectors x and y in relation to time. Then the time discretization is introduced and the locations of nodes in iteration t are determined based on the values calculated in the previous iteration. The iteration formulas obtained have the following form:

$$\begin{aligned} x_t &= (A + \eta 1)^{-1}(x_{t-1} - f_x(x_{t-1}, y_{t-1})) \\ y_t &= (A + \eta 1)^{-1}(y_{t-1} - f_y(x_{t-1}, y_{t-1})) \end{aligned} \tag{8}$$

where η is the value of the iteration step. The problem in equations (8) is that reversing the pentadiagonal matrix yields a matrix in which all elements are not zero. The number of addition and multiplication operations executed in a single iteration is therefore high and grows proportionally to the square of the matrix dimension, i.e. proportionally to the squared number of nodes in the model.

2.3 Motion Equation Model

The next model used in this project and proposed in article [9] is the application of a specific physical interpretation of the deformable model. This model is treated here as a flexible object of a specific mass moving within an environment of a defined viscosity. Energy E_S is minimized by changing it into the kinetic energy of moving masses of nodal points, subsequently lost as a result of moving within a viscous environment. To model the shifts of individual nodal points, a motion equation of the following form is used:

$$m\frac{\delta^2 v(s,t)}{\delta t^2} + l\frac{\delta v(s,t)}{\delta t} = F(s,t) \tag{9}$$

$$F(s) = -\nabla E_S(s) \tag{10}$$

where $v(s,t)$ is the vector of the nodal point coordinates, m is the mass assigned to every node of the graph, l is the viscosity coefficient of the environment, and F is the vector representing all forces acting on the nodes of the structure. The force F for a single nodal point can be determined as the negated value of the gradient of energy E_S calculated in the image (10). The use of the motion equation (9) to describe contour dynamics makes it possible to quickly determine the contour balance state and does not require determining the total minimum value of energy E_S shown by equation (2). In the computer implementation, equation (9) is presented in the discrete form of:

$$m[v(s,t)-2v(s,t-1)+v(s,t-2)]+l[v(s,t)-v(s,t-1)]=F(s,t-1) \tag{11}$$

After determining the location of the nodal point at the moment t, we obtain a formula allowing the location of nodal point at the time t to be calculated iteratively based on the values of forces F and their location in the previous two iterations. We obtain:

$$v(s,t)=\frac{F(s,t-1)+m(2v(s,t-1)-v(s,t-2))+lv(s,t-1)}{m+l} \tag{12}$$

The numerical convergence and stability of equation (12) depends on the values of parameters m and l, as well as on the way in which force F has been defined. In the case of deformable models, the value of this force depends on many factors, including the features of the analyzed image. The energy minimization method coupled with the motion equation makes it possible to subsequently, in individual iterations, change the location of individual nodal points or of all points at the same time. In the first case, the order of node location modification can be random or defined. If the location of all nodes is modified in the same iteration, equation (12) can be written in the matrix form. The use of the same notation as for equations (8) generates equations in the following form:

$$\begin{aligned} x_t &= \frac{Ax_{t-1}+f_x(x_{t-1},y_{t-1})+m(2x_{t-1}-x_{t-2})+lx_{t-1}}{m+l} \\ y_t &= \frac{Ay_{t-1}+f_y(x_{t-1},y_{t-1})+m(2y_{t-1}-y_{t-2})+ly_{t-1}}{m+l} \end{aligned} \tag{13}$$

The use of equations (13) requires fewer addition and multiplication operations than of equations (8). In the case of the active contour, matrix A is a pentadiagonal one. For other models, it is a sparse matrix in which the number of elements per row is constant. Consequently, the number of operations increases linearly along with the increasing number of nodal points, and not with the square of their number. This is why this method is more convenient for models with a large number of nodal points.

2.4 Gradient Vector Flow Snake

The gradient vector flow snake (GVFs) used in this project had been proposed in publications [14, 15]. The GVFs is an active contour model that minimises energy function (2) by satisfying the Euler equation

$$\alpha v''(s,t)-\beta v''''(s,t)+\vec{v}=0 \tag{14}$$

This can be viewed as a force balance equation

$$F_{int} + F_{ext}^{(g)} = 0 \tag{15}$$

where $F_{int} = \alpha v''(s,t) - \beta v''''(s,t)$ and $F_{ext}^{(g)} = \vec{v}$. The internal force F_{int} discourages snake to stretching and bending. The α and β prameters control snake's tension and ridgity, respectively, and $v'(s,t)$ and $v''''(s,t)$ are the 2nd and 4th order partial derivatives of $v(s,t)$ with respect to paramter s. The force $F_{ext}^{(g)} = \vec{v}(x,y)$ is called the gradient vector flow (GVF) field. The gradient vector flow field is defined as the vector of the field $\vec{v}(x,y) = [u(x,y), \bar{u}(x,y)]$ which minimises the energy functional

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + \bar{u}_x^2 + \bar{u}_y^2) + |\nabla g|^2 |\vec{v} - \nabla g|^2 dxdy \tag{16}$$

whereas u_x, u_y, $\bar{u}_x$ and $\bar{u}_y$ are partial derivatives of the appropriate functions, while μ is the regularization parameter. The μ parameter should be set depending on the level of noise in the image (the more noisy the image the higher should the μ be raised). The ∇g parameter is the image gradient. In particular, it can be said that if ∇g is low, the energy equation is dominated by the sum of squared partial derivatives of the vector field, which yields an only slightly varied field. On the other hand, if ∇g is high, the second expression found in the sum dominates the integral equation, which is then minimised by setting the value of $\vec{v} = \nabla g$. This produces the desired effect of a gradual change of the field in uniform areas (where the $g(x,y)$ value is constant). The GVF field functional can be determined using the following Euler equations:

$$\begin{aligned} \mu\nabla^2 u - (u - g_x)(g_x^2 + g_y^2) &= 0 \\ \mu\nabla^2 \bar{u} - (\bar{u} - g_y)(g_x^2 + g_y^2) &= 0 \end{aligned} \tag{17}$$

where ∇^2 is the Laplacian operator. The equation (17) can be presented in their discreet form:

$$\begin{aligned} u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - b(x,y)u(x,y,t) + c^1(x,y) \\ \bar{u}_t(x,y,t) &= \mu\nabla^2 \bar{u}(x,y,t) - b(x,y)\bar{u}(x,y,t) + c^2(x,y) \end{aligned} \tag{18}$$

where

$$\begin{aligned} b(x,y) &= g_x(x,y)^2 + g_y(x,y)^2 \\ c^1(x,y) &= b(x,y)g_x(x,y) \\ c^2(x,y) &= b(x,y)g_y(x,y) \end{aligned}$$

The coefficients $b(x,y)$, $c^1(x,y)$ and $c^2(x,y)$ may be calculated once and fixed for entire iterative process. To define the entire iterative solution, it was assumed that indices i, j and n refer, respectively, to variables x, y, t. Let the distance between pixels along the Ox and Oy axes be, respectively, Δx and Δy, and the time step for each iteration be expressed as Δt. Then, the required partial derivatives can be approximated [14]. If these approximations are substituted in equations (18), this produces the following iterative GVF solution:

$$\begin{aligned} u_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)u_{i,j}^{n} + r(u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + \\ &\quad + u_{i,j-1}^{n} - 4u_{i,j}^{n}) + c_{i,j}^{1}\Delta t \\ \bar{u}_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)\bar{u}_{i,j}^{n} + r(\bar{u}_{i+1,j}^{n} + \bar{u}_{i,j+1}^{n} + \bar{u}_{i-1,j}^{n} + \\ &\quad + \bar{u}_{i,j-1}^{n} - 4\bar{u}_{i,j}^{n}) + c_{i,j}^{1}\Delta t \end{aligned} \tag{19}$$

where

$$r = \frac{\mu \Delta t}{\Delta x \Delta y} \tag{20}$$

Assuming that coefficients b, c^1 and c^2 are bounded, then equation (19) is stable if the Courant–Friedrichs–Lewy [2] step size of $r \leq 1/4$ is preserved. Having Δx, Δy and μ values fixed and taking definition r from (20), we can estimate the restriction of the Δt time-step which must be kept to ensure the GVF convergence:

$$\Delta t \leq \frac{\Delta x \Delta y}{4\mu} \tag{21}$$

The convergence expressed in (21) can be determined faster for coarse images, e.g. when Δx and Δy are high. If μ is high and the GVF field is expected to be smooth, the convergence in (19) will be slow (as the Δt must be low).

Figure 3 (b) shows an example with GVF external forces obtained from the USG image presented in figure 3(a).

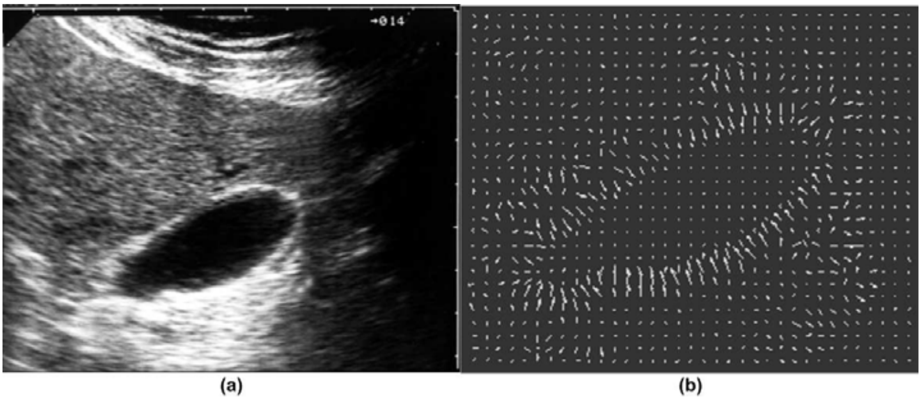


Fig. 3. Finding GVF external forces. (a) Sample USG image of the gallbladder. (b) GVF external forces, zoom in of the area containing the gallbladder shape.

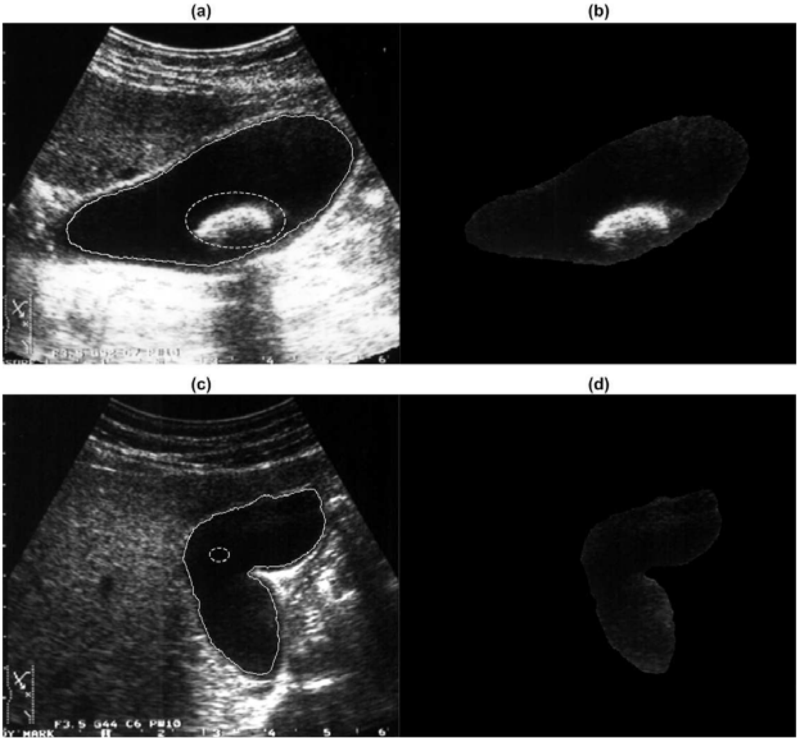


Fig. 4. The gallbladder segmentation in USG images using active contour methods. The dashed line shows manually initiated contour inside the gallbladder shape. (a), (b) An image with visible cholecystolithiasis. (c), (d) A gallbladder fold.

3 Gallbladder Segmentation in a US Image

The proposed method for extracting the gallbladder shape in USG images makes use of the calculated values of coordinates identifying the gallbladder contour determined using one of the active contour models presented in sections 2.2–2.4. The contour is initiated manually inside the gallbladder shape. In order to extract the organ from the image, we have defined two areas identifying image fragments: GB - the area inside the gallbladder contour and BG - the area constituting the image background. Under these assumptions, the segmentation is executed in such a way that in the USG image showing the gallbladder and defined by the mapping $g : M^2 \to Z$, its fragment is replaced with the BG area in which all pixels are set to black in colour. We obtain:

$$g' = \begin{cases} g & \text{if } (x, y) \in GB \\ 0 \text{ (black)} & \text{if } (x, y) \in BG \end{cases} \tag{22}$$

Figures 4(a) and 4(c) show images with the gallbladder contour marked. Figures 4(b) and 4(d) contain the US images with the segmented shape of the gallbladder. Figures 4(a) and 4(b) show images with with lithiasis, while figures 4(c) and 4(d) a fold of the gallbladder.

Table 1. Test results for three measurements based on 600 USG images of the gallbladder, where the following active contour models were applied: (ME) membrane equation (MO) motion equation (GVFs) gradient vector flow snake. Mean Test Results – coming from the three measurements.

Patient	**No. of images**	**ME %**	**MO %**	**GVFs %**
No lesions	300	85.2%	88.2%	89.1%
Lithiasis	110	77.3%	76.5%	77.8%
Polyp	90	79.4%	79%	80.3%
Fold/Turn	100	82.5%	83%	84.2%
Total	600	81.1%	81.6%	82.8%
Measurement 1				
No lesions	300	84%	85.2%	86%
Lithiasis	110	75.1%	75.6%	76.5%
Polyp	90	79.7%	79%	80.3%
Fold/Turn	100	81.2%	82%	83.2%
Total	600	80%	80.4%	81.5%
Measurement 2				
No lesions	300	87.3%	88.1%	89%
Lithiasis	110	79.2%	78.3%	80%
Polyp	90	81.2%	81%	81.5%
Fold/Turn	100	83.7%	84.2%	85.3%
Total	600	82.8%	82.9%	83.9%
Measurement 3				
No lesions	300	85.5%	87.1%	88%
Lithiasis	110	77.2%	76.8%	78.1%
Polyp	90	80.1%	79.6%	80.7%
Fold/Turn	100	82.4%	83%	84.2%
Total	600	81.3%	81.6%	82.7%
Mean Test Results				

4 Completed Experiments and Selected Research Results

In order to estimate the precision of models used to determine the approximate contour of the gallbladder, the Dice's similarity coefficient was used. Images from the Department of Image Diagnostics of the Regional Specialist Hospital in Gdańsk, Poland, were used in the research on USG image analysis. Dice's similarity coefficient is a value making it possible to compare the percentage similarity of sets. These sets can be defined as areas with defined pixel numbers in the analysed digital image. It was assumed that $|Lv_{accon}|$ is the number of pixels found in the area delineated using the active contour method, while $|Lv_{manual}|$ is the number of pixels in the area isolated by the radiologist. The number of pixels found in the common area is $|Lv_{accon} \cap Lv_{manual}|$. Dice's similarity coefficient is defined as follows:

$$s = \frac{2 \cdot |Lv_{accon} \cap Lv_{manual}|}{|Lv_{accon}| + |Lv_{manual}|} \times 100\% \tag{23}$$

Table 1 shows the results of three experiments based on measurements taken by three different physicians specialising in radiology. Table 1 shows experimental results for three active contour models used: the membrane equation (ME) and the motion equation (MO), the gradient vector flow snake (GVFs). Results of experiments for particular disease units are listed in the order of the number of cases. Data presented in Table 1 indicates that the results obtained using the three active contour models are comparable. Table 1 also shows the mean values based on all measurements taken by three different physicians. The mean value of Dice's similarity coefficient based on Tab. 1 for the 600 tested USG images of the gallbladder amounted to: 81.3% for the membrane equation (ME) and 81.6% for the motion model (MO), 82.7% for the gradient vector flow model (GVFs). The best (but not by far) results were produced using the gradient vector flow model. The mean Dice's coefficient for the three models used equals 81.8%.

5 Summary and Further Research Directions

This article presents a method of extracting the shape of the gallbladder from US images developed for a computer system supporting the early diagnostics of gallbladder lesions. First, the histogram normalisation transformation was executed allowing the contrast of USG images to be improved. The approximate edge of the gallbladder is determined by applying one of the active contour models like the membrane equation and the motion equation as well as the gradient vector flow model. The contour is initiated manually inside the gallbladder shape. The fragment of the image located outside the gallbladder contour is eliminated from the image. The active contour method with the applied models yielded precise results for both healthy organs and those showing specific disease units, namely: lithiasis, polyps, folds and turns of the gallbladder. For the 600 USG images, the mean Dice's similarity coefficient for the three active contour models applied was equal to 81.8%. Further research will be aimed at reducing the error for images showing such lesions as lithiasis and polyps, if they are located close to the gallbladder edge. Currently, research is also conducted to identify lesions using the AdaBoost (Adaptive Boosting) and SVM (Support Vector Machines) methods in processed USG images of the gallbladder after the uneven background of the image is eliminated using the method presented here. In both of the above machine learning methods which are now at the experimental stage, if the image background is uniform the process of learning and classyfing features (i.e. lesions) are more efficient.

Acknowledgements. This research was financed with state budget funds for science for 2009-2012 as research project of the Ministry of Science and Higher Education: N N519 406837.

References

1. Aarnink, R.G., Pathak, S.D., de la Rosette, J.J., Debruyne, F.M., Kim, Y., et al.: Edge detection in prostatic ultrasound images using integrated edge maps. Ultrasonics 36, 635–642 (1998)
2. Ames, W.F.: Numerical Methods for Partial Differential Equations, 3rd edn. Academic, New York (1992)
3. Bodzioch, S.: Information reduction in digital image and its influence on the improvement of recognition process. Automatics, Semi-annual Journal of the AGH University of Science an Technology 8(2), 137–150 (2004)
4. Ciecholewski, M.: Gallbladder Segmentation in 2-D Ultrasound Images Using Deformable Contour Methods. In: Torra, V., Narukawa, Y., Daumas, M. (eds.) MDAI 2010. LNCS, vol. 6408, pp. 163–174. Springer, Heidelberg (2010)
5. Cvancarova, M., Albregtsen, T.F., Brabrand, K., Samset, E.: Segmentation of ultrasound images of liver tumors applying snake algorithms and GVF. International Congress Series (ICS), pp. 218–223 (2005)
6. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, Englewood Cliffs (2008)
7. Hamou, A.K., Osman, S., El-Sakka, M.R.: Carotid Ultrasound Segmentation Using DP Active Contours. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 961–971. Springer, Heidelberg (2007)
8. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. International Journal of Computer Vision 1(4), 321–331 (1988)
9. Leymarie, F., Levine, M.D.: Simulating the Grassfire Transform using an Active Contour Model. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(1), 56–75 (1992)
10. Neuenschwander, W., Fua, P., Kuebler, O.: From Ziplock Snakes to Velcro Surfaces. In: Automatic Extraction of Man Made Objects from Aerial and Space Images, pp. 105–114. Birkhaeuser Verlag Basel, Monte Verita (1995)
11. Richard, W.D., Keen, C.G.: Automated texture-based segmentation of ultrasound images of the prostate. Comput. Med. Imaging Graph 20(3), 131–140 (1996)
12. Roberts, M.G., Cootes, T.F., Adams, J.E.: Automatic segmentation of lumbar vertebrae on digitised radiographs using linked active appearance models. Proc. Medical Image Understanding and Analysis 2, 120–124 (2006)
13. Szczypiński, P., Strumiłło, P.: Application of an Active Contour Model for Extraction of Fuzzy and Broken Image Edges. Machine GRAPHICS & VISION 5(4), 579–594 (1996)
14. Xu, C., Prince, J.L.: Snakes, Shapes, and Gradient Vector Flow. IEEE Transactions on Image Processing 7(3), 359–369 (1998)
15. Xu, C., Prince, J.L.: Gradient vector flow: A new external force for snakes. In: IEEE Proc. Conf. on Computer Vision and Pattern Recognition, pp. 66–71 (1997)

Multi-view Alpha Matte for Free Viewpoint Rendering

Daniel Herrera C., Juho Kannala, and Janne Heikkilä

Machine Vision Group, University of Oulu, Finland
{dherrera,jkannala,jth}@ee.oulu.fi

Abstract. We present a multi-view alpha matting method that requires no user input and is able to deal with any arbitrary scene geometry through the use of depth maps. The algorithm uses multiple observations of the same point to construct constraints on the true foreground color and estimate its transparency. A novel free viewpoint rendering pipeline is also presented that takes advantage of the generated alpha maps to improve the quality of synthesized views over state-of-the-art methods. The results show a clear improvement on image quality by implicitly correcting depth map errors, providing more natural boundaries on transparent regions, and removing artifacts.

1 Introduction

Transparency in a scene is often desirable and usually unavoidable. It can be the result of hair, semi-transparent materials, motion blur, or even aliasing. It gives the objects in the scene a realistic look as boundaries in real scenes are usually not pixel sharp.

A very popular application that suffers greatly from transparency artifacts is free viewpoint rendering. The goal is to render new views by interpolating from nearby cameras [1]. New view synthesis is particularly useful for 3D TV. Mixed boundary pixels produce ghosting artifacts in the synthesized view that significantly reduce its quality.

This paper deals with estimating the transparency of objects in an image, also known as an alpha map, using multiple views of the scene. It is a challenging problem since the transparency can have different causes and the object boundaries are often complex (e.g. hair). We also address the application of the obtained alpha maps to the free viewpoint rendering problem to improve the quality of novel views.

1.1 Alpha Matting Background

The literature contains many single view alpha matte estimation algorithms, including a comprehensive online benchmark [2]. Due to the under constrained nature of the problem they all require user input to identify some pure foreground and background regions to be used as examples. Some accept a sparse labeling

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 98–109, 2011.

in the form of strokes, while others require dense labeling in the form of a tri-map. This is a considerable limitation since the need for user input limits the applicability of the algorithms, particularly for video sequences.

Single view alpha matting algorithms can be divided into three categories according to the assumptions made for an image: color sampling, alpha propagation, and optimization methods. Color sampling methods (e.g. shared sampling [3]) take samples from nearby labeled regions. They assume color smoothness to interpolate the alpha values between the labeled regions, usually requiring a dense tri-map. Alpha propagation methods assume that the alpha values are correlated to some local image statistics and use this to interpolate the alpha values (e.g. Closed form matting [4]). These methods often allow sparse user input. Optimization methods combine the previous two approaches to exploit their strengths (e.g. Robust matting [5]). Although very impressive results have been shown for single view alpha matting [2] it is expected that a multi-view approach would improve the existing methods since more information is available and several observations of the same point can be used.

Zitnick et al. presented a free viewpoint rendering system that estimates the alpha matte along depth discontinuities [6]. It uses a variant of Bayesian matting [7] to estimate colors and opacities for mixed boundary pixels. Although the stereo and rendering is multi-view, the matting is performed using a single view. Moreover, they assume a fixed width boundary which limits the applicability in scenes with large semi-transparent regions.

Hasinoff et al. [8] propose a method to estimate transparency at object boundaries using boundary curves in 3D space. They use a multi-view approach but limit theirselves to mixed boundary pixel transparency. Intrinsic material transparency is not addressed and objects are assumed to be opaque. Joshi et al. [9] suggest a multi-view variance measure to estimate transparency. The approach computes a tri-map and propagates color statistics. This imposes limitations on the color statistics of the scene. Moreover, it does not use all available information by using only the variance of the samples.

Wexler et al. [10] present a multi-view environment matting algorithm to estimate the light transfer function of a foreground object (e.g. a magnifying glass). They include alpha estimation in their algorithm but only handle planar backgrounds in their paper. Moreover, they assume an alpha value independent of viewpoint, which limits the algorithm to planar foregrounds as well. The most closely related work is that of Wexler et al. in [11]. They developed a multi-view approach to alpha matte estimation under a Bayesian framework. They show very good results but limit their model to planar layers. Moreover, their model has an alpha value independent of view, which is not suitable for mixed boundary pixels.

The goal of our matting stage is to generate a layered depth image [12] from each input camera. However, we focus on the construction of this LDI from real world images while estimating transparency. Even modern LDI approaches like [13] suffer from artifacts due to mixed pixel boundaries and transparency.

1.2 Free Viewpoint Rendering Background

A review of the latest free viewpoint rendering methods [1] shows that one of the dominant approaches is to calculate a depth map for each image and then warp the pixel colors to the new view using camera calibration information. The problem with this approach is that traditional depth maps have only a single depth per pixel and do not take transparency into account. This results in ghosting artifacts. Recent methods like Müller et al. [14] attempt to discard mixed pixels to remove the artifacts. Yet, this approach discards information, suffers from unnaturally sharp boundaries, and still produces artifacts for complicated semi-transparent regions.

Our approach also shares a strong similarity with Fitzgibbon et al. [15]. We use a similar scanning of the optical rays to find matching colors amongst the images. Our approach is novel in that it uses linear constraints on RGB space to estimate the true color of semi-transparent points while their approach ignores transparency issues.

2 Modeling Transparency

When the transparency and color of an object are unknown the observed color of a pixel can be the result of different situations. As mentioned in [10], if the background is known to be white and the pixel is a 50% combination of red and white, this can be due to any of the following:

1. Object is pink.
2. Object is red with a transparency of 50%.
3. Object is red but covers only 50% of the pixel (mixed boundary pixel).

Both transparency types are view dependent. The former because light rays will traverse different paths through the object, and the latter because a 3D point observed from a different view might not be a 2D boundary pixel any more.

Our model for a semi-transparent pixel p in image i is described by the following matting equation:

$$M_i = \alpha_i F + (1 - \alpha_i) B_i \tag{1}$$

The observed color M_i is a mixture of the foreground and background colors. It assumes a Lambertian surface, which results in a single foreground color F shared by all images. Yet, the background color B_i and alpha value α_i are view dependent. Because most of the work is done individually for each pixel, the index p is omitted.

3 Multi-view Alpha Estimation

Our algorithm requires the camera projection matrix and depth map for each input image. Using this information all pixels can be back-projected into 3D

Algorithm 1. Multi-view alpha algorithm

```
for all $i \in$ Images do                                                  ▷ Sample collection
    for all $p \in$ Image($i$) do
        object_cluster$_i(p) \leftarrow$ $FindCluster\ (p, depth_{min})$
        depth$_{obj} \leftarrow$ $depth$(object_cluster)
        background_cluster$_i(p) \leftarrow$ $FindCluster\ \left(p, \text{depth}_{obj} + \epsilon_f\right)$
        $B_i(p) \leftarrow$ $ref\_color$(background_cluster)
    end for
end for
for all $i \in$ Images do
    for all $p \in$ Image($i$) do
        sample_set $\leftarrow \emptyset$                                  ▷ Sample assembly
        for all pixel $\in$ object_cluster$_i(p)$ do
            $j \leftarrow$ pixel.image
            sample.M $\leftarrow$ pixel.color
            sample.B $\leftarrow B_j$(pixel)
            If sample is stable Then add to sample_set
        end for
        Project samples to main constraint                                 ▷ Alpha estimation
        $F_i(p) \leftarrow$ farthest color along RGB ray
        $\alpha_i^*(p) \leftarrow \frac{\|M_i(p)-B_i(p)\|}{\|F_i(p)-B_i(p)\|}$
    end for
end for
Minimize energy using graphcut                                             ▷ Alpha smoothing
```

world space and several observations of the same scene point can be grouped together. The main objective of the algorithm is to estimate B_i, F, and α_i for each pixel. Because we can obtain several samples for a scene point and its background, alpha estimation can be done pixel-wise and no tri-map or user input is needed.

Our method is summarized in Algorithm 1. It can be divided into four stages. First, color samples are collected for each pixel and its background. Second, the samples are assembled together into geometric constraints. Then, using these constraints the true color and alpha value are estimated. These first three stages treat each pixel individually. The final stage uses a graph cut minimization to enforce spatial smoothness in the alpha map. Each stage is described in detail in the following sections.

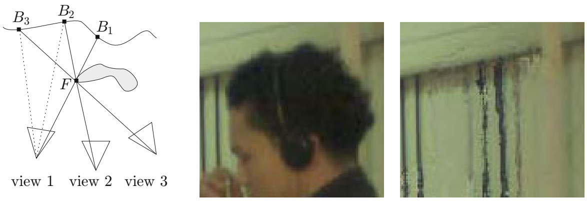
3.1 Sample Collection

Instead of using neighbor pixels from the same image, as is common in most alpha matting algorithms, our approach takes advantage of the fact that multi-view systems observe a point in the scene several times from different angles. Because of parallax the point is observed each time with a different background. The background itself can often be directly observed in a different image, as illustrated in Figure 1.

The colors observed for the same point are grouped together in clusters. Each cluster can have as many samples as there are cameras in the system. The algorithm scans the pixel's optical ray to find the first two distinct clusters in space. The first is denoted the foreground object cluster and contains the observed colors from all the views where the corresponding space point is visible. The second

Algorithm 2. Find color cluster algorithm

```
function FindCluster($pixel, depth_0$)
    $c^* \leftarrow \emptyset$                                              ▷ Best cluster found
    for $d = depth_0 ... depth_{max}$ do
        $c_d \leftarrow \emptyset$                                          ▷ Cluster at depth $d$
        $p_{ref}(x, y, z) \leftarrow$ back-project $pixel$ using $d$
        for $j \in Views$ do
            $p_j(u, v, w) \leftarrow$ project $p_{ref}$ to view $j$
            $w_{map} \leftarrow nearest\_neighbor(depth_j, u, v)$
            if $|w - w_{map}| \leq \epsilon_w$ then
                $rgb \leftarrow bilinear\_interpolation(j, u, v)$
                Add $rgb$ to $c_d$
            end if
        end for
        if $score(c_d) > score(c^*)$ then
            $c^* \leftarrow c_d$                                            ▷ New best cluster
        end if
        if $d - depth(c^*) \geq \epsilon_f$ then
            break                                 ▷ No cluster found for a while, end search
        end if
    end for
    return $c^*$
end function
```



(a) (b) Original (c) Background

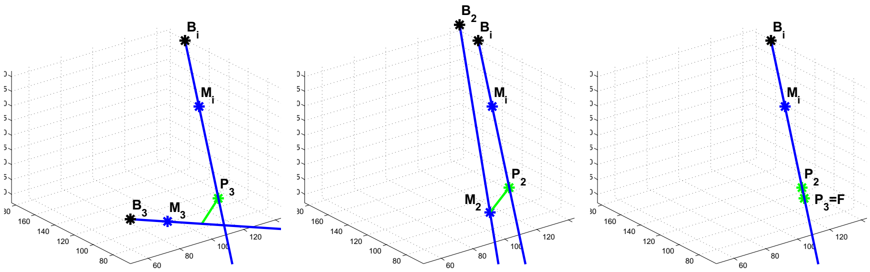
Fig. 1. Background recovery. (a) The background color for views 2 and 3 is directly observed by view 1. Image taken from [8]. (b) and (c) show an example of the recovered background.

is the background cluster and contains observed colors for the background. The method of collecting samples is detailed in Algorithm 2.

Each discretized depth d along the pixel's optical ray is projected onto the epipolar line of the other images. If the expected depth and the pixel's depth are similar, the pixel is added to a cluster at this position d. Due to noise and necessary tolerances, this procedure will obtain many similar clusters at nearby depths. These are essentially the same cluster at slightly different displacements. To select the best cluster for a point in space the candidate clusters are ranked according to the following formula:

$$score = \underset{M_k \in \text{cluster}}{median} \left(\|M_{\text{ref}} - M_k\| \right) \tag{2}$$

where the median is over all the samples in the cluster. The choice of reference color M_{ref} differs for the object and background clusters. For the object cluster it is the pixel color of the current view M_i. This creates a bias towards higher alpha values as it tries to find similar colors, but maximizes the chances of finding a



(a) Different background (b) Similar background (c) Final constraints

Fig. 2. Projection of samples onto main constraint. According to the final result in (c), P_3 is selected as $F_i(p)$ and M_i is assigned an alpha of 42%.

match with the same foreground color. For the background cluster, the sample M_j is selected whose camera j is closest to i because camera calibration and depth map noise have a smaller impact on nearby cameras.

The object cluster has the observed colors for this point $\{M_j | j \in$ [views where point is not occluded]$\}$. The background cluster is discarded and only the reference color is stored for the following stages. This reference color becomes the background color for the current pixel B_i. If only one cluster is found then no estimation is performed for this pixel (i.e. $F_i(p) = B_i(p) = M_i(p), \alpha_i^*(p) = 0$).

3.2 Sample Assembly

From (1) it can be seen that M_i lies on the line segment between B_i and F in RGB space. Because we do not know F we can use each sample to create a ray in RGB space that starts from B_i and passes through M_i. One ray is constructed for each entry in the object cluster. For an entry from image j, the background is obtained from the corresponding B_j.

3.3 Alpha Estimation

Since M_i is directly observable and B_i was estimated in the previous stage, the remaining task is to estimate F in Eq. (1). However, in order to facilitate new view synthesis, we would like to have an image-based representation for the color of the foreground objects (i.e. pre-rendered per source view) and hence we estimate the foreground layers $F_i(p)$ in a view dependant manner. To this effect, using the assembled rays in RGB space one can derive two types of constraints, as shown in Figure 2 and detailed in this section.

The first type of constraint is derived from the fact that all pixels belonging to the same foreground object cluster should share the same F, thus all rays should intersect at F (Fig. 2a). This is the underlying idea used for triangulation in standard blue screen matting [16]. However, rays originating from backgrounds with very similar color have very unstable intersection points, demonstrated in Figure 2b. In the case where the backgrounds are exactly the same color the rays are collinear.

The second type of constraint captures the idea that M_i lies in the line segment $\overline{B_iF}$. This means that F must lie on the $\overrightarrow{B_iM_i}$ ray at least as far as M_i. This gives an upper bound to the observed alpha values. This is specially useful for samples which are always observed with similar background colors and their intersection is therefore unreliable. If at least one image sees the true color of the point (i.e. $M_i = F$), which is a common case for mixed pixels at object boundaries, then we can still recover the true alpha value even if the background is non-textured.

To estimate $F_i(p)$ for a pixel p in image i, we first consider the ray defined by B_i and M_i to be the main ray. Each M_j from the foreground object cluster of the pixel p is then projected onto this ray in one of two ways, depending on the intersection angle between the rays:

$$P_j = \begin{cases} \Big((M_j - B_i) \cdot \hat{d}_i\Big)\, \hat{d}_i + B_i & \text{if } \angle ij \leq \epsilon_\angle \\ \text{Ray intersection} & \text{else} \end{cases} \tag{3}$$

where $\hat{d}_i$ is the ray direction from B_i to M_i.

If the angle between rays is lower than the threshold, the intersection is considered unreliable. Because this is caused by similar backgrounds, M_j can be directly projected onto the main ray (Fig. 2b). If the intersection angle is above the threshold, the point on $\overrightarrow{B_iM_i}$ closest to $\overrightarrow{B_jM_j}$ is used as the sample's projection P_j (Fig. 2a).

Once all samples have been projected onto the main ray (Fig. 2c), $F_i(p)$ is taken as the farthest P along the ray. The alpha value is then calculated as the distance of M_i to B_i relative to $F_i(p)$:

$$\alpha_i^*(p) = \frac{\|M_i(p) - B_i(p)\|}{\|F_i(p) - B_i(p)\|} \tag{4}$$

3.4 Alpha Smoothing

The previous stage estimates the foreground and background colors as well as the alpha value for each pixel. However, this is done independently for each pixel and is noisy. We can improve this estimate by taking spatial information into account. Since the alpha gradient directly contributes to the total gradient, we assume that regions with low color variation imply low alpha variation. This is exploited by applying a graph cut algorithm [17] to the obtained alpha values. The continuous interval $[0, 1]$ of alpha values is discretized into 100 labels with constant separation. The energy to be minimized is of the standard form:

$$E = \sum_{p \in I} E_d(p) + \lambda \sum_{p,q \in I} E_s(p, q) \tag{5}$$

where λ controls the weight of the spatial term relative to the data term.

The data term controls how much the new alpha deviates from the previous estimation. A truncated L1 norm is used as a robust cost measure:

$$E_d(p) = min\left(\left|\alpha(p) - \alpha^*(p)\right|, \epsilon_\alpha\right) \tag{6}$$

The spatial term penalizes variations in alpha value where the image gradient is low. However, if the depth of the recovered clusters differs, the spatial term is set to zero because the pixels belong to different objects.

$$E_s(p,q) = \begin{cases} \frac{min(|\alpha(p)-\alpha(q)|,\epsilon_\alpha)}{|\nabla M(p,q)|+1} & \text{if } |Z_i - Z_j| < \epsilon_z \\ 0 & \text{else} \end{cases} \tag{7}$$

3.5 Noise Considerations

There are three sources of noise for the algorithm: camera calibration parameters, depth map, and RGB noise. Each of these was analyzed to determine their impact in the estimation.

Camera calibration errors lead to an inaccurate optical ray for each pixel. The effect is directly visible in the plot of the epipolar line. We tested this effect in our datasets [6] using both the provided camera calibration and estimating the parameters using off-the-shelf structure from motion techniques. In both cases the epipolar line's inaccuracy was visibly less than half a pixel. We therefore assume sufficiently accurate calibration parameters.

The depth map on the other hand, presents considerable errors. Even though the quality of the depth map can be improved by using better stereo methods, it will still likely contain inaccuracies. This is taken into account in the sample collection stage. The clustering of the backprojected points and ranking of the clusters provides robustness against some depth map errors.

RGB noise has a stronger impact on pixels where the observed and background colors are similar. This can be measured by the length of each sample constraint (i.e. $\|M_j - B_j\|$). Constraints with a small length have an unstable direction and its projection is unreliable. Therefore, if the length is smaller than a given threshold the constraint is ignored. If the main constraint is to be ignored then no alpha value is calculated for the pixel (i.e. $F_i(p) = M_i(p), \alpha_i^*(p) = 1$).

4 Free Viewpoint Rendering

As an application for the obtained alpha maps, a free viewpoint rendering system was developed that handles transparent layers appropriately. The algorithm takes four layers as input: left background, left foreground, right background, and right foreground. Each layer has a depth map, an RGB texture, and an alpha map. The layer components are obtained directly from the output of the alpha estimation output for the left and right views. Areas where no alpha estimation could be performed have an empty foreground, with the original image color and depth used for the background layer. Left and right layers are merged to produce the final background color B_n, foreground color F_n, and alpha value α_n.

Each layer is first warped to the novel viewpoint independently. Small cracks that appear due to the forward warping are filled using the same crack-filling algorithm presented in [14]. Cracks are found by looking for depth values that are significantly larger than both neighboring values in horizontal, vertical, or diagonal directions. The median color of neighboring pixels is then used to fill in the cracks. Warped background layers are then combined pixel by pixel using a soft z threshold:

$$B_n = \begin{cases} \frac{d_l B_l + d_r B_r}{d_l + d_r} & \text{if } \left|Z_l^b - Z_r^b\right| < \epsilon_z \\ B_l & \text{else if } Z_l^b < Z_r^b \\ B_r & \text{else} \end{cases} \tag{8}$$

where d_l and d_r are the distances from the novel view's camera center to the left and right views' camera centers respectively. Merging of the foreground layers must take transparency into account. First the left and right foreground colors are combined. If both foreground pixels are close to each other, the final foreground color is interpolated between the two. If they are far apart, it is assumed that they represent different transparent layers and are thus combined using (1):

$$F_n = \begin{cases} \frac{d_l F_l + d_r F_r}{d_l + d_r} & \text{if } \left|Z_l^f - Z_r^f\right| < \epsilon_z \\ \alpha_l F_l + (1 - \alpha_l) F_r & \text{else if } Z_l^f < Z_r^f \\ \alpha_r F_r + (1 - \alpha_r) F_l & \text{else} \end{cases} \tag{9}$$

$$\alpha_n = \begin{cases} max\left(\alpha_l, \alpha_r\right) & \text{if } \left|Z_l^f - Z_r^f\right| < \epsilon_z \\ 1 - (1 - \alpha_l)(1 - \alpha_r) & \text{else} \end{cases} \tag{10}$$

Finally, (1) is applied to produce the final output color using F_n, α_n, and B_n. Because the foreground layers already have an alpha channel no extra processing is necessary for the transparent regions or the boundary mixed pixels.

5 Results

5.1 Alpha Maps

Figure 3 shows the obtained alpha maps for the well known ballet and breakdancers datasets [6]. A close up of two relevant regions is presented in Figure 4. The dancers in the scene have a mixed pixel boundary several pixels wide, as seen in 4. The alpha values for these mixed pixels were succesfully recovered without any user input. Hair presents a challenge for alpha estimation and even though the semi-transparent region of Figure 4 has an uneven width, its alpha matte was also extracted properly. The central region of the breakdancer has no alpha values because the background could not be observed in any of the images. Yet the mixed pixel boundary was also detected.

Figure 4 shows how the algorithm labels as semi-transparent the area where the yellow sleeve and black vest meet. These pixels are indeed mixed pixels as can be observed by the mixture of yellow and black on the border. However, when the algorithm classifies them as transparent, it incorrectly assumes that they are mixed with the wall behind.

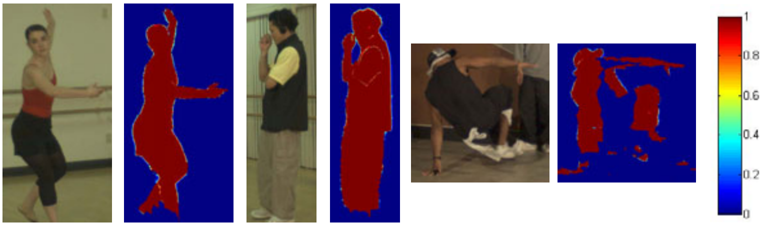


Fig. 3. Extracted alpha maps for the characters in the scenes

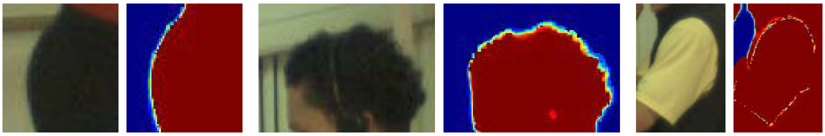


Fig. 4. Close up of semi-transparent regions. **Left:** Boundary pixels. **Middle:** Semi-transparent hair. **Right:** Incorrect estimation.

5.2 Free Viewpoint Rendering

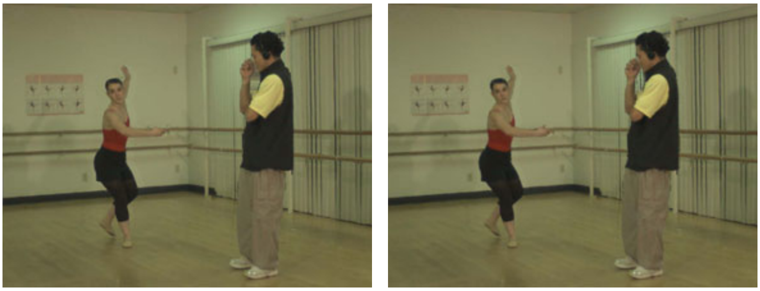
A novel view generated using our method is presented in Figure 5. Müller et al.'s state-of-the-art method presented in [14] was implemented and used as a comparison. At a broad scale, both algorithms produce novel views of similar quality. Close ups of the most relevant differences are presented in Figure 6.

On Figure 6a it can be observed how an error in the depth map causes the thumb to be warped incorrectly by Müller et al.'s method. The alpha matte estimation stage of our algorithm succesfully recovers from this error in the depth map and assigns the thumb to the proper place.

A semi-transparent region made of hair is presented on Figure 6b. Müller et al.'s method produces an unnaturally sharp and even boundary for the hair. The alpha map obtained with our method allows a more natural look of the hair.

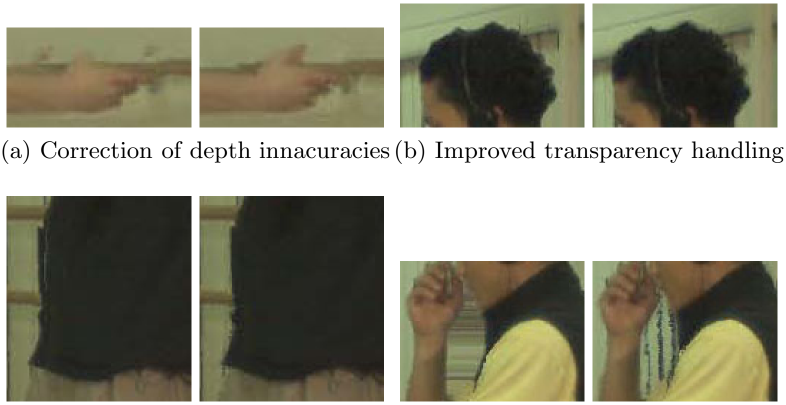
Figure 6c shows an artifact present in Müller et al.'s approach due to the wall being incorrectly assigned to the foreground, similar to a ghosting artifact. Our method does not suffer from this type of artifacts. However, our method presents more noise on the border. The noise suggests that the alpha smoothing stage could be improved. The current algorithm enforces spatial smoothness only on the alpha map and not in the foreground or background color maps.

Finally, Figure 6d shows that the naïve hole filling approach used in [14] is not suited to big holes in the background. Because our method uses the entire dataset for the alpha estimation stage, the background can be recovered from other images and no hole filling is necessary.



(a) Müller et al.'s method [14] (b) Our method using alpha matte

Fig. 5. Comparison of synthesized views from a novel viewpoint



(a) Correction of depth innacuracies (b) Improved transparency handling

(c) Removal of line artifact on left border (d) Naïve hole filling vs. recovered background

Fig. 6. Comparison of synthesized views from a novel viewpoint. **Left column:** Müller et al.'s method. **Right column:** our proposed method.

6 Conclusions

We presented a multi-view alpha estimation algorithm that requires no user interaction. It handles arbitrary scene geometry using pre-computed depth maps. It automatically detects semi-transparent pixels in the images. The algorithm handles mixed boundary pixels and hair regions correctly estimating their transparency and true colors.

Using the results of the alpha estimation algorithm, a novel free viewpoint rendering pipeline was developed and compared to the state of the art. The alpha estimation stage allowed the free viewpoint rendering algorithm to correct some depth map errors. The obtained results are of high quality and removed several artifacts found in the state-of-the-art methods. Future research can focus on better use of spatial information during alpha estimation and in simultaneous depth and transparency estimation.

References

1. Smolic, A.: 3d video and free viewpoint video-from capture to display. In: Pattern Recognition (2010)
2. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: CVPR (2009)
3. Gastal, E., Oliveira, M.: Shared sampling for real-time alpha matting. Computer Graphics Forum 29(2) (2010)
4. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. In: PAMI (2007)
5. Wang, J., Cohen, M.: Optimized color sampling for robust matting. In: CVPR (2007)
6. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. ACM Trans. on Graph (Proc. SIGGRAPH) (2004)
7. Chuang, Y., Curless, B., Salesin, D., Szeliski, R.: "A bayesian approach to digital matting. In: CVPR (2001)
8. Hasinoff, S., Kang, S., Szeliski, R.: Boundary matting for view synthesis. Computer Vision and Image Understanding 103(1) (2006)
9. Joshi, N., Matusik, W., Avidan, S.: Natural video matting using camera arrays. ACM Trans. Graph 25 (2006)
10. Wexler, Y., Fitzgibbon, A., Zisserman, A.: Image-based environment matting. In: Proceedings of the 13th Eurographics Workshop on Rendering (2002)
11. Wexler, Y., Fitzgibbon, A.W., Zisserman, A.: Bayesian estimation of layers from multiple images. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 487–501. Springer, Heidelberg (2002)
12. Shade, J., Gortler, S., He, L., Szeliski, R.: Layered depth images. In: SIGGRAPH (1998)
13. Frick, A., Kellner, F., Bartczak, B., Koch, R.: Generation of 3d-tv ldv-content with time-of-flight camera. In: 3DTV Con. (2009)
14. Müller, K., Smolic, A., Dix, K., Merkle, P., Kauff, P., Wiegand, T.: View synthesis for advanced 3d video systems. EURASIP Journal on Image and Video Processing (2008)
15. Fitzgibbon, A., Wexler, Y., Zisserman, A.: Image-based rendering using image-based priors. In: ICCV (2003)
16. Smith, A., Blinn, J.: Blue screen matting. In: Proc. of ACM SIGGRAPH (1996)
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. In: PAMI, vol. 23(11) (2002)

Using Spatially Distributed Patterns for Multiple View Camera Calibration

Martin Grochulla, Thorsten Thormählen, and Hans-Peter Seidel

Max-Planck-Institut Informatik, Saarbrücken, Germany
{mgrochul,thormae}@mpi-inf.mpg.de

Abstract. This paper presents an approach to intrinsic and extrinsic camera parameter calibration from a series of photographs or from video. For the reliable and accurate estimation of camera parameters it is common to use specially designed calibration patterns. However, using a single pattern, a globally consistent calibration is only possible from positions and viewing directions from where this single pattern is visible. To overcome this problem, the presented approach uses multiple coded patterns that can be distributed over a large area. A connection graph representing visible patterns in multiple views is generated, which is used to estimate globally consistent camera parameters for the complete scene. The approach is evaluated on synthetic and real-world ground truth examples. Furthermore, the approach is applied to calibrate the stereo-cameras of a robotic head on a moving platform.

1 Introduction

Camera parameter estimation is the task of finding the intrinsic and extrinsic camera parameters, which describe the projection of the 3D scene onto the 2D image plane of the camera. From one calibrated camera the line of sight for a given pixel can be computed; in combination with stereo algorithms two calibrated cameras can be used in combination with stereo algorithms to estimate the depth for a given pixel [1]. Furthermore, camera calibration is required for a number of computer vision applications in areas such as augmented reality, robot navigation, or special effects generation.

A common method for camera calibration is the usage of a calibration object or calibration pattern for which the geometry is known. The knowledge of the geometry of the pattern provides points in 3D space, while the corresponding 2D points are extracted from the image. With these extracted 2D-3D correspondences the camera parameters can be estimated. Popular approaches for camera calibration were presented by Tsai [2,3] and Zhang [4,5], both using calibration patterns.

To compute the intrinsic parameters of the camera Zhang uses at least two images of the pattern from different orientations. Tsai on the other hand uses only a single image of a calibration pattern to estimate the extrinsic and intrinsic camera parameters in a two stage approach. In both approaches the calibration pattern consists of squares arranged in a grid. The corners of the squares are

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 110–121, 2011.

the 3D points used for calibration. Thus, four 3D points for each square are obtained. However, finding the correct 2D position of the corner points in the image can be difficult and error-prone considering the noise and blur present in the images.

Tsai and Zhang both use a single pattern for camera parameter estimation. However, if multiple cameras or if cameras in a larger environment have to be calibrated, the problem arises that this may not be possible with a single pattern, since only cameras that see the pattern can be calibrated.

If the task is to calibrate multiple cameras in a scene, one possibility is to use the approach of Ueshiba and Tomita [6]. This approach uses a single calibration pattern, which is placed at three or more locations, where a separate set of images is taken for each location. Another possibility is to use multi-camera self-calibration [7]. In this approach, instead of calibration patterns, a single laser pointer is moved in the calibration volume. Tracking the position of the laser pointer in each image of each camera allows to self-calibrate the cameras. However, both approaches require static cameras and, thus, can not handle multiple images of a single moving camera.

If the task is to calibrate a moving camera in the scene, self-calibration can be employed. This approach does not require a special calibration object. Intrinsic camera parameters are computed from multiple uncalibrated images taken by the camera. The movement of the camera provides enough constraints for computing the intrinsic parameters [8,9]. However, camera self-calibration is a complex and difficult task, where degenerate cases can occur.

For the application of augmented reality, Fiala et al. [10,11] developed a system called ARTag that employs multiple coded markers to calibrate the camera. This system consists of a set of different markers and algorithms to detect the orientation and the position of the markers in the image. The goal is to augment the image/video with rendered 3D virtual content by detecting the relative position and orientation of several markers to each other.

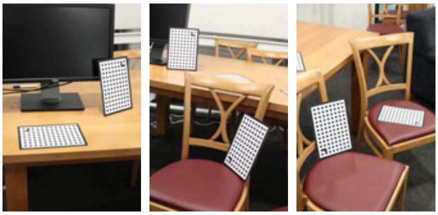


Fig. 1. The suggested approach allows camera calibration from images that have a partial overlap. Each one of the shown images contains two calibration patterns where one of these patterns is also visible in the next image.

In this paper, we present an approach for performing camera calibration from a series of images or from a video with multiple patterns. We neither restrict our approach to require a pattern to be visible in all views nor a camera to see all patterns. In contrast to existing work, the approach is very general and works with one or multiple moving or static cameras.

The approach is easy to apply in practice, as a user only has to distribute the calibration patterns in the scene such that in each view some of them are visible (see Fig. 1 for an example). The patterns are coded so that they can be identified in different images. It is not necessary for all patterns to be visible in all views; a few patterns per view are sufficient as long as relative position and orientation between every pattern or camera can be computed. Instead of using the corners of the squares on the pattern as 2D points, we use the centers of gravity of the projected squares. This has the advantage of a more reliable detection. However, the center of gravity does not always coincide with the geometric center of the squares under perspective projection. Therefore, these 2D points must be refined after an initial parameter estimation of cameras and patterns, in order to achieve a camera calibration with high accuracy.

2 Camera Calibration

This chapter describes our approach to estimate the intrinsic and extrinsic parameters of multiple cameras (either static or moving) using multiple coded patterns.

Tsai and Zhang are using the corners of the squares on the calibration patterns as points. Finding those corner points becomes less accurate with smaller size of the pattern in the image. Hence, we use the centers of gravity as initial 2D points, as they are easier to detect. In order to extract enough 2D-3D correspondences for camera parameter estimation, the patterns used in our approach consist of eight rows of squares with twelve squares in a row, arranged in a grid (see Fig. 2). To detect the orientation of the pattern, we use an L-shaped marker in one corner of the pattern, which replaces three squares. To be able to distinguish different patterns, we use an identifier for each pattern. The identifier is coded in the last row of the pattern. It is a binary coded number with squares representing 0s and rectangles representing 1s. The rectangles are twice as long and half as thick as the squares resulting in the same surface area as the squares. The grid of squares, including the coded identifier and the marker, are surrounded by a frame. Using such a calibration pattern, provides enough 2D-3D correspondences for the estimation process.

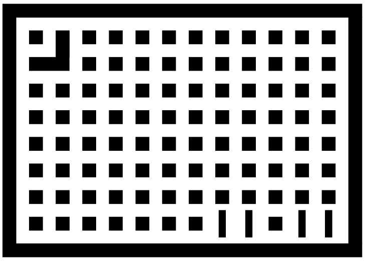


Fig. 2. Calibration pattern used in our approach. The L-shaped marker is used to detect the orientation of the pattern, while in the last row a pattern identifier is encoded binary (here pattern no. 27 = 11011_2 is shown).

2.1 Pattern Identification and Point Extraction

To identify the patterns and extract 2D points, we proceed as follows. First, we threshold an image with a specified threshold t resulting in a binary image. Color images are converted to gray-scale before thresholding. In the binary image we perform a region analysis, where a single region consists of all pixels having the same value (black or white) and being 4-connected. Patterns are then identified in the image as regions having a given number of neighbor regions The L-shaped marker is then identified as the second biggest neighbor region within the pattern. Knowing the orientation of the pattern, we are able to identify the last row of the pattern. In order to distinguish squares from rectangles, which encode the pattern identifier, we compute the standard deviation of all 2D pixel positions in the region. In practice, this is already enough to distinguish squares from rectangles since the standard deviation for the rectangles will be significantly bigger than the one of the squares. Thereby, the expected standard deviation of the squares is known from looking at the second last row of the pattern, which only contains squares.

Finally, the initial 2D points for the parameter estimation are computed as the mean of the pixel positions of each region. Note that the mean of the pixel positions of a region corresponds to the center of gravity of the projected square. Although the center of gravity is the same as the geometric center for a square or a rectangle in 2D, this does not hold for projected squares or rectangles in 3D space. However, the center of gravity is a good approximation of the geometric center and our algorithm does work well with these measurements. Nevertheless, once camera parameters are estimated, the measured 2D points can be compensated with the current camera parameters in order to refine the camera parameters.

2.2 Connection Graph Generation

By distinguishing different patterns in the images, we can generate a connection graph. This graph is an abstract representation of the connections between camera views and visible patterns. In the graph cameras and patterns are represented by nodes (see Fig. 3). A camera node is connected to a pattern node by an edge, if the pattern represented by its node is visible in the view of the camera represented by its node. The edge means position and orientation of a camera with respect to a pattern can be estimated.

Our approach uses the following two ideas. On the one hand, if two patterns are visible in one image, the position and orientation between those patterns can be estimated (see section 2.3: single view alignment). On the other hand, if one pattern is visible in two different views, the position and orientation between the two cameras can be estimated (see section 2.3: multiple view alignment).

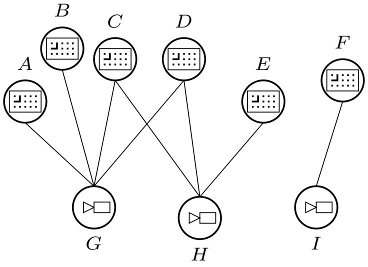


Fig. 3. Connection graph. Edges between camera nodes (G, H, and I) and pattern nodes (A – F) represent the ability to estimate camera parameters. A path from node A to node E indicates the possibility to compute position and orientation of these two patterns relative to each other, while this is not possible for nodes E and F.

For the example shown in Fig. 3, this means relative positions and orientations between these nodes can be estimated, as there is an edge between each pattern node A, B, C, and D and the camera node G. By looking at the edges of the transitive closure of the connection graph, it is possible to determine if it is also possible to estimate the relative transformation between two camera nodes in the graph. For the example shown in Fig. 3 it is possible to relate camera nodes G and H, but not H and I.

2.3 Camera Parameter Estimation

Having generated the connection graph, we will now show how positions and orientations of cameras and patterns in the scene are estimated. For simplicity, in the following we will describe the problems as if the scene is observed by multiple static cameras. However, a single moving camera or multiple moving cameras can be handled in the same way by just generating a new virtual static camera for each point in time.

The estimation of the camera parameters is done in six steps (compare Fig. 4):

1. Estimation of position and orientation between a single pattern and the camera using Tsai's approach,
2. Alignment of all patterns visible in one image,
3. Estimation of positions and orientations between *all* patterns in an image and the camera,
4. Alignment of all cameras and all patterns,
5. Estimation of positions and orientations between *all* patterns and *all* cameras, and
6. Refinement of 2D points and re-estimation of camera parameters (optional)

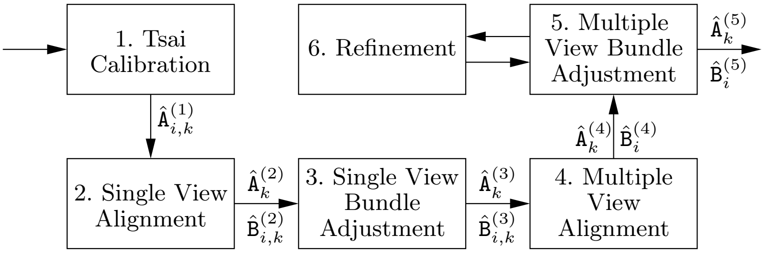

1. Tsai Calibration
6. Refinement
5. Multiple View Bundle Adjustment
$\hat{\mathtt{A}}_k^{(5)}$
$\hat{\mathtt{B}}_i^{(5)}$
$\hat{\mathtt{A}}_{i,k}^{(1)}$
$\hat{\mathtt{A}}_k^{(4)}$
$\hat{\mathtt{B}}_i^{(4)}$
2. Single View Alignment
$\hat{\mathtt{A}}_k^{(2)}$
$\hat{\mathtt{B}}_{i,k}^{(2)}$
3. Single View Bundle Adjustment
$\hat{\mathtt{A}}_k^{(3)}$
$\hat{\mathtt{B}}_{i,k}^{(3)}$
4. Multiple View Alignment

Fig. 4. Algorithm overview

In our approach, camera view k is represented by its projection matrix $\mathtt{A}_k$:

$$\mathtt{A}_k = \begin{bmatrix} f_k & 0 & p_{x,k} \\ 0 & f_k & p_{y,k} \\ 0 & 0 & 1 \end{bmatrix} [\,\mathtt{I}\,|\,\mathbf{0}\,] \begin{bmatrix} \mathtt{R}_k & -\mathtt{R}_k\,\underset{\sim}{\mathbf{C}}_k \\ \mathbf{0}^\top & 1 \end{bmatrix} = \mathtt{K}_k\,[\,\mathtt{I}\,|\,\mathbf{0}\,] \begin{bmatrix} \mathtt{R}_k & -\mathtt{R}_k\,\underset{\sim}{\mathbf{C}}_k \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (1)$$

with the 3×3 calibration matrix $\mathtt{K}_k$, the 3×3 rotation matrix $\mathtt{R}_k$, and the 3-vector $\underline{\mathbf{C}}_k$. The calibration matrix $\mathtt{K}_k$ contains the intrinsic camera parameters, where f_k is the focal length and $p_{x,k}$ and $p_{y,k}$ are the principal point offsets in x- and y-direction, respectively. The rotation matrix is composed of consecutive rotations around the y-, x- and z-axis with Euler angles φ, ϑ, and ρ: $\mathtt{R} = \mathtt{R}_z(\rho) \cdot \mathtt{R}_x(\vartheta) \cdot \mathtt{R}_y(\varphi)$. The camera center is represented by $\underline{\mathbf{C}}_k$. A pattern i is represented by a 4×4 transformation matrix $\mathtt{B}_i = \left[\begin{smallmatrix} \mathtt{S}_i & -\mathtt{S}_i \underline{\mathbf{D}}_i \\ \mathbf{0}^\top & 1 \end{smallmatrix}\right]$, with 3-vector $\underline{\mathbf{D}}_i$ representing the pattern center and 3×3 rotation matrix $\mathtt{S}_i$ composed of consecutive rotations around the y-, x- and z-axis with Euler angles α, β, and γ. The projection of a 3D point $\mathbf{P}$ of pattern i given in homogeneous coordinates in the pattern coordinate system is then given by

$$\mathbf{p} = \mathtt{K}_k \left[\, \mathtt{I} \,|\, \mathbf{0} \,\right] \begin{bmatrix} \mathtt{R}_k & -\mathtt{R}_k \underline{\mathbf{C}}_k \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathtt{S}_i & -\mathtt{S}_i \underline{\mathbf{D}}_i \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{P}, \tag{2}$$

where $\mathbf{p}$ is the corresponding 2D point in the image plane of camera k given in homogeneous coordinates.

Radial distortion is modeled as follows. Let $(x_u, y_u)^\top$ be the undistorted position of the projection of the 3D point $\underline{\mathbf{P}}$. The distorted position of the projection of $\underline{\mathbf{P}}$ is then modeled as $x_d = \big(1 + \kappa_3 r_u^2 + \kappa_5 r_u^4\big) x_u$ and $y_d = \big(1 + \kappa_3 r_u^2 + \kappa_5 r_u^4\big) y_u$, where r_u is the distance of $(x_u, y_u)^\top$ from the principal point $(p_x, p_y)^\top$. Here, $(1 + \kappa_3 r_u^2 + \kappa_5 r_u^4)$ is an approximation of the real radial distortion function with a Taylor series and κ_3 and κ_5 are the parameters describing the lens distortion.

For the sake of clarity, we are denoting the resulting cameras matrices $\mathtt{A}_k$ and pattern transformations $\mathtt{B}_i$ of the m-th processing step with an additional index: $\mathtt{A}_k^{(m)}$ and $\mathtt{B}_i^{(m)}$ (compare Fig. 4).

Tsai Calibration. Having identified all patterns i in a camera view k, we use Tsai's approach [3] to estimate the position and orientation of the camera $\hat{\mathtt{A}}_{i,k}^{(1)}$ relative to the pattern i. To be able to estimate the parameters, we have to provide 2D-3D correspondences between the image and the pattern. The (measured) 2D points $\tilde{\mathbf{p}}_i$ of pattern i are given by the 2D-3D point extraction (see section 2.1). Here we use the centers of gravity of the found regions. The corresponding 3D points $\mathbf{P}_i$ in the pattern coordinate system are given by the known structure of the pattern. Since our calibration pattern is planar, we assume for the sake of simplicity and without loss of generality that all 3D points lie in the x-y-plane and that the geometric center of the whole pattern lies at the origin. During this first processing step we define that the local coordinate system of the pattern coincides with the world coordinate system. Thus, we have: $\hat{\mathtt{B}}_{i,k}^{(1)} = \mathtt{I} \quad \forall k$.

Since we assume that the pattern lies in the x-y-plane around the origin, Tsai's algorithm provides estimated parameters $\hat{\mathtt{A}}_{i,k}^{(1)}$ for the location and orientation of camera k with respect to pattern i by minimizing the cost function:

$$\underset{\hat{\mathtt{A}}_{i,k}^{(1)}}{\operatorname{argmin}} \sum_{i,j,k} d\big(\tilde{\mathbf{p}}_{j,k},\, \hat{\mathtt{A}}_{i,k}^{(1)} \mathtt{B}_{i,k}^{(1)} \mathbf{P}_j\big)^2 \qquad \forall\, i,\, k, \tag{3}$$

where $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between homogeneous points $\mathbf{x}$ and $\mathbf{y}$.

Single View Alignment. Having an estimate $\hat{\mathtt{A}}^{(1)}_{i,k}$ of the position and orientation of the camera k with respect to every visible pattern i in the image, we are now able to compute parameters $\hat{\mathtt{B}}^{(2)}_{i,k}$ for all patterns in the image. Since there is only one camera view for each image in reality, the different estimated camera parameters $\hat{\mathtt{A}}^{(1)}_{i,k}$ are in fact resulting from different positions of the patterns in the scene. Therefore, we align the different estimated cameras to a single reference camera with $\mathtt{R} = \mathtt{I}, \underline{\mathbf{C}} = \mathbf{0}^\top$ for every camera view k. From the different camera parameters

$$\hat{\mathtt{A}}^{(1)}_{i,k} = \hat{\mathtt{K}}^{(1)}_{i,k} \, [\, \mathtt{I} \,|\, \mathbf{0} \,] \begin{bmatrix} \hat{\mathtt{R}}^{(1)}_{i,k} & -\hat{\mathtt{R}}^{(1)}_{i,k} \, \hat{\underline{\mathbf{C}}}^{(1)}_{i,k} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{4}$$

we now compute estimates of the positions of the patterns $\hat{\mathtt{B}}^{(2)}_{i,k}$ in 3D space relative to a reference camera $\hat{\mathtt{A}}^{(2)}_{k}$ for all patterns i in all views k. This is done by setting

$$\hat{\mathtt{B}}^{(2)}_{i,k} := \begin{bmatrix} \hat{\mathtt{R}}^{(1)}_{i,k} & -\hat{\mathtt{R}}^{(1)}_{i,k} \, \hat{\underline{\mathbf{C}}}^{(1)}_{i,k} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad \forall \, i, \, k \quad \text{and} \quad \hat{\mathtt{A}}^{(2)}_{k} := \frac{1}{n_k} \sum_i \hat{\mathtt{K}}^{(1)}_{i,k} \, [\, \mathtt{I} \,|\, \mathbf{0} \,] \quad \forall \, k, \tag{5}$$

with n_k denoting the number of patterns visible in view k. Note that we simply average the intrinsic parameters $\hat{\mathtt{K}}^{(1)}_{i,k}$ from Tsai's estimations to get an estimate of the intrinsic parameters of the reference camera $\hat{\mathtt{A}}^{(2)}_{k}$.

For the single view alignment, we have $\hat{\mathtt{A}}^{(1)}_{i,k} \mathtt{B}^{(1)}_{i,k} \mathbf{P} = \mathbf{p}^{(1)} \approx \mathbf{p}^{(2)} = \hat{\mathtt{A}}^{(2)}_{k} \hat{\mathtt{B}}^{(2)}_{i,k} \mathbf{P}$, as can be verified by Eq. (2). Here, $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(2)}$ are only approximately equal due to the parameter averaging in Eq. (5).

Single View Bundle Adjustment. After the single view alignment we perform a bundle adjustment for every camera view k. The bundle adjustment minimizes the reprojection error. The reprojection error is the sum of distances between the measured 2D points $\tilde{\mathbf{p}}_{j,k}$ in the image plane and the projections of corresponding estimated 3D world points $\hat{\mathtt{B}}_{i,k} \mathbf{P}_j$ (both represented in homogeneous coordinates). Bundle adjustment uses non-linear Levenberg-Marquardt optimization ($\hat{\mathtt{A}}^{(2)}_{k}$ and $\hat{\mathtt{B}}^{(2)}_{i,k}$ are used for the initialization of $\hat{\mathtt{A}}^{(3)}_{k}$ and $\hat{\mathtt{B}}^{(3)}_{i,k}$):

$$\operatorname*{argmin}_{\hat{\mathtt{A}}^{(3)}_{k}, \, \hat{\mathtt{B}}^{(3)}_{i,k}} \sum_{i,j} d\big(\tilde{\mathbf{p}}_{j,k}, \, \hat{\mathtt{A}}^{(3)}_{k} \hat{\mathtt{B}}^{(3)}_{i,k} \mathbf{P}_j\big)^2 \qquad \forall \, k \quad . \tag{6}$$

Multiple View Alignment. After having estimated camera parameters $\hat{\mathtt{A}}^{(3)}_{k}$ and pattern parameters $\hat{\mathtt{B}}^{(3)}_{i,k}$ for every single image k with the single view bundle adjustment, we now estimate globally consistent camera and pattern parameters $\hat{\mathtt{A}}^{(4)}_{k}$ and $\hat{\mathtt{B}}^{(4)}_{i}$, respectively. Using the generated connection graph, we are able to select two images k and k' that have at least one pattern i in common.

Consistent parameters are computed for these two images by fixating the camera and pattern parameters of one image ($\hat{\mathtt{A}}_k^{(4)} := \hat{\mathtt{A}}_k^{(3)}, \hat{\mathtt{B}}_i^{(4)} := \hat{\mathtt{B}}_{i,k}^{(3)}$) and transform the camera and pattern parameters of the other image in the following way: $\mathtt{T}_{k,k'} := \hat{\mathtt{B}}_{i,k}^{(3)} \big(\hat{\mathtt{B}}_{i,k'}^{(3)}\big)^{-1}$ is a transformation to align views k and k' where pattern i is the link between those views. The equation $\hat{\mathtt{A}}_{k'}^{(4)} := \hat{\mathtt{A}}_{k'}^{(3)} \mathtt{T}_{k,k'}^{-1}$ transforms the camera of view k' and the equation $\hat{\mathtt{B}}_i^{(4)} := \mathtt{T}_{k,k'} \hat{\mathtt{B}}_{i,k'}^{(3)}$ aligns all patterns of view k'. However, we only transform pattern that have not already been aligned before. Using the connection graph we align all views by consecutively aligning one unaligned view with all views that have already been processed. In addition, using the connection graph we are able to detect constellations where the views cannot be aligned.

Multiple View Bundle Adjustment. If all cameras and all patterns have been aligned, we perform another bundle adjustment to minimize the reprojection error of the patterns $\hat{\mathtt{B}}_i^{(5)}$ in all camera views $\hat{\mathtt{A}}_k^{(5)}$ (similarly, $\hat{\mathtt{A}}_k^{(4)}$ and $\hat{\mathtt{B}}_i^{(4)}$ are used to initialize of $\hat{\mathtt{A}}_k^{(5)}$ and $\hat{\mathtt{B}}_i^{(5)}$):

$$\underset{\hat{\mathtt{A}}_k^{(5)},\ \hat{\mathtt{B}}_i^{(5)}}{\operatorname{argmin}} \sum_{i,j,k} d\big(\tilde{\mathbf{p}}_{j,k}, \hat{\mathtt{A}}_k^{(5)} \hat{\mathtt{B}}_i^{(5)} \mathbf{P}_j\big)^2 \quad . \tag{7}$$

Refinement. Since the center of gravity of the projected square does not back-project to the geometric center of the square, 2D points are optionally refined.

Having computed globally consistent camera and pattern parameters $\hat{\mathtt{A}}_k^{(5)}$ and $\hat{\mathtt{B}}_i^{(5)}$ and knowing the size of the squares (or rectangles), we are able to project the corners of the squares into the image plane. For each square we get four points forming a quadrilateral. From these four corner points we can then compute the projection of the geometric center $\hat{\mathbf{c}}_{\text{geo}}$ and the projection of the center of gravity $\hat{\mathbf{c}}_{\text{grav}}$ of the square.

The projection of the geometric center is the projection of the intersection of the diagonals of that quadrilateral. For the projection of the center of gravity of the square we first compute the centers of gravity of all four possible triangles of the quadrilateral. The center of gravity of a triangle is the arithmetic mean of its corners. The four computed centers of gravity form another quadrilateral. The projection of the center of gravity of the original quadrilateral is then the intersection of the diagonals of the second quadrilateral.

We then update the initial 2D points $\tilde{\mathbf{p}}_{j,\,\text{init}}$ with

$$\tilde{\mathbf{p}}_{j,\,\text{new}} := \tilde{\mathbf{p}}_{j,\,\text{init}} - \hat{\mathbf{c}}_{\text{grav}} + \hat{\mathbf{c}}_{\text{geo}} \tag{8}$$

and repeat the multiple view bundle adjustment of section 2.3 with the refined 2D points. If necessary, the refinement of the 2D points together with the multiple view bundle adjustment can be iterated. However, we observed the largest improvement to usually occur after the first iteration.

Table 1. The synthetic image sequence. **Left**: Comparison between estimated camera parameters and ground truth without 2D point refinement. **Right**: Comparison between estimated camera parameters and ground truth with one iteration of 2D point refinement.

Δf	$\Delta \mathbf{C}$	RMSE of $\{\varphi, \vartheta, \rho\}$
[mm]	[mm]	[rad]
0.00861	0.07769	1.8215E−04
0.06058	0.26194	9.3651E−04
0.01667	0.10513	7.0942E−04
0.00287	0.03357	2.5241E−04
0.00593	0.11377	1.1414E−04
0.00130	0.13067	1.1775E−04
0.00637	0.06130	9.9409E−05
0.00428	0.07432	8.3503E−04
0.05864	0.11363	1.1946E−03
RMSE for whole series of images		
0.02936	0.12170	6.3581E−04

Δf	$\Delta \mathbf{C}$	RMSE of $\{\varphi, \vartheta, \rho\}$
[mm]	[mm]	[rad]
0.00166	0.01962	1.7127E−04
0.01492	0.05603	7.9734E−04
0.00311	0.02640	6.3498E−04
0.00106	0.01355	2.2049E−04
0.00872	0.05367	1.1054E−04
0.00238	0.03902	1.2050E−04
0.00026	0.01868	1.0056E−04
0.00161	0.03465	6.7190E−04
0.06134	0.07239	1.1350E−03
RMSE for whole series of images		
0.02852	0.10208	5.6700E−04

3 Results

This section presents results of tests performed to evaluate our approach. In a first test, we used synthetic data to be able to compare the estimated camera parameters with the ground truth. In a second test, we took images from several calibration patterns located on scale paper to compare estimated pattern positions with measured pattern positions. Finally, we applied our method to calibrate two cameras of a robotic head which can perform human-like movements.

3.1 Synthetic Ground Truth Example

We rendered a series of 9 images of a scene in which we placed six calibration patterns (see Fig. 5). Generating a synthetic series of images enabled us to compare our estimation results with the ground truth (see Tab. 1). The rendered virtual room had a size of approximately 16 square meters, which is important to put the accuracy of results in Tab. 1 into relation. Compared to the overall extend of the scene, the observed errors can be regarded as very low.

We performed camera parameter estimation using our approach. In the first run we estimated camera and pattern parameters *without* refining the 2D points obtained from Sec. 2.1. In a second run we then used the 2D point refinement from Sec. 2.3. Although we found the estimation results using the initial 2D points to be good, they could be improved by the 2D point refinement (an improvement of approx. 3%, 16%, and 11% percent for focal length, camera center and camera rotation, respectively). As expected, the first iteration of the refinement resulted in the biggest improvements, while further iterations did not improve the results significantly.

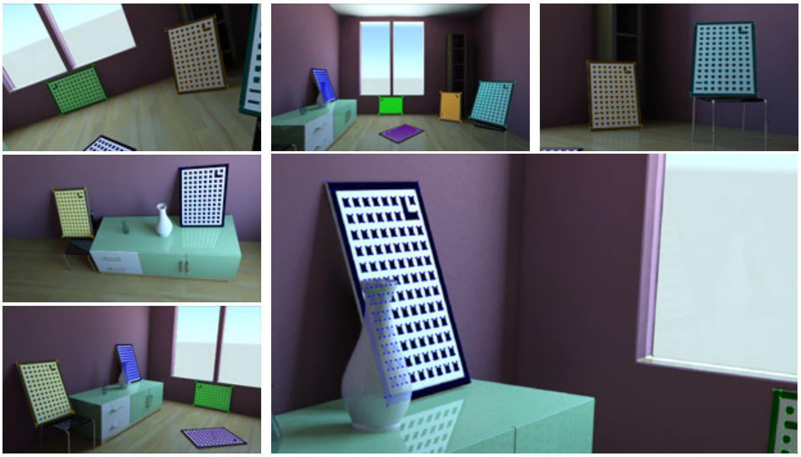


Fig. 5. The synthetic image sequence. **Top row** and **left column**: 5 out of 9 images shown. The positions of the projections of the estimated patterns into the image planes of the estimated cameras can be seen. **Right bottom**: Detail of one of the images. Due to occlusion this pattern is not visible in the image, however, from the other images the position of the pattern can be estimated accurately.

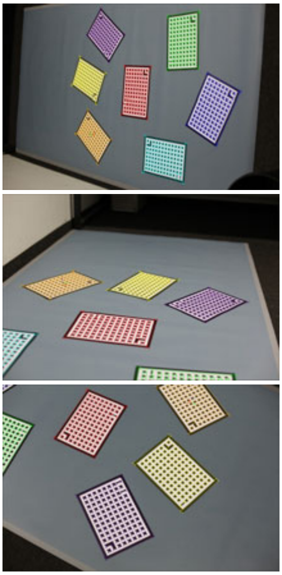
3.2 Real-World Ground Truth Example

For a real-world example we printed seven calibration patterns and arranged them on a paper with millimeter scale. Twelve images of this scene were taken. Since we placed the pattern on the scale paper, in this example all patterns lay in one plane. By using scale paper we were able to measure the corners of the patterns. From the corner points we then computed the centers of the patterns. After applying our camera calibration approach, we were able to compare the estimated pattern positions to the measured ones (see Tab. 6b). The largest absolute difference in Tab. 6b is 1.14 mm in relation to a 583 mm absolute pattern distance (corresponding to a relative deviation of 0.2%).

3.3 Application Example

We applied our approach to the calibration of the stereo-cameras of a robotic head. The cameras of the robotic head are able to move like human eyes and the head of the robot is mounted on a rotating platform. Our goal was to calibrate both cameras for different viewing directions. Because of the ability to look in different direction with the cameras, it is impossible to calibrate the cameras using a single pattern only.

With our method it was possible to calibrate the cameras. The result is shown in Fig. 7. By distributing several calibration patterns in front of the robotic head we managed to see at least one pattern in every image of the robot's cameras. Additionally, we found the calibration procedure to be very easy to apply, as we did not have to pay attention to locate or align the calibration patterns in a specific way with respect to each other.



—	876.94	354.41	368.27	477.54	823.98	829.50
876.94	—	714.22	711.34	399.43	285.58	333.02
354.41	714.22	—	568.29	382.47	559.94	807.66
368.27	711.34	568.29	—	380.13	784.79	540.83
477.54	399.43	382.47	380.13	—	406.04	425.21
823.98	285.58	559.94	784.79	406.04	—	582.95
829.50	333.02	807.66	540.83	425.21	582.95	—

—	876.63	354.20	368.03	477.58	823.16	829.40
876.63	—	713.65	711.24	399.09	285.16	332.37
354.20	713.65	—	567.77	382.19	559.11	806.94
368.03	711.24	567.78	—	380.14	784.00	541.16
477.58	399.09	382.19	380.14	—	405.20	424.76
823.16	285.16	559.11	784.00	405.20	—	581.81
829.40	332.37	806.94	541.16	424.76	581.81	—

—	0.30	0.21	0.23	0.04	0.82	0.11
0.30	—	0.58	0.10	0.34	0.42	0.65
0.21	0.58	—	0.51	0.28	0.83	0.72
0.23	0.10	0.51	—	0.01	0.80	0.33
0.04	0.34	0.28	0.01	—	0.83	0.44
0.82	0.42	0.83	0.80	0.83	—	1.14
0.11	0.65	0.72	0.33	0.44	1.14	—

(a) Image sequence: 3 out of 12 images shown. Patterns placed on paper with millimeter scale. Projections of estimated patterns into the camera images are overlaid.

(b) Evaluation results. All values are given in millimeters. Distance between pattern i and j is given in column i and row j. **Top**: Measured distances between pattern centers. **Middle**: Distances between estimated pattern positions. **Bottom**: Difference between top and middle table.

Fig. 6. Real-world image sequence and evaluation results

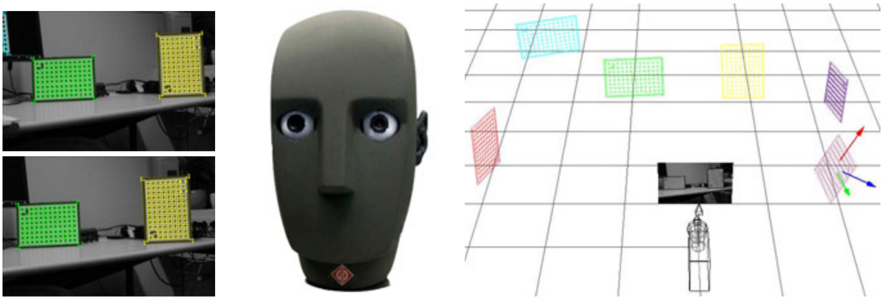


Fig. 7. Application example: This robotic head has two cameras as eyes. The images at the left are two camera images taken with the eye-cameras (calibration patterns are overlaid). The scene at the right shows all reconstructed calibration patterns in 3D space.

4 Conclusion

In this paper we have presented an approach for calibrating multiple camera views in a globally consistent coordinate frame. We make use of multiple calibration patterns that can be distributed in the scene, which makes our approach flexible and easy to use. Furthermore, it addresses shortcomings of single pattern based calibration methods. Intrinsic and extrinsic camera parameters are estimated as well as position and orientation of the calibration patterns. For reliable and accurate estimation results we have simplified the 2D point extraction. After an initial parameter estimation the approximate 2D points are refined to increase accuracy.

Our method has been evaluated on a synthetic and a real-world series of images showing the high flexibility and accuracy of the method. Additionally, the approach has been applied to calibrate the stereo-cameras of a robotic head.

Future work will address how to use several patterns without pattern identifiers, as it should be possible to distinguish the patterns by their relative positions in space.

References

1. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: Stereo and Multi-Baseline Vision, pp. 131–140 (2001)
2. Tsai, R.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: Computer Vision and Pattern Recognition, pp. 364–374 (1986)
3. Tsai, R.: 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 3, 323–344 (1987)
4. Zhang, Z.: Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In: International Conference on Computer Vision, vol. 1, pp. 666–673 (1999)
5. Zhang, Z.: A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1330–1334 (2000)
6. Ueshiba, T., Tomita, F.: Plane-based Calibration Algorithm for Multi-camera Systems via Factorization of Homography Matrices. In: International Conference on Computer Vision, vol. 2, pp. 966–973 (2003)
7. Svoboda, T., Martinec, D., Pajdla, T.: A Convenient Multicamera Self-Calibration for Virtual Environments. Presence: Teleoperators and Virtual Environments 14, 407–422 (2005)
8. Maybank, S.J., Faugeras, O.D.: A Theory of Self-Calibration of a Moving Camera. International Journal of Computer Vision 8, 123–152 (1992)
9. Hartley, R.I.: An Algorithm for Self Calibration from Several Views. In: Computer Vision and Pattern Recognition, pp. 908–912 (1994)
10. Fiala, M.: ARTAG Rev2 Fiducial Marker System: Vision based Tracking for AR. Presented at the Workshop of Industrial Augmented Reality (2005)
11. Fiala, M.: ARTag – augmented reality (2009), http://www.artag.net/ (retrieved on March 10, 2011)

Optimal Gabor Filters and Haralick Features for the Industrial Polarization Imaging

Yannick Caulier[1] and Christophe Stolz[2]

[1] Fraunhofer Institute for Integrated Circuits. Am Wolfsmantel 33,
91058 Erlangen, Germany

[2] Laboratoire Electronique Informatique et Image. 12, rue de la fonderie,
71200 Le Creusot, France

Abstract. During the past decade, computer vision methods for inline inspection became an important tool in a lot of industrial processes. During the same time polarization imaging techniques rapidly evolved with the development of electro-optic components, as e.g. the polarization cameras, now available on the market. This paper is dedicated to the application of polarization techniques for visually inspecting complex metallic surfaces. As we will shortly recall, this consists of a direct image interpretation based on the measurement of the polarization parameters of the light reflected by the inspected object. The proposed image interpretation procedure consists of a Gabor pre-filtering and a Haralick feature detector. It is demonstrated that polarization images permit to reach higher classification rates than in case of a direct interpretation of images without polarization information.

1 Introduction

The inspection of complex industrial parts for the inline real-time inspection, requires adapted and efficient information retrieval and processing approaches. This implies that the lighting technology but also the corresponding processing methodology must be adapted to the specificities of the inspection task. The central and decisive element is the surface to be characterized, i.e. the task of automatically visual enhancing and classifying the defective regions to be detected. Workpiece geometry, reflectivity and handling possibility are the major parameters which must be taken into consideration for the determination of the appropriate inspection methodology. Within this context, existing quality control methods are based on the interpretation of the disturbances induced by the surface on the projected light. Important information can be the geometry of the light wave in case of structure-based approaches, the wavelength for multispectral methods or the phase of the wave for polarization-based methods.

The commonly used formalism for the description of the polarization state of an electromagnetic wave is the Stokes vector [Goldstein, 2003]. The complete polarization information can be described with the Stokes vector, consisting of 4 parameters S0, S1, S2, S3 and describing the type and degree of polarization

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 122–132, 2011.

of the wave. The computation of this vector necessitates a sensor which is sensitive to the polarization information. In general CCD cameras are coupled with additional polarization or liquid cristal filters. Concerning the method for the interpretation of the polarization information, we distinguish between direct and indirect approaches. In case of the former, the relevant information is directly retrieved from the recorded polarization images for a qualitative interpretation [Terrier et al., 2008]. For the latter, the polarization is used to recover the depth information for a quantitative interpretation [Morel et al., 2006].

This paper is focussed on the direct interpretation of the polarization information, related to important surface and material properties, such as the geometry, the texture, the reflectivity or the type. A new direct surface interpretation method, based on the computation of appropriate features determined with an appropriate image processing chain, is proposed. As the proposed research is dedicated to the optimal characterization of industrial surfaces, a reference database has specially been defined for the purposes of this paper. The class and number of images were chosen in accordance with previous investigations dedicated to a similar inspection task, where a structured light approach instead of a polarization one has been considered [Caulier and Bourennane, 2010]. The considered artificially produced defects have a clear geometry. This will help to define general rules for further quality control experiments based on real defects.

The purpose of the paper is to propose a stepwise polarization-based image processing approach, consisting of an enhancement of the relevant image information by means of a Gabor filter method [Chengjun and Wechsler, 2003] and of a characterization of this image signature using the statistic approach of Haralick co-occurrence matrices [Porebski, 2008]. The first evaluates the amount of energy for certain directions and image resolutions, the second evaluates the occurrence of pixel pairs in an image. The classification rate is used for the evaluation of the complete method, i.e. for the determination of the optimal parameters of both approaches.

Different open and relevant problems within the context of non-destructive testing for industrial quality control are tackled. Purpose of the proposed researches are therefore:

- to determine how far the visual enhancement based on polarization imaging is relevant for the industrial quality control,
- to evaluate if polarization-based approaches permit a direct interpretation of the image contents,
- to evaluate if a previous Gabor image filtering leads to more appropriate image signatures in case of direct image interpretation approaches,
- to find out which polarization features are the most appropriate,
- to propose a complete optimal image processing chain for polarization imaging.

The rest of the paper is organized as follows. The proposed method is described in section 2. Section 3 addresses the experimental results. Section 4 concludes this paper.

2 Proposed Method

2.1 Polarization-Based Image Enhancement Principle

As described previously, the purpose of this paper is the surface characterization by means of the direct interpretation of polarization images. The geometrical and textural properties of the samples to be characterized, which are illuminated with a diffuse unpolarized light, are contained in the polarization state of the reflected light by the surfaces. This information is described by the Stokes vector and processed in the corresponding Stokes images for the direct interpretation of the surfaces to be characterized.

$\mathbf{I}_{S0}$ which represents the total intensity without polarizer has the property $\mathbf{I}_{S0}^2 = \mathbf{I} >= \mathbf{I}_{S1}^2 + \mathbf{I}_{S2}^2 + \mathbf{I}_{S3}^2$, can be assimilated to the intensity image, i.e. the image which would be obtained without polarization filter. In case of the considered diffuse illumination in this paper, this would signify that $\mathbf{I}_{S0}$ permits to reveal textural surface characteristics. $\mathbf{I}_{S1}$ and $\mathbf{I}_{S2}$ are the difference of two images corresponding to 90 degree rotated polarization filter positions $\{0, 90\}$ and $\{45, 135\}$. As the textural surface changes do not influence the polarization of the incoming light, contrariwise to the geometrical surface variations, this means that these two Stokes images permit to reveal the geometrical surface information. To facilitate the interpretation of the Stokes images, we evaluate the linear degree of polarization image $\mathbf{I}_{dop}$ and the angle of polarization $\mathbf{I}_{AMP}$ image. The latter is linked to the local slope of the surface.

The considered surfaces were chosen in accordance with the open problems tackled within the industrial inspection context : metallic parts with different defect types. The reference samples for image database elaboration were chosen in order to be representative of the surfaces to be characterized, in accordance with the requirements defined by the quality of automotive cast parts. This is the reason why both non-acceptable and acceptable surfaces have been considered. Furthermore, two different types of defects, synonymous of geometrical and texture changes of the surface, were recorded. These three defect classes corresponding to the considered acceptable surfaces, and non-acceptable geometrical and textural ones, are named Ω_{OK}, Ω_{3D} and Ω_{2D}. The database contains self-made artificial defects of different sizes, depths (0.4 to 1.0 mm), and material (aluminum, copper, steel, brass).

Fig. 1 shows the polarization principle by means of a reference sphere and five samples of the considered reference database. A majority of the considered defects are geometrical ones, whereas the remaining correspond to painting marks. The three Stokes images $\mathbf{I}_{S0}$, $\mathbf{I}_{S1}$ and $\mathbf{I}_{S2}$ are depicted for the six pieces. The degree and angle of polarization images $\mathbf{I}_{dop}$ and $\mathbf{I}_{aop}$ are shown for the sphere.

Fig. 1 shows that the considered Stokes, degree and angle of polarization images permit the enhancement of geometrical and textural surface information. Polarization even enables the discrimination of these two surface classes Ω_{3D} and Ω_{2D}, see the images of the sphere and the central depicted artificial geometrical defect. The three highlighted images of classes Ω_{OK} and Ω_{3D} in Fig. 1 clearly show that the intensity image $\mathbf{I}_{S0}$ does not reveal the relevant information for a clear discrimination between acceptable and non acceptable surfaces.

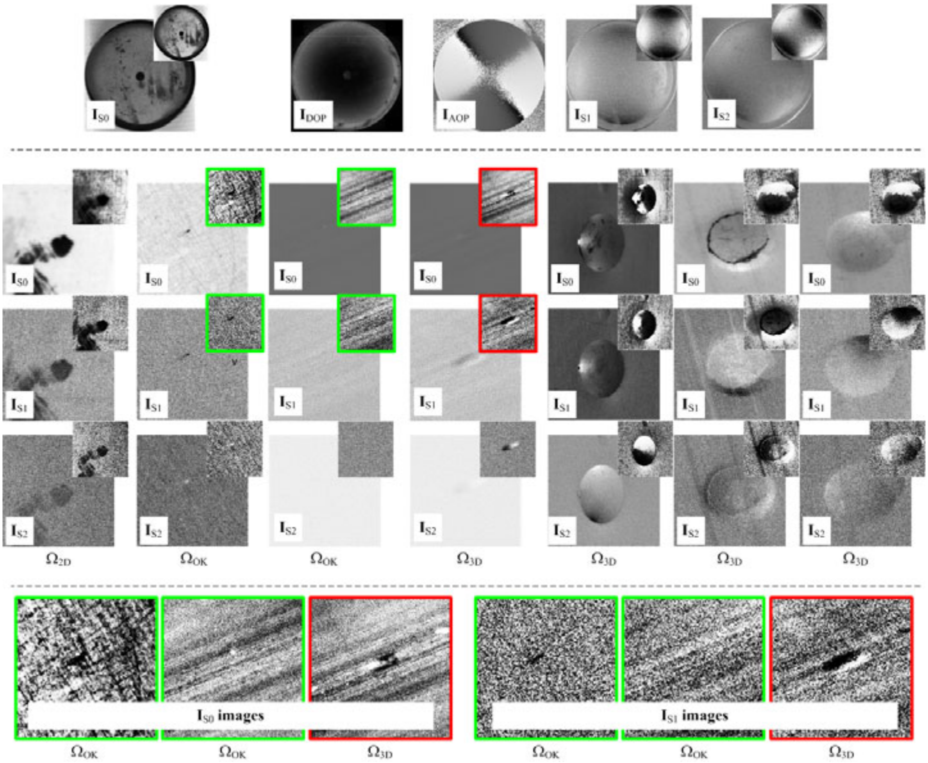


Fig. 1. Polarization-based image visual enhancement principle. Upper images: principle explanation with a sphere. Middle images: some exemples taken from the reference database Bottom images: selected examples showing the relevance of polarization information, as depth defects are better visually enhenced with $\mathbf{I}_{S1}$ then with $\mathbf{I}_{S0}$ images. For a better understanding the original image and its contrast enhanced are depicted.

Thus, as all the considered scenes recordings and therefore the relevant information to be extracted and characterized, are represented with grey level images, the relevant surface information will be characterized by local grey level variations corresponding to local geometric and/or textural variations synonymous of defective surface parts.

According to the surface types or recording conditions, different types of perturbing noise exist. Noise can be due to acceptable geometrical surface variations, such as the grinding marks on the artificial surface, but also to the painting on the surface inducing "salt and pepper" perturbations. Indeed, the micro-structures of the painted coatings locally provoke higher light reflections. These images also show that optimal recording conditions were deliberately not considered, as the purpose is to define an approach for complex industrial surface interpretation. A non perfectly homogeneous lighting was considered, so that additional grey level variations non synonymous of critical surface, such as perturbing glares e.g., were considered.

The purpose is now to define an appropriate approach permitting the classification of the considered surfaces in case of the three classes problem $\{\Omega_{3D}, \Omega_{2D}, \Omega_{OK}\}$. In the following the proposed feature-based surface classification approach is described.

2.2 The Gabor Filters

As seen previously, the revealed geometrical and textural surface characteristics are depicted with different local grey level variations corresponding to different structures of different shapes and size in the Stokes images. A Gabor approach was considered to be an appropriate enhancing function, as these filters permit the enhancement of image structures of different shapes, frequencies and orientations. In the following a brief overview of 2D Gabor is provided.

According to the definition of Dunn [Dunn, 1995] which is based on the definition of Daugman [Daugman, 1985], a 2D Gabor filter h is an oriented complex sinusoidal wave h_{sin} modulated by a 2D Gaussian envelope h_{gau}, $h = h_{sin}.h_{gau}$. Filter main parameters are the wavelength $\lambda = 1/f$, f is the frequency, the standard deviation σ and the orientation α. Different values of these parameters permit the elaboration of different filters of different shapes, sizes and directions.

For the purpose of this paper, the Gabor filter definition given by Kovesi will be considered. The author defines three output images, $\mathbf{I}_r$, $\mathbf{I}_i$, $\mathbf{I}_a$, results of the convolution of the input image $\mathbf{I}_{in}$ with the (i) real part h_r of the 2D filter h, the (ii) imaginary part h_i of the 2D filter h and the (iii) amplitude of both real and imaginary images. Filter description according to Kovesi [Kovesi, 2011] is provided by the following equation:

$$f_r = cos(\frac{2.\pi}{\lambda}x\cdot) \cdot e^{-(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} \cdot f_{\alpha_g} \quad f_i = sin(\frac{2.\pi}{\lambda}x\cdot)e^{-(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} \cdot f_{\alpha_g}$$

$$\text{where} \quad \sigma_x = \lambda \cdot k_x \quad \text{and} \quad \sigma_y = \lambda \cdot k_y$$

$$\mathbf{I_r} = \mathbf{I}_{in} * f_r \quad \mathbf{I}_i = \mathbf{I}_{in} * f_i \quad \mathbf{I}_a = \sqrt{\mathbf{I}_r^2 + \mathbf{I}_i^2} \qquad (1)$$

f_{α_g} is a rotating function defined for an angle α in degrees (an angle of 0 gives a filter that responds to vertical features). The scale factors k_x and k_y control the filter σ_x and σ_y relative to the wavelength of the filter. This is done so that the shapes of the filters are invariant to the scale. k_x and k_y control the shape of the filter in the x- and y-directions.

The above described filters were designed for different shapes and sizes in the x- and y-directions. However, as no specific surface orientation of defect geometry is considered, no differentiation was done between the horizontal and vertical image directions, so that $\sigma = \sigma_x = \sigma_y$ and $k = k_x = k_y$.

2.3 Image Enhancement with Gabor Filters

The purpose is now to use the previously defined Gabor filters in order to enhance the depicted defective surface regions and, if possible, reduce the non-defective ones. As different defective regions sizes and types are considered, this problem is equivalent to determine the most optimal Gabor filter determined by its parameters (λ,k). In case of the considered defective surfaces, these parameters should be adapted to the size of the defects, which are between 1 to 100 pixels, if we consider e.g. the 3D defect borders or the 2D defect widths.

The "brute-force" approach would consist of varying the filter parameters (λ,k) for the whole range, in order to define the most optimal values. However, if we consider that the image information to be enhanced, i.e. the relevant signatures are contained in the low and high frequencies bands, the task will consist of finding a range of (λ,k) corresponding to these two frequency bands.

According to equation 1, λ permits to regulate the modulation of the *cos* and *sin* waves with the Gaussian envelope. For $\lambda = 1$ the filtering is equivalent to an image bluring with a Gaussian kernel, and therefore reveals low image frequencies. Higher values of λ and of k permit image filterings with Gaussian kernel modulated with *cos* or *sin* envelopes, which is equivalent to convolve the image with second or first derivative filter kernels. Thus, in case of high image frequencies enhancement and if the defect edges are considered as important images signatures, a variation of the variance σ of the Gaussian kernel between $\sigma \in [2 : 4]$ seems to be an adequate choice. Therefore, the following ranges of the Gabor filters were considered, $\lambda \in \{1, 2\}$ and $k \in \{1, 2\}$, so that $\sigma \in \{1, 4\}$.

Fig. 2 depicts some of the considered Gabor filters and examples of the considered output images $\mathbf{I}_r$, $\mathbf{I}_i$ and $\mathbf{I}_a$ for the three considered surface classes Ω_{OK}, Ω_{2D} and Ω_{3D}.

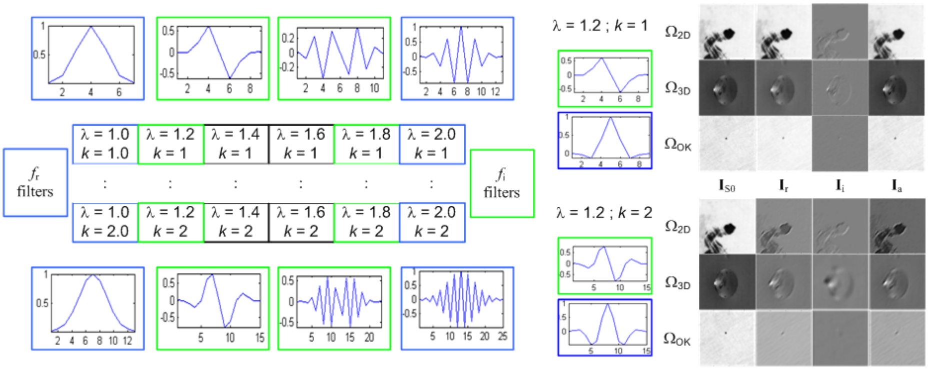


Fig. 2. Left: involved Gabor filters. Right: Example of enhanced structures with two Gabor filter for $\lambda = 1.2$ and $k = 2$.

The considered ranges of the filter parameters and the Gabor filter definition [Kovesi, 2011], permit to consider three kind of filtering techniques: (i) smoothing ones for $\lambda = k = 1$, (ii) first derivative ones for the imaginary filter part h_i and (iii) second derivative ones for the real filter part h_r. The images of Fig. 2 show how the different filters smooth or enhance the edges of the images according to the considered filter value parameters for $\lambda \in \{1, 2\}$ and $k \in \{1, 2\}$.

This prefiltering step shows how the considered Gabor approach permits to reveal different image structures, as e.g. the edges or the textures. However, the spatial arrangement and size of these structures are specific for each of the three considered surface types. These are circular structures for the Ω_{3D} defect class, no structures or linear ones for the Ω_{OK} defect class and random structures for the Ω_{2D} defect class.

The next step of the proposed approach consists of the classification of these structures by means of a texture analysis approach. For the purpose of the paper Haralick features will be considered.

2.4 The Texture Analysis Approach

Concerning the evaluation of different texture-based processing approaches, the involved methods were chosen according to the surface characteristics. The Haralick [Porebski, 2008] approach is a statistic-based method which permits to evaluate the occurrence of pixel pairs in an image. Each pair is characterized by its spatial and grey level relation. Main parameters are the pixel pair spatial and intensity characteristics, their distance in pixel $d_h \in \{1, d_{max}\}$, with d_{max} the maximum distance in the considered image, and their angle in degrees $\alpha_h \in \{0, 45, 90, 135\}$, measured counterclockwise with the horizontal image direction.

For the purposes of this paper, three Haralick features are considered: the contrast h_c, the homogeneity h_h and the energy h_e. Each feature value is the average of the four feature values for the four pixel directions $\alpha_h \in \{0, 45, 90, 135\}$.

2.5 Considered Image Processing Chain

The purpose is now to evaluate if the proposed Gabor pre-filtering approach leads to an improvement of the surface characterization and, if this is the case, to determine which Gabor filter is the most optimal. For this, two different image analysis methods are considered. The first directly computes the co-occurrence matrices and the three considered Haralick features on the Stokes image $\mathbf{I}_{S0}$, $\mathbf{I}_{S1}$ and $\mathbf{I}_{S2}$. The second, pre-filters the Stokes images with Gabor filters for four different angles $\alpha_g \in \{0, 45, 90, 135\}$, and computes for each direction the three images. $\mathbf{I}_r$ corresponding to the even filter h_r, $\mathbf{I}_i$ obtained with the odd filter h_i and the amplitude $\mathbf{I}_a$ images.

Fig. 3 shows the image analysis chain including both methods, and gives an example on two different surfaces, a non-acceptable Ω_{3D} and acceptable Ω_{OK} one.

The images in Fig. 3 show the influence of the polarization information in casse of the surface interpretation. The effect of the Gabor filtering on the visual enhancement on two different surface types is depicted in the depicted images. Whereas for the non-acceptable image, the grey level energy is spatially located on the defect, this energy is spread on the whole acceptable image. This means that the statistics defining the grey level distributions are different for both. This is the reason why the second order co-occurrence matrices approach is applied. These matrices are characterized using the three considered Haralick features h_c, h_h and h_e.

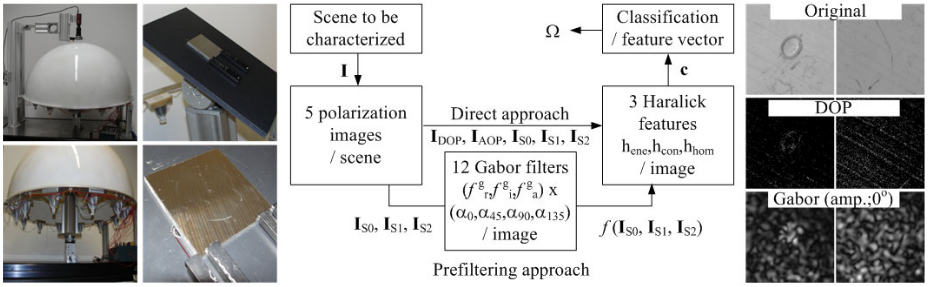

Scene to be characterized
I
5 polarization images / scene
Direct approach
$I_{DOP}, I_{AOP}, I_{S0}, I_{S1}, I_{S2}$
12 Gabor filters (f^g_r, f^g_i, f^g_a) x $(\alpha_0, \alpha_{45}, \alpha_{90}, \alpha_{135})$ / image
I_{S0}, I_{S1}, I_{S2}
$f(I_{S0}, I_{S1}, I_{S2})$
Prefiltering approach
3 Haralick features $h_{ene}, h_{con}, h_{hom}$ / image
c
Classification / feature vector
Ω
Original
DOP
Gabor (amp.;0°)

Fig. 3. Left: the recording set-up. Middle: the considered image processing chain. Right: two processed surfaces of class Ω_{3D} and Ω_{OK}

2.6 Laboratory Set-Up and Reference Database

The considered laboratory set-up consists of a lighting, a positioning element of the surfaces to be characterized and a CCD camera coupled with a linear rotating polarizer. The diffuse lighting is generated with 12 circular arranged LED-lamps of 5 W each around the object. The lamps, which are oriented on the top, are covered by a white opaque hemisphere for diffuse lighting generation. The acquisition part is composed of a 16 bits AVT Dolphin camera with a resolution of 1280x960 pixel with a linear Schneider Kreuznach polarization filter in front of the lens.

A set of seven different surfaces of aluminium copper, steel and brass was used for the elaboration of the reference database. On each of these seven surfaces one artificial 0.4 mm depth 3D defect was created. This surface region corresponds to a Ω_{3D} class. Some surfaces also contain 2D paint marks corresponding to Ω_{2D} class regions. All other surface regions are acceptable or do not contain critical defects. These belong therefore to the Ω_{OK} class. Each surface was recorded 12 times, which corresponds to 12 rotations of 30 degrees of the surface along the optical camera axis. In order to have an appropriate polarization contrast, the reference surfaces were positioned with an angle of 35 degrees with the optical axes of the camera.

3 Experimental Results

The experimental results concern the validation of the proposed preprocessing approach with Gabor filters and the determination of the appropriate Gabor and Haralick parameters. The final purpose is the determination of the most appropriate features for the surface characterization using the direct interpretation using polarization imaging.

3.1 Classification and Feature Selection

The determination of the appropriate Gabor and Haralick parameters was done by means of the classification rate C_r as evaluation criteria. Two different pro-

Table 1. Selected features, classification and false positive rates c_{sel}, C_p and C_{fp} for the different considered values of λ, k and h. The C_{fp} are indicated as exposants of C_p. The highest classification rates are highlighted for each value of h. The depicted selected features correspond to these highlighted values. The confusion matrix for the most optimal parameters $h = 60, w = 1.2, k = 1.2$ is listed.

h=1							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	79^{01}	90^{01}	85^{00}	89^{00}	88^{00}	86^{00}	$h_c(I_{dop})$
1.2	78^{01}	88^{01}	90^{00}	89^{00}	88^{00}	80^{00}	$h_h(I^a_{0^o}(I_{S0}))$
1.4	78^{01}	87^{01}	$\mathbf{91^{00}}$	90^{00}	87^{00}	80^{00}	
1.6	78^{01}	86^{01}	91^{00}	88^{00}	87^{00}	83^{00}	
1.8	78^{01}	90^{00}	90^{00}	85^{00}	86^{00}	80^{00}	
2.0	78^{01}	84^{00}	90^{00}	85^{00}	86^{00}	80^{00}	

h=60							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	81^{01}	96^{00}	88^{00}	88^{00}	88^{00}	84^{00}	$h_c(I_{S1})$
1.2	80^{01}	$\mathbf{96^{01}}$	90^{00}	88^{00}	88^{00}	83^{00}	$h_c(I_{dop})$
1.4	81^{01}	94^{01}	90^{00}	88^{00}	85^{00}	82^{00}	$h_e(I^i_{45^o}(I_{S1}))$
1.6	81^{01}	92^{00}	89^{00}	87^{00}	87^{00}	84^{00}	$h_c(I^i_{0^o}(I_{S2}))$
1.8	80^{01}	93^{00}	90^{00}	87^{00}	86^{00}	82^{00}	$h_c(I^i_{45^o}(I_{S2}))$
2.0	81^{01}	90^{00}	91^{00}	87^{00}	86^{00}	82^{00}	

h=5							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	79^{00}	91^{00}	87^{00}	88^{00}	89^{00}	86^{00}	$h_h(I_{dop})$
1.2	78^{00}	89^{01}	89^{00}	87^{00}	87^{00}	81^{00}	$h_c(I^r_{0^o}(I_{S1}))$
1.4	78^{00}	88^{00}	91^{00}	86^{00}	85^{00}	80^{00}	$h_e(I^i_{45^o}(I_{S1}))$
1.6	78^{00}	87^{00}	89^{00}	87^{00}	87^{00}	93^{00}	$h_c(I^r_{90^o}(I_{S1}))$
1.8	79^{00}	$\mathbf{91^{00}}$	88^{00}	84^{00}	85^{00}	80^{00}	
2.0	79^{00}	88^{00}	89^{00}	85^{00}	87^{00}	79^{00}	

h=80							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	80^{01}	94^{01}	86^{00}	89^{00}	89^{00}	83^{00}	$h_c(I_{S1})$
1.2	79^{01}	$\mathbf{95^{01}}$	90^{00}	86^{00}	87	81	$h_e(I_{dop})$
1.4	79^{01}	94^{01}	90^{00}	88^{00}	87^{00}	82^{00}	$h_c(I^i_{45^o}(I_{S0}))$
1.6	79^{01}	92^{01}	89^{00}	87^{00}	88^{00}	84^{00}	$h_c(I^i_{135^o}(I_{S1}))$
1.8	79^{01}	93^{00}	90^{00}	87^{00}	86^{00}	88^{00}	$h_c(I^i_{0^o}(I_{S2}))$
2.0	79^{01}	89^{00}	90^{01}	86^{00}	87^{00}	83^{00}	$h_h(I^i_{45^o}(I_{S2}))$

h=10							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	81^{00}	$\mathbf{93^{00}}$	84^{00}	88^{00}	89^{00}	87^{00}	$h_e(I_{S1})$
1.2	80^{00}	92^{00}	89^{00}	86^{00}	88^{00}	81^{00}	$h_h(I_{dop})$
1.4	80^{00}	89^{00}	92^{00}	87^{00}	86^{00}	81^{00}	$h_c(I^i_{135^o}(I_{S1}))$
1.6	80^{00}	87^{00}	89^{00}	87^{00}	86^{00}	85^{00}	$h_h(I^i_{90^o}(I_{S2}))$
1.8	80^{00}	91^{00}	89^{00}	87^{00}	86^{00}	83^{00}	
2.0	80^{00}	90^{00}	90^{00}	86^{00}	87^{00}	84^{00}	

h=100							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	81^{00}	$\mathbf{93^{00}}$	85^{00}	88^{00}	89^{00}	85^{00}	$h_c(I_{S1})$
1.2	80^{00}	91^{00}	91^{00}	88^{00}	88^{00}	82^{00}	$h_e(I_{S1})$
1.4	80^{00}	90^{00}	89^{00}	87^{00}	87^{00}	82^{00}	$h_c(I_{dop})$
1.6	79^{00}	88^{00}	89^{00}	88^{00}	89^{00}	85^{00}	$h_e(I^i_{0^o}(I_{S0}))$
1.8	79^{00}	89^{00}	90^{00}	86^{00}	88^{00}	82^{00}	$h_c(I^i_{90^o}(I_{S0}))$
2.0	79^{00}	89^{00}	89^{00}	87^{00}	87^{00}	86^{00}	$h_c(I^i_{45^o}(I_{S1}))$
							$h_h(I^i_{45^o}(I_{S2}))$

h=20							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	81^{00}	$\mathbf{94^{00}}$	87^{00}	88^{00}	88^{00}	85^{00}	$h_c(I_{dop})$
1.2	80^{00}	93^{00}	89^{00}	87^{00}	87^{00}	81^{00}	$h_c(I^i_{45^o}(I_{S0}))$
1.4	80^{00}	92^{00}	91^{00}	88^{00}	87^{00}	83^{00}	$h_c(I^i_{90^o}(I_{S0}))$
1.6	80^{00}	90^{00}	89^{00}	88^{00}	88^{00}	85^{00}	$h_c(I^r_{0^o}(I_{S1}))$
1.8	80^{00}	90^{00}	88^{00}	86^{00}	86^{00}	82^{00}	$h_h(I^i_{90^o}(I_{S1}))$
2.0	80^{00}	87^{00}	91^{00}	85^{00}	86^{00}	83^{00}	$h_e(I^i_{135^o}(I_{S1}))$

h=120							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	78^{00}	87^{01}	81^{00}	90^{00}	90^{00}	88^{00}	$h_e(I_{dop})$
1.2	78^{00}	83^{01}	89^{00}	90^{00}	86^{00}	79^{00}	$h_h(I^i_{135^o}(I_{S0}))$
1.4	77^{00}	83^{00}	91^{00}	89^{00}	85^{00}	80^{00}	$h_h(I^a_{0^o}(I_{S1}))$
1.6	78^{00}	81^{00}	$\mathbf{92^{00}}$	91^{00}	88^{00}	83^{00}	
1.8	78^{00}	87^{00}	92^{00}	87^{00}	86^{00}	82^{00}	
2.0	78^{00}	83^{00}	89^{00}	88^{00}	87^{00}	82^{00}	

h=40							
k/w	1.0	1.2	1.4	1.6	1.8	2.0	c_{sel}
1.0	81^{00}	$\mathbf{95^{00}}$	90^{00}	90^{00}	88^{00}	85^{00}	$h_e(I_{S1})$
1.2	81^{00}	94^{00}	89^{00}	88^{00}	87^{00}	82^{00}	$h_h(I_{dop})$
1.4	81^{00}	93^{00}	90^{00}	89^{00}	86^{00}	81^{00}	$h_c(I^i_{45^o}(I_{S0}))$
1.6	81^{00}	90^{00}	88^{00}	88^{00}	89^{00}	84^{00}	$h_h(I^i_{90^o}(I_{S1}))$
1.8	81^{00}	91^{00}	89^{00}	88^{00}	86^{00}	83^{00}	$h_e(I^i_{45^o}(I_{S2}))$
2.0	81^{00}	88^{00}	92^{00}	88^{00}	87^{00}	84^{00}	

Confusion matrix for h=60, w=1.2, k=1.2
$C_p = 96.1$

classified as ->	$\mathbf{\Omega}_{OK}$	$\mathbf{\Omega}_{2D}$	$\mathbf{\Omega}_{3D}$
$\mathbf{\Omega}_{OK}$	11	1	1
$\mathbf{\Omega}_{2D}$	1	6	1
$\mathbf{\Omega}_{3D}$	0	1	83

cessing chains were considered, i.e. whether a Gabor filter-based preprocessing was applied or not. Then, the rate C_p was computed for $\lambda \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, $k \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ and $h \in \{1, 5, 10, 20, 40, 60, 80, 100, 120\}$. The determination of the adequate features is done for the highest rate C_p, once for each value of h. Classification and wrapper-based feature selection were done using a Naive Bayes approach. The results are depicted in Table 1. The notations used for the selected features are self-explaining.

For the depicted results in Table 1, different remarks concerning the values of λ, k, h and the number and type of selected features can be done. These remarks hold for each of the nine 6×6 tables corresponding to the 9 values of h.

Highest classification rates were achieved for $\lambda = 1.2$ and $\lambda = 1.4$, whereas lowest for $\lambda = 1.0$. Concerning the influence of h highest rates are observed when

$h \in [40 : 80]$. The number of selected features remains low < 10 in comparison with the total amount of computed features > 50. In general, the features computed from the Gabor filtered images were more often selected than in case of the direct approach.

3.2 Result Interpretation

The fact that lower rates were reached for $\lambda = 1$ shows that the high image frequencies are important signatures, as for $\lambda = 1$ the images are smoothed. In other words, this means that the enhanced lower image frequencies do not contain the relevant polarization information.

An important geometrical feature seems to be the edges of the considered geometrical defects. The fact that highest classification rates were achieved for $\lambda = 1.2$, $\lambda = 1.4$ and $h \in [40 : 80]$, shows that the optimal combination consists of a (i) preliminary high pass Gabor filtering in order to enhance the defect borders and then of a (ii) characterization by means of Haralick distance-based features. In case of the considered defects $h \in [40 : 80]$, which corresponds to the observed optimal distance range, is also half the width of the geometrical defects.

Concerning the use of the Stokes images $\mathbf{I}_{S0}$, $\mathbf{I}_{S1}$, $\mathbf{I}_{S2}$ or the degree and angle of polarization images $\mathbf{I}_{dop}$, $\mathbf{I}_{aop}$ for a polarization-based image interpretation, former ones seem to be more appropriate than the latter. Indeed, if the features issued from $\mathbf{I}_{S0}$ and $\mathbf{I}_{dop}$ images were mostly selected, this is not the case for features computed on $\mathbf{I}_{aop}$ images. A possible explanation could be that the indetermination in the angle of polarization determination, which brings abrupt changes/discontinuities on a continuous Ω_{3D} surface.

Then, for the importance of polarization information, the feature selection results are also an indicator of feature importance. We remark, that for the all the considered values of h the first selected features are allways the polarization ones. This shows the relevance of polarization images in comparison to a purelly diffuse approach in case of the considered inspection task. This also corresponds to our explanation in case of Fig. 1. The depicted exemples clearly show the adequate visual enhencement in case of $\mathbf{I}_{S1}$ images in comparison to $\mathbf{I}_{S0}$ images.

4 Conclusion

In this paper a polarization-based direct image interpretation approach has been proposed and evaluated. The method consists of a preliminary filtering of the polarization image by means of Gabor filters and then of a characterization using the second order statistic approach of Haralick. The evaluation concerned the relevance of Gabor filters, the relevance of Stokes images in comparison to the degree and angle of polarization images, and the characterization of the whole proposed image content description method.

The results showed that (i) high classification rates $> 90\%$ can be reached and that (ii) the optimal polarization image characterization consists of a preliminary

Gabor filtering permitting to enhance higher frequency image values, combined with a feature-based characterization of the image edges. The proposed investigations showed the avantages of using Gabor filters and Haralick features. The evaluation methodology based on a classification rate computation and feature selection for different values of λ, k and h, permitted to retrieve optimal Gabor and Haralick parameters.

This approach can be applied to any other inspection task within the context of industrial inspection. Thus, possible further works could consist of considering a huger reference database containing more reference surfaces and also other types of defects.

References

Caulier and Bourennane, 2010. Caulier, Y., Bourennane, S.: Visually inspecting specular surfaces: A generalized image capture and image description approach. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)

Chengjun and Wechsler, 2003. Chengjun, L., Wechsler, H.: Independent component analysis of gabor features for face recognition. IEEE Trans. on Neural Networks 14(4), 919–928 (2003)

Daugman, 1985. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filter. Optical Society of America 2, 1160–1169 (1985)

Dunn, 1995. Dunn, D.: Optimal gabor filters for texture segmentation. IEEE Trans. in Image Processing 7(4), 947–964 (1995)

Goldstein, 2003. Goldstein, D.: Polarized Light (2003)

Kovesi, 2011. Kovesi, P.D.: Matlab and octave functions for computer vision and image processing (2011), http://www.csse.uwa.edu.au/~pk/Research/MatlabFns

Morel et al., 2006. Morel, O., Stolz, C., Meriaudeau, F., Gorria, P.: Active lighting applied to 3d reconstruction of specular metallic surfaces by polarization imaging. Applied Optics 45(17), 4062–4068 (2006)

Porebski, 2008. Porebski, A.V.N.M.L.: Haralick feature extraction from lbp images for color texture classification. In: First Workshops on Image Processing Theory, Tools and Applications, IPTA 2008, pp. 1–8 (2008)

Terrier et al., 2008. Terrier, P., Devlaminck, V., Charbois, J.-M.: Segmentation of rough surfaces using a polarization imaging system. Journal of the Optical Society America 25(2), 423–430 (2008)

Joint Histogram Modelling for Segmentation Multiple Sclerosis Lesions

Ziming Zeng[1,2] and Reyer Zwiggelaar[2]

[1] Faculty of Information and Control Engineering,
Shenyang Jianzhu University, Liaoning, China
[2] Department of Computer Science, Aberystwyth University, UK
{zzz09,rrz}@aber.ac.uk

Abstract. This paper presents a novel methodology based on joint histograms, for the automated and unsupervised segmentation of multiple sclerosis (MS) lesion in cranial magnetic resonance (MR) imaging. Our workflow is composed of three steps: locate the MS lesion region in the joint histogram, segment MS lesions, and false positive reduction. The advantage of our approach is that it can segment small lesions, does not require prior skull segmentation, and is robust with regard to noisy and inhomogeneous data. Validation on the BrainWeb simulator and real data demonstrates that our method has an accuracy comparable with other MS lesion segmentation methods.

1 Introduction

Multiple sclerosis (MS) is a disease of the central nervous system, which is the most common non traumatic neurological disease in young adults [1]. In clinical practice, Magnetic Resonance Imaging (MRI) plays an important role for determining MS lesions size and localization of affected tissue. Manual segmentation of MS lesions is both challenging and time-consuming. Several techniques have been proposed for automatic segmentation of MS lesions. Methods based on one modality can extract large lesions [2], but small lesions are difficult to distinguish from noise and image inhomogeneities. Most techniques rely on multiple modalities and exploit differences in contrast between various tissues [3,4]. Dugas-Phocion et al. [5] used multi-sequence MRI (T1,T2,T2 FLAIR, Proton Density) within an EM based probabilistic framework to segment MS lesions. Leemput et al. [6] proposed a fully automated atlas-based approach for MS lesion segmentation with T1, T2 and PD sequences. Aït-Ali et al. [7] proposed a multi-dimensional parametric method to segment MS lesions with multi-sequence MRI (T1, T2, PD) data. However, MRI data with significant noise and density inhomogeneities still provides a challenge. Also, the image registration process may increase segmentation errors. In response to these existing challenges, a scheme based on joint T1 and T2 histogram modelling is proposed to automatically segment MS lesions.

This paper is organized as follows. The MS lesion segmentation approach is described in Section 2. Section 3 provides the evaluation results and analysis

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 133–144, 2011.

on MR images from BrainWeb and clinical data. Conclusions and future explorations are presented in Section 4.

2 Overview of the Proposed Segmentation Model

As preprocessing, a mutual information method is used to registration the different modalities. Our segmentation method contains three steps (see Fig. 1 for an overview). In the first step, each corresponding pair of T1 and T2 slices are used to generate a joint histogram (256×256). Subsequently, the grey matter (GM) and white matter (WM) clusters in the joint histogram space are estimated and the MS lesion region is defined. In the second step, potential MS lesion areas are segmented. In the third step, a false positive reduction method is used to refine the segmentation results.

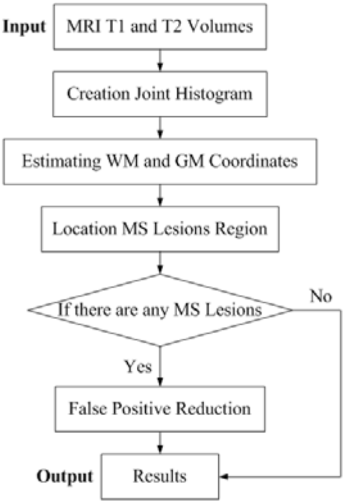


Fig. 1. Schematic representation of the proposed approach

2.1 Initial MS Lesions Region in Joint Histogram

The joint histogram is used to represent the number of occurrences of a pair of grey level values corresponding to the same position in two images. In brain MRI data, various anatomical tissues result in different grey levels and their distribution in the joint histogram for T1 and T2 MRI have distinct characteristics. In our work, joint histograms are used to incorporate the information which reflects the tissue distribution relationship for T1 and T2 for MS lesions. Fig. 2(a-d) show an example of a joint histogram and the distribution of various anatomical tissues. Compared with the histograms of T1 and T2 (i.e. Fig. 2(c) based on Fig. 2(a) and Fig. 2(b), respectively), tissues are separated in the joint histogram (i.e. Fig. 2(d) generated by Fig. 2(a) and Fig. 2(b)) whereas these overlap in the individual T1 and T2 histograms. Hence, the joint histogram can solve the overlap problem for various tissues. Fig. 2(e) shows the joint histogram based on a MS case in which we can see that the region related to the MS lesions in the joint histogram deviates from normal brain tissue, and the location of MS lesions is near the grey matter (GM) and white matter (WM) regions. The corresponding relationship of MS lesions in T1 and T2 is different from other tissues. Specifically, the MS lesions exhibit hyposignals in T1 and hypersignals in T2, with respect to normal white matter intensities [5,8]. This rule can be used to locate the general distribution of MS lesions in the joint histogram.

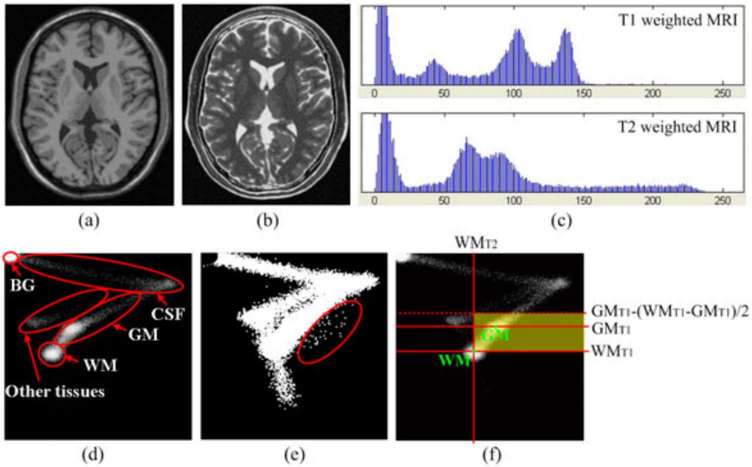


Fig. 2. An example of the joint histogram combining the information from T1 (a) and T2 (b) MRI images. (c) The T1 and T2 individual histograms. (d) The joint histogram with the various tissue regions indicated: BG: background, GM: grey matter, WM: white matter, and CSF: cerebrospinal fluid. (e) Binary joint histogram (The MS lesions region is located by a red circle). (f) MS lesions region (translucent yellow region) overlayed on the joint histogram.

We calculate the coordinates of the GM and WM cluster centers in the joint histogram, which are defined as (GM_{T1}, GM_{T2}) and (WM_{T1}, WM_{T2}), respectively. Based on the GM and WM cluster center locations we estimate the MS region within the joint histogram. For T1, the grey level of MS lesions is similar to GM tissue and expected to be hyposignal with regard to normal WM tissue [5,8]. With regard to T1 we define the boundaries of the MS lesions regions as WM_{T1} and $GM_{T1} - (WM_{T1} - GM_{T1})/2$ as indicated in Fig. 2(f). It should be noted that the choice of these T1 boundaries have an arbitrary aspect, but variations in these have indicated the robustness of the developed approach. With respect to T2, the grey level of MS lesions is hypersignal compared to normal white matter [5], so the MS lesions region can be defined as above WM_{T2}. The defined region also includes some other tissue regions which are excluded in subsequent steps.

2.2 MS Lesions Area Extraction in the Histogram

In this step, joint histograms are generated for each T1 and T2 volume. We select some slices to calculate the center coordinates of GM and WM. In general, the WM or GM cluster centers are less well defined for slices at the top and bottom of the volumes, but well defined at center slices. We find the coordinate centers by using average coordinates of the middle slices in the joint histogram volume. Firstly, we select the middle slice in the volume. To remove the background (BG) from the joint histogram, we detect the maximum in the joint histogram and use morphology to remove the surrounding area. Subsequently, a median filter (with a 5×5 window size) which can preserve image details is used to decrease the noise in the joint histogram. The choice of a median filter is arbitrary and could be replaced by other smoothing approaches such as Gaussian filtering. In order to consider both density information and spatial information, the class-adaptive Gaussian Markov modelling approach (CAGMRF) [9] is used to segment the joint histogram into five initial classes. We want to ensure that the highest intensity area (containing both the WM and GM regions) has two sub-regions, and we increase the number of groups until this is achieved. We obtain the coordinates corresponding to the maximum grey value in the WM and GM region by using WM and GM segmented areas separately. The WM and GM center coordinates are propagated to neighboring slices by first using a small circle around the coordinates. The circle area is used as the initial mask on the neighbor slices to find the coordinates which can be used for the next slice central points. Finally, according to our proposed model, we can define a rectangle region for MS lesions in the joint histogram by using the average coordinate of WM and GM which are generated in the previous steps. An example of the process is shown in Fig. 3.

At this stage, some other parts, such as WM and GM regions, are still included in the rectangle region. Fig. 4 shows an example of removing the non-lesions region. Firstly, a fuzzy C-mean method with two clusters is used to extract the brain tissue region. The result is shown in Fig. 4(b). Subsequently, erosion with a circular structuring element (radius equal to 3 pixels) is used to disconnect the normal brain tissue from the other regions and a 4-connected labeling method is used to identify the normal brain region (assuming the normal tissue tends to be the largest region present). Subsequently, the eroded area is regained by a dilation (radius equal to 1 pixel). Fig. 4(c) shows the result. We use the normal tissue region as a mask to remove the area which also exists in the rectangle region. Finally, we draw a line which is determined by the coordinate centers of WM and GM , and remove the region on the left side of the line. After removing the normal tissue region, the remaining region, which is shown in Fig. 4(d), represents the MS lesions. With this area, we can recognize whether MS lesions exist in the volume and then segment MS lesions by using this region.

2.3 MS Lesions Slices Recognition

The abscissa and ordinate of each pixel coordinate inside the MS region in the joint histogram which is generated from the previous steps are considered as the

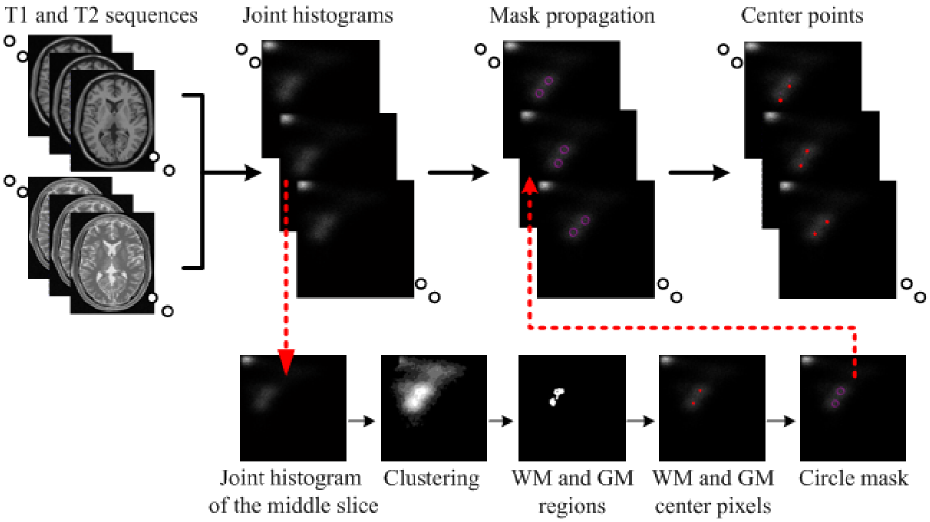


Fig. 3. The GM and WM coordinates estimation process

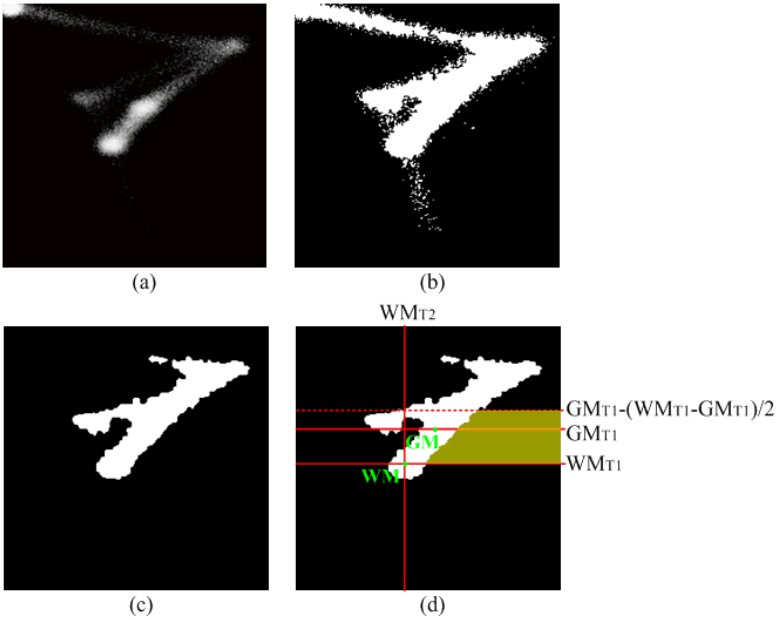


Fig. 4. An example of non-lesion region removal. (a) Joint histogram, (b) Fuzzy C-mean segmentation, (c) Morphological processing, (d) MS lesions region (translucent yellow region) overlayed on the joint histogram after non-lesion region removal.

grey level value in T1 and T2, respectively. Then two images are generated by using all the grey level values to the pixels in T1 and T2. Subsequently, the MS lesions can be found by logical 'and' for the two images. After that, a morphology method is used to fill small holes. By considering the volume information, false positive regions are removed. Specifically, all the connected areas in each slice generated by the previous steps are labeled, and only those regions which are located in roughly the same position in two neighboring slices in a volume are considered as real regions and all the others are treated as false positives. We use the interested regions as masks to extract the corresponding pixels in T2, then convolve the generated slice with a Gaussian kernel (radius is defined as 5). We define a threshold u to determine whether MS lesions exist in this slice. For each slice, if the maximum pixel value of the image generated by the previous steps is above u, the MS lesion exists and subsequently the false positive reduction step is used. Otherwise, the procedure moves to the next slice in question.

2.4 False Positive Reduction

For those slices which contain potential MS regions, false positive reduction is used to refine the segmentation results. In this step, a kernel function has been introduced for region-based active contour segmentation in an effort to solve the intensity inhomogeneities problem by extracting the intensity information at local regions as defined by the kernel. A region-scalable model (RSF) [10] and the GCS method [11] are used to obtain a convex behavior for the energy fitting function. Subsequently, the data energy fitting function is minimized along the deformation of the contour by using a split Bregman technique [12]. The energy function is defined as [10]

$$E(\phi, c_1, c_2) = \varepsilon(\phi, c_1, c_2) + \mu P(\phi) \tag{1}$$

where the level set regularization term is defined as

$$P(\phi) = \int \frac{1}{2}(\mid \nabla\phi(x) \mid -1)^2 dx \tag{2}$$

The region-scalable energy fitting function is defined as

$$\varepsilon(\phi, c_1, c_2) = \sum_{i=1}^{2} \lambda_i \int K_\sigma(x-y) \mid f_i(x) - I(y) \mid^2 M_i(\phi(y))dy \tag{3}$$

where the Gaussian kernel function is given by $K_\sigma(u) = (1/(2\pi\sigma^2))e^{-|u|^2/2\sigma^2}$. $K_\sigma(x-y)$ can be regarded as a weight on points y with regard to the center point x. Due to the local aspect of the kernel function, the effect on ε generated by I(y) is almost zero when y is further away from the center position x. The local fitting energy ε is determined by the value of σ. In this case, the energy of small region is small and these can be easily removed. In Eq. 3, $M_1(\phi) = H(\phi), M_2(\phi) = 1 - H(\phi)$, ϕ denotes the boundary of MS lesions. The Heaviside function H can be approximated by H_ε [13].

$$H_\varepsilon(x) = \frac{1}{2}\left[1 + \frac{2}{\pi} arctan\left(\frac{x - 0.5}{\xi}\right)\right] \tag{4}$$

According to the derivation written by Li et al. [10], the optimal functions $c_1(x)$, $c_2(x)$ that minimize $E(\phi, c_1, c_2)$ are obtained by:

$$c_i(x) = \frac{K_\sigma(x) * [M_i^\varepsilon(\phi(x))I(x)]}{K_\sigma(x) * M_i^\varepsilon(\phi(x))}, i = 1, 2 \tag{5}$$

For fixed $c_1(x)$, $c_2(x)$, the function ϕ is defined as

$$\frac{\partial \phi}{\partial t} = -\delta(\phi)(\lambda_1 e_1 - \lambda_2 e_2) - div(\frac{\nabla \phi}{\mid \nabla \phi \mid}) \tag{6}$$

where δ is the derivative of H_ε. $e_i(x) = \int K_\sigma(y - x) \mid I(x) - c_i(y) \mid_2 dy, i = 1, 2$.

The simplified flow represents the gradient descent for minimizing the energy:

$$E(\phi) = \mid \nabla \phi \mid + \langle \phi \cdot r \rangle \tag{7}$$

where $r = \lambda_1 e_1 - \lambda_2 e_2$. Bresson et al. [14] transformed the constrained optimization problem to an unconstrained optimization problem by restricting the solution to lie in a finite interval: $0 \le \phi \le 1$, the global convex model can be written as

$$min_{0 \le \phi \le 1} E(\phi) = min_{0 \le \phi \le 1}(\mid \nabla \phi \mid + \langle \phi \cdot r \rangle) \tag{8}$$

Goldstein et al. [12] used the Split Bregman algorithm to solve the global convex model. The Split Bregman algorithm for the minimization of Eq. 8 proposed by Yang et al. [15] can be summarized as

```
1: while $\| \phi^{k+1} - \phi^k \| > \Psi$ do
2:     Define $r^k = \lambda_1 e_1^k - \lambda_2 e_2^k$
3:     $\phi^{k+1} = GS(r^k, \overrightarrow{d^k}, \overrightarrow{b^k}, \lambda)$
4:     $\overrightarrow{d}^{k+1} = shrink_g(\overrightarrow{b^k} + \nabla \phi^{k+1}, 1/\lambda)$
5:     $\overrightarrow{b}^{k+1} = \overrightarrow{b}^k + \nabla \phi^{k+1} - \overrightarrow{d}^{k+1}$
6:     Find $\Omega^k = \{x : \phi^k(x) > \mu\}$
7:     Update $e_1^k$ and $e_2^k$
8: end while
```

where $GS(r^k, \overrightarrow{d^k}, \overrightarrow{b^k}, \lambda)$ denotes the Gauss-Seidal iteration method, the $\overrightarrow{b}$, $\overrightarrow{d}$ are auxiliary variables, the $shrink_g$ is a shrinkage frame (see [12,15]), and Ω is the refined MS lesions region.

After false positive reduction, we use the volume information to identify the false negative regions. Specifically, all the interested regions segmented by the previous steps are marked as true positive. Finally, considering the volume information, the corresponding relationship is used to find the false negative pixels in the neighboring slices which are generated by the second step.

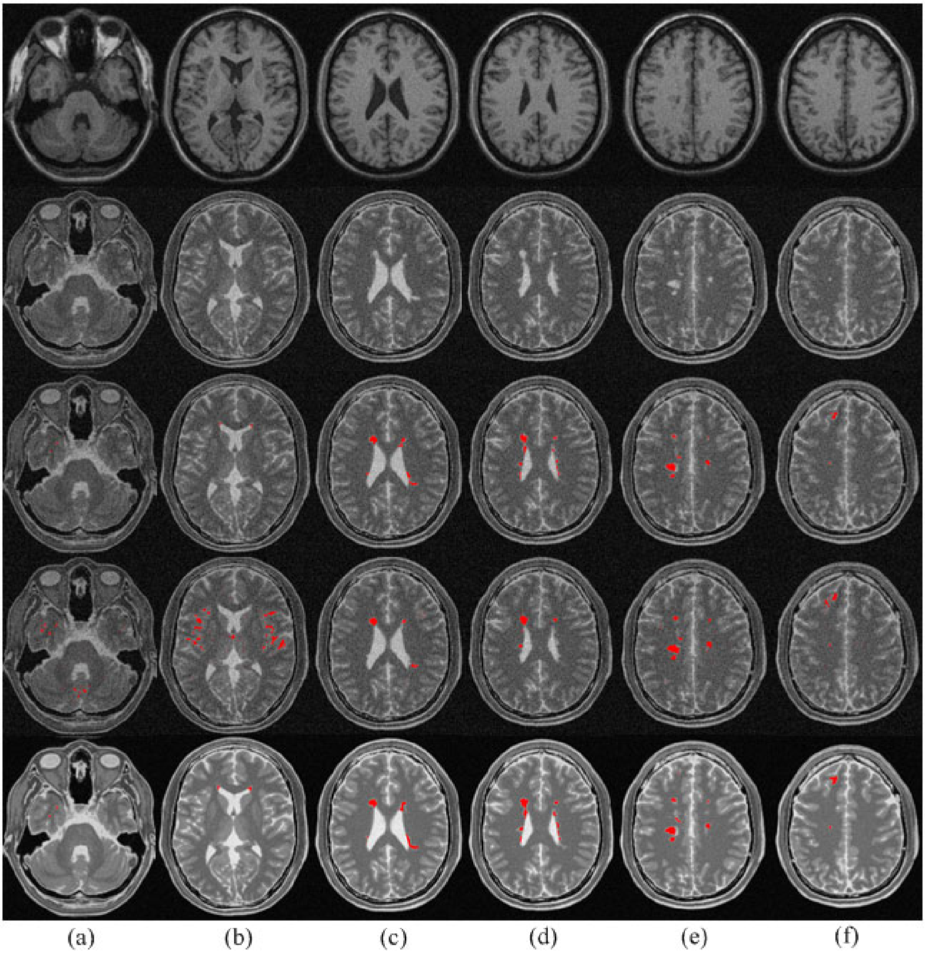


Fig. 5. Segmentation results. Red: MS Lesions. From left to right: slice 40, slice 81, slice 97, slice 100, slice 104, slice 111. The first row shows T1 MRI, the second row shows T2 MRI, the third row shows our results overlayed (red) on T2, the fourth row shows the segmentation results on T2 by using Rouaïnia's method [2], the fifth row shows the ground truth (red) overlayed on T2.

3 Experimental Results

The approach was tested on 9% noise, 40% intensity non-uniformity BrainWeb [16] simulated MRI data. Usually people report on lesion detection on MRI images with 3% noise level. The T1 and T2 volumes ($181 \times 217 \times 181$ voxels) are co-registered. The voxel size was $1mm^3$. In the first step, all the joint histograms are generated. Slice 90 in the histogram sequence is selected to find the center coordinate of the WM and GM clusters. Then two circle masks with

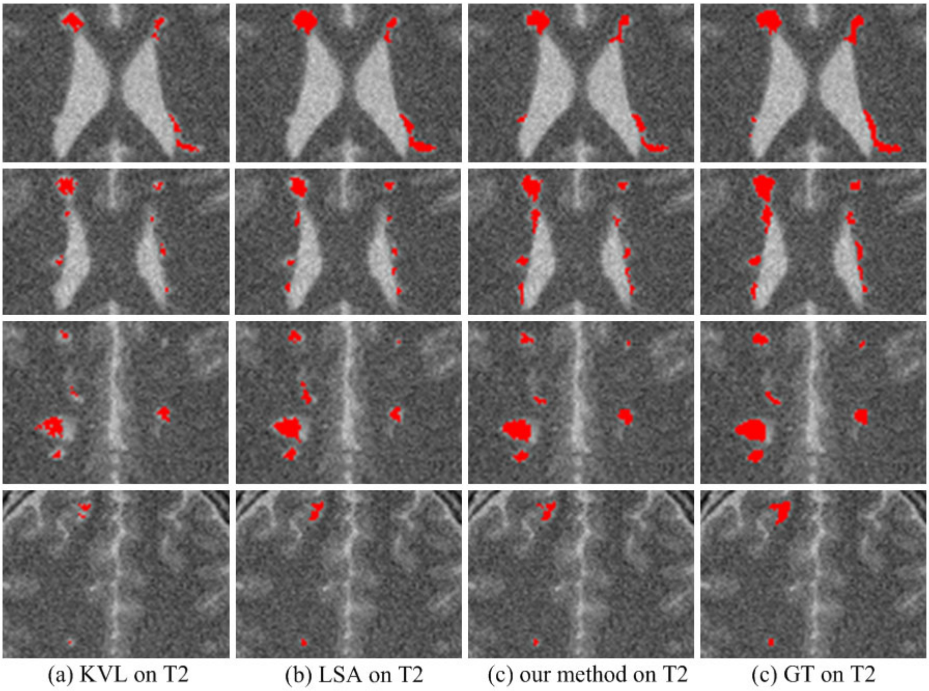

(a) KVL on T2 (b) LSA on T2 (c) our method on T2 (c) GT on T2

Fig. 6. Zoom in of segmentation results. Red: MS Lesions. The first row: slice 97. The second row: slice 100. The third row: slice 104. The fourth row: slice 111.

radius equal to 4 are used to find the WM and GM coordinates in slices 80 to 100. The average coordinates of the cluster centers are WM (119, 82) and GM (93, 91), respectively. According to the MS lesions distribution model, the distribution region can be defined in the joint histogram. The range is from 80 to 119 on T1, and from 119 to 256 on T2. Subsequently, FCM and morphology are used to remove the normal brain tissue. In the second step, false positive regions are removed. The segmentation results are convolved with a Gaussian kernel (radius equal to 5). A threshold u=30 is defined to determine whether MS lesions exist. In the third step, the minimization of the Region-Scalable energy fitting method is used to reduce false positive pixels for the selected slices with MS lesions. Finally, the false negative interested regions are found by using corresponding relationship. In Eq. 1, $\mu = 1$. In Eq. 3, $\lambda_1 = 10, \lambda_2 = 20$, the parameter σ is defined as 5. In Eq. 4, $\xi = 0.1$. The iteration termination criterion of Split Bregman Ψ is defined as 10^{-6}. Some of our results which are compared with Rouaïnia's method [2] are shown in Fig. 5. We can see that our method is more robust and deals well with high noise levels and inhomogeneous data. In the case of high density inhomogeneous data, density method based on one modality (e.g. [2]) will misclassify other tissues as MS lesions.

We also compared our method with the Leemput's method [6] (referred to as KVL) and Aït-Ali's method [7] (referred to as LSA) by using the BrainWeb [16]

Table 1. Dice index for lesion segmentation on Brainweb data for different noise levels

Algorithm	3% noise	5% noise	7% noise	9% noise
KVL [6]	0.80	0.73	0.61	0.47
LSA [7]	0.79	0.75	0.74	0.70
Our algorithm	0.81	0.79	0.77	0.74

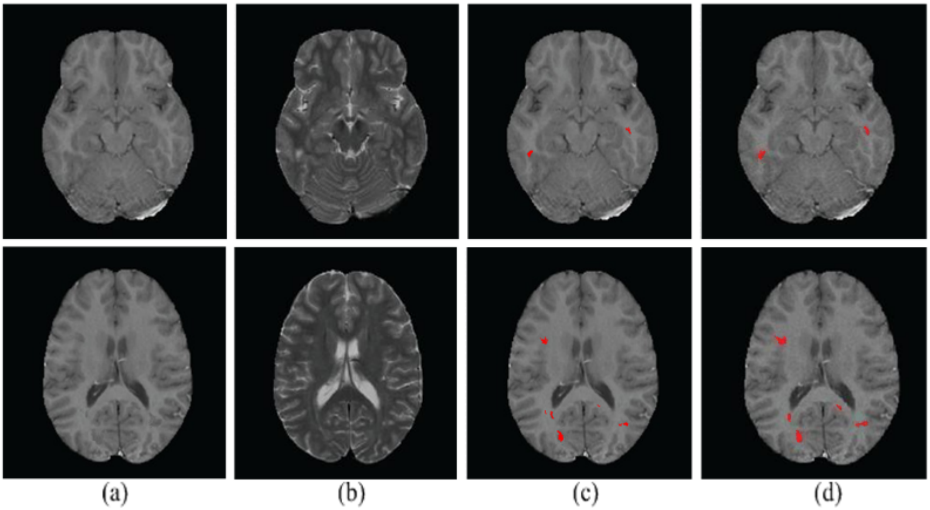


Fig. 7. Real brain images. (a) T1, (b) T2, (c) Our segmentation results (red) overlayed on T1, (d) Manual segmentation of MS lesions (red) overlayed on T1.

data. The following co-registered modalities T1, T2 and PD were used for the KVL and LSA segmentation methods as proposed in their papers. In our method, we only use the co-registered T1 and T2 volumes. Experiments were done on slices 60 to 120 which contain 93% of the lesions. The Dice Similarity Coefficient (DSC) is used to compare segmentations. Given two targets R1 and R2, the DSC is defined by:

$$DSC(R1, R2) = 2 \times (R1 \cap R2)/(R1 + R2). \tag{9}$$

where R1 and R2 denote the number of voxels assigned to the segmentation results and the ground truth, respectively. In our experiment, different levels of noise were added: 3%, 5%, 7% and 9% (one case at each noise level was used). Tab. 1 shows MS lesion Dice results for KVL, LSA and our method. In all the cases, our method shows improved results. In the KVL implementation (using the publicly available code/tool box), the statistical brain atlas of SPM99 [17] was normalized to the target brain volume images. In Fig. 6, four example slices are shown. The automatic segmentation results of the test data show improved results compared to alternative methods. However, compared with the ground truth segmentation, the lesions are slightly under-segmented. Since the intensity of MS lesions changes gradually into normal tissue, and the lesions boundary contain noise, it is still difficult to identify the optimal segmentation. We can use the above results as a good initialization for a level set method to provide a refined lesion boundary, so a structural over-segmentation and under-segmentation may not be problematic. Also we can increase the erosion parameter, the results would be improved in some cases.

Our algorithm are also tested on two real MRI volumes (512 × 512 × 512 voxels of $0.5 \times 0.5 \times 0.5\ mm^3$). The average coordinates are calculated as WM (209, 21) and GM (201, 28), respectively. The results compared with the manual segmented results by a human expert are shown in Fig. 7. Again, this shows a slight under-segmentation of MS lesions. However, as discussed in the previous paragraph, this is not seen as a significant problem.

4 Conclusion and Future Work

In this paper, we have proposed a new method which can automatically recognize and segment MS lesions. This is achieved by using a joint histogram that exploits multi-sequence information to locate the MS lesions region in each slice. This region is automatically estimated by using median filtering, class-adaptive Gauss-Markov modeling, fuzzy C-means, morphology methods. We tested our method on multi-modality images from BrainWeb. The evaluation showed that our method based on joint histograms can effectively and automatically recognize and segment MS lesions. Our method has three advantages. Firstly, it can effectively solve the grey level overlap problem between MS lesions and other tissues in the brain. Secondly, our method does not need to remove the skull and can deal with any slice in the MRI volume. Thirdly, compared with state-of-art methods, our approach is more robust and less sensitive to noise and inhomogeneous data. However, it should be clear that the developed methodology has a slight heuristic feel to it and there are a number of parameters. With respect to the latter, we have investigated ranges for most parameters which indicated robustness.

In the future, we will validate our algorithm on a larger clinical database, by comparing our segmentation results of MS lesions with ones manually segmented according to type (young, inflammatory, necrosis) and their evolution characteristics over time.

References

1. Grimaud, J., Lai, M., Thorpe, J., Adeleine, P., Wang, L., Barker, G.J., Plummer, D.L., Tofts, P.S., McDonald, W.I., Miller, D.H.: Quantification of MRI lesion load in multiple sclerosis: a comparison of three computer-assisted techniques. Magnetic Resonance Imaging 14, 495–505 (1996)
2. Rouaïnia, M., Medjram, M.S., Doghmane, N.: Brain MRI segmentation and lesions detection by EM algorithm. World Academy of Science. Engineering and Technology 24, 139–142 (2006)

3. Sajja, B.R., Datta, S., He, R., Mehta, M., Gupta, R.K., Wolinsky, J.S., Narayana, P.A.: Unified approach for multiple sclerosis lesion segmentation on brain MRI. Annales of Biomedical Engineering 34(1), 142–151 (2006)
4. Shiee, N., Bazin, P.L., Pham, D.L.: Multiple Sclerosis Lesion segmentation using statistical and topological atlases. In: Medical Image Analysis on Multiple Sclerosis(MIAMS) Workshop in MICCAI (2008)
5. Dugas-Phocion, G., Gonzalez, M.A., Lebrun, C., Chanalet, S., Bensa, C., Malandain, G., Ayache, N.: Hierarchical segmentation of multiple sclerosis lesions in multi-sequence MRI. Biomedical Imaging: Nano to Macro (2004)
6. Leemput, K.V., Maes, F., Vandermeulen, D., Colchester, A., Suetens, P.: Automated segmentation of multiple sclerosis lesions by model outlier detection. IEEE Transactions on Medical Imaging 20(8), 677–688 (2001)
7. Aït-Ali, L.S., Prima, S., Hellier, P., Carsin, B., Edan, G., Barillot, C.: STREM: A robust multidimensional parametric method to segment MS lesions in MRI. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 409–416. Springer, Heidelberg (2005)
8. Iannucci, G., Minicucci, L., Rodegher, M., Sormani, M.P., Comi, G., Filippi, M.: Correlations between clinical and MRI involvement in multiole sclerosis assessment using T1, T2 and MT histograms. Journal of the Neurological Sciences 171, 121–129 (1999)
9. Wang, W.H., Feng, Q.J., Liu, L., Chen, W.F.: Segmentation of brain MR images through class-adaptive Gauss-Markov random field model and the EM algorithm. Journal of Image and Graphics 13(3), 488–493 (2008)
10. Li, C.M., Kao, C.Y., John, C., Ding, Z.H.: Minimization of Region-Scalable Fitting Energy for Image Segmentation. IEEE Trans. Image Processing 17(10), 1940–1949 (2008)
11. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of denoising and segmentation models. SIAM Journal on Applied Mathematics 66, 1632–1648 (2006)
12. Goldstein, T., Bresson, X., Osher, S.: Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction. Journal of Scientific Computing 45, 272–293 (2010)
13. Wang, W.H., Feng, Q.J., Chen, W.F.: Active contour based on Region-scalable fitting energy. To be Presented at Chinese Journal of Computers (2011)
14. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast Global Minimization of the Active Contour/Snake Model. Journal of Mathematical Imaging and Vision 28, 151–167 (2007)
15. Yang, Y.Y., Li, C.M., Kao, C.Y., Osher, S.: Split Bregman Method for Minimization of Region-Scalable Fitting Energy for Image Segmentation. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammound, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., Avila, L. (eds.) ISVC 2010. LNCS, vol. 6454, pp. 117–128. Springer, Heidelberg (2010)
16. Brain Web. Montreal Neurological Institute, McGill University (2006), http://www.bic.mni.mcgill.ca/brainweb/
17. spm99 (2000), http://www.fil.ion.ucl.ac.uk/spm/

Surface Reconstruction of Scenes Using a Catadioptric Camera

Shuda Yu and Maxime Lhuillier

LASMEA UMR 6602, UBP/CNRS, Campus des Cézeaux, 63177 Aubière, France
http://www.lasmea.univ-bpclermont.fr/Personnel/Maxime.Lhuillier/

Abstract. This paper presents a method to reconstruct a surface from images of a scene taken by an equiangular catadioptric camera. Such a camera is convenient for several reasons: it is low cost, and almost all visible parts of the scene are projected in a single image. Firstly, the camera parameters and a sparse cloud of 3d points are simultaneously estimated. Secondly, a triangulated surface is robustly estimated from the cloud. Both steps are automatic. Experiments are provided from hundreds of photographs taken by a pedestrian. In contrast to other methods working in similar experimental conditions, ours provides a manifold surface in spite of the difficult (passive and sparse) data.

1 Introduction

The automatic 3d modeling of scenes from image sequence is still an active area of research. In our context where the scene is an environment and the view points are in the neighborhood of the ground (not aerial images), the use of a wide view field camera is a natural choice since we should reconstruct everything which is visible around the view points. There are two steps: geometry estimation and surface estimation. The former estimates the camera parameters in conjunction with a sparse cloud of 3d points of the scene surface. The camera parameters are the successive 6D poses (rotation + translation) where the images are taken, and intrinsic parameters which define the projection function of the camera. The latter estimates a list of textured triangles in 3d from the images. In the ideal case, the triangle list is a manifold surface which approximates the scene surface. "Manifold surface" means that every point in the list has a neighborhood which has the disk topology. Thus the triangle list has neither hole nor self-intersection and it partitions the 3d space in "inside" and "outside" regions.

Now, our approach is compared with previous work. An exhaustive survey is outside the paper scope and we focus on the most close approaches. In contrast to the majority of other reconstruction systems, we don't try to reconstruct a dense set of points using dense stereo methods. We think that a well chosen and sparse set of features is enough in many cases and for several applications such as interactive walkthrough and vehicle localization. We also note that only a small number of reconstruction systems use a catadioptric camera (a convex mirror of revolution mounted in front of a perspective camera) and nothing more; the

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 145–156, 2011.

most used acquisition hardware is defined by one or several perspective camera(s) pointing in several directions like the Ladybug [16].

A 3d modeling method of scene using a catadioptric camera was developed. Firstly, the camera parameters (poses and intrinsic parameters) and a sparse cloud of 3d points are estimated [11]. Then a quasi-dense reconstruction is done and approximated by triangles [12]. However, these triangles are not fully connected and the resulting 3d models have holes. Comparing with this method, ours uses the same geometry estimation but a different surface estimation which provides a manifold surface. We describes these two steps in Sections 2 and 3, respectively.

An accurate surface reconstruction method from fully calibrated camera was developed [7]. A great number of interest points are matched (in this context, a non negligible part of them have low accuracy or even are false positive). Then, a method based on 3d Delaunay of the reconstructed points and on optimization (Graph-Cut) of all point visibilities (similar to [8]) is used to obtain an approximation of the surface. Last, the surface is refined by minimizing a global correlation score in the images. In practice, results are provided on sequences with a reduced number of images (dozens of images). In this work, no information is provided on how to obtain a manifold surface: the second optimization (correlation) needs the manifold property but the first optimization (visibility) does not enforce this property. We also use a 3d Delaunay of the reconstructed points and optimize the point visibilities, but our 3d point cloud is sparser and more accurate (it is provided by bundle adjustment of Harris interest points).

A reconstruction system based on a costly hardware mounted on a car was also developed [17]. It involves several perspective cameras pointing in several directions, accurate GPS/INS (global positioning system + inertial navigation system). The approach is briefly summarized as follows. Firstly, successive poses are estimated using Kalman fusion of visual reconstruction, INS and GPS. Secondly, interest points are detected and tracked in the images; then a sparse cloud of 3d points is obtained. Third, this cloud is used to select planes in 3d, which are used to drive a denser reconstruction. Fourth, the obtained 3d depth maps are merged by blocks of consecutive images. Last, a list of triangles which approximate the dense 3d information is generated. This approach is incremental and real-time, it allows reconstruction of very long video sequences. However, the resulting surface is not manifold since the triangles are not connected. This problem could be corrected by using a merging method such as [4] followed by a marching cube [13], but this is not adequate to large scale scene since it requires a regular subdivision of space into voxels. For this reason (and other reasons mentioned in [8]), an irregular subdivision of space into tetrahedra is more interesting for large scale scene.

A few works renounce to reconstruct a dense cloud of points for 3d scene modeling. In [15] and [14], the main goal is real-time reconstruction. Both papers reconstruct a sparse cloud of points, add them in a 3d Delaunay triangulation, select the "outside" tetrahedra thanks to visibility (a tetrahedron is "outside" if it is between a 3d point and a view point where the point is seen). The remaining tetrahedra are “inside” and the surface result is the list of triangles between inside and outside tetrahedra. Both works are experimented on very simple scenes and their surfaces are not guaranteed to be manifold.

Note that a lot of Delaunay-based surface reconstruction methods exist [2] to reconstruct a (manifold) surface from an unorganized point cloud, but these methods require a dense enough cloud and ignore the visibility constraints provided by a geometry estimation step. In our context where visibility constraints are used, the list of adequate methods is reduced to [5,8,15,7,14]. Among these works, only [5] gives some details on how to obtain a manifold surface, but this is experimented on very small input cloud.

2 Geometry Estimation

Here we introduce the catadioptric camera and summarize problems to be solved in the catadioptric context: matching, geometry initialization and refinement.

2.1 Catadioptric Camera

Our low cost catadioptric camera is obtained by mounting a mirror of revolution in front of a perspective camera (the Nikon Coolpix 8700) thanks to an adapter ring. We assume that the catadioptric camera has a symmetry axis and the scene projection is between two concentric circles (Fig. 1). The camera model defines the projection function from the camera coordinate system to the image. Our model is central (all observation rays of the camera go through a point called the centre) with a general radial distortion function. This distortion function is a polynomial which maps the ray angle a (between the symmetry axis and the line between a 3d point and the centre) to the image radius r (the distance between the point projection and the circle center in the image). This model is a Taylor-based approximation which simplifies the geometry estimation.

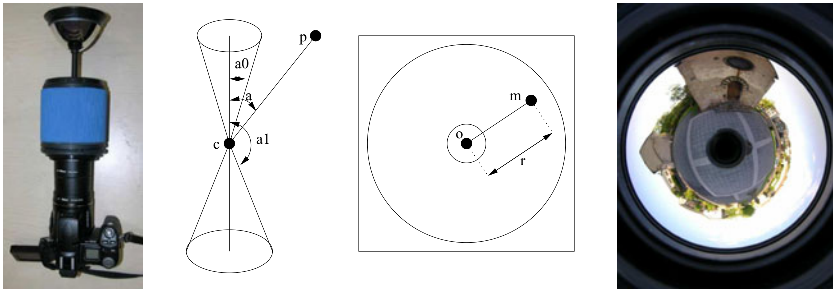


Fig. 1. From left to right: catadioptric camera, camera model (view field and image), and catadioptric image. The camera model has center **c**, symmetry axis (vertical line), view field bounded by two angles $a0$ and $a1$. Point **p** with ray angle a (such that $a0 \leq a \leq a1$) is projected to **m** with radius r between the two concentric circles.

During a sequence acquisition, the camera is hand-held and mounted on a monopod such that the symmetry axis is (roughly) vertical. The mirror [1] is designed such that the distortion function is linear (equiangular camera) with a very large view field: 360 degrees in the horizontal plane and about 40 degrees below and above. In practice, the user alternates a step forward in the scene and a shot to get a sequence of photographs. We don't use video (although the acquisition is more convenient) since the video hardware under similar view field and image quality is more costly.

2.2 Matching

Here we explain how to match image points detected in two successive images of the sequence. According to the acquisition process described in Section 2.1, the camera motion is a sequence of translations on the ground and rotations around axes which are (roughly) vertical. In this context, a high proportion of the image distortion (due to camera motion) is compensated by image rotation around the circle center. The Harris point detector is used since it is invariant to such image rotation and it has a good detection stability. We also compensate for the rotation in the neighborhood of the detected points before comparing the luminance neighborhood of two points using the ZNCC score (Zero Mean Normalized Cross Correlation). Here a list of point pairs matched in the two images is obtained. Then the majority of image pixels are progressively matched using match propagation [10]. Last the list is densified: two Harris points satisfying the propagation mapping between both images are added to the list.

2.3 Geometry Initialization

Firstly, the radial distortion function (Section 2.1) which maps the ray angle to the image radius is initialized. On the one hand, approximate values of the two angles which define the view field (above and below the horizontal plane) are provided by the mirror manufacturer. On the other hand, the two circles which bound the scene projection are detected in images. Since the two angles are mapped to the radii of the two circles, we initialize the radial distortion function by the linear function which links these data. Second, the ray directions of the matched Harris points are estimated thanks to the calibration. Third, the 2-view and 3-view geometries (camera poses and 3d points) of consecutive images pairs are robustly estimated from the matched ray directions and RANSAC applied on minimal algorithms (more details in [11]). Fourth, we estimate the whole sequence geometry (camera poses and 3d points) using adequate bundle adjustments (BAs) applied in a hierarchical framework [11]. Remind that BA is the minimization of the sum of squared reprojection errors by the (sparse) Levenberg-Marquardt method [6].

2.4 Geometry Refinement

As mentioned in Section 2.1, our camera model is an approximation. Furthermore, the two angles used for the initialization above are also approximations of the true angles (they depend on the unknown pose between mirror and the perspective camera). For these reasons, our current model should be refined. Therefore the linear radial distortion polynomial is replaced by a cubic polynomial whose the four coefficients should be estimated [11]. Then we apply a last BA to refine simultaneously all camera poses, 3d points and the four coefficients. A 2D point is involved in BA iff its reprojection error is less than a threshold (2 pixels).

3 Surface Estimation

Firstly, the link between the 3d Delaunay triangulation and the surface estimation problem is described. Then we explain how to obtain a manifold surface from the data provided by the geometry estimation in Section 2. These data are the cloud P of 3d points $\mathbf{p}_i$, the list C of view points $\mathbf{c}_j$ (3d location of images), and the visibility lists V_i of $\mathbf{p}_i$ (i.e. $\mathbf{p}_i$ is reconstructed from the view points $\mathbf{c}_j$ such that $j \in V_i$). The surface estimation has four steps: 3d Delaunay, Ray-Tracing, Manifold Extraction, and Post-Processing.

3.1 From 3d Delaunay Triangulation to Surface Estimation

Let T be the 3d Delaunay triangulation of P. Remind that T is a list of tetrahedra which "partitions" the convex hull of P such that the vertices of all tetrahedra are P. By "partitions", we means that the union of tetrahedra is the convex hull and the intersection of two tetrahedra t_0 and t_1 is either empty or a t_0 vertex or a t_0 edge or a t_0 triangle (i.e. a t_0 facet). Here are two useful properties of T [2]: (1) the edges of all tetrahedra roughly define a graph neighborhood of P in the different directions and (2) two different triangles of the tetrahedra are either disjoint, or have one vertex in common, or have two vertices (and its joining edge) in common.

Assume that P samples an unknown surface S_0. We would like to approximate S_0 by a triangle list S whose vertices are in P. The denser sampling P of S_0, the better approximation of S_0 by S. Since this approximation boils down to define connections between points of P which are neighbors on the surface, a possible approach is to search S as a subset of the facets of all tetrahedra of T. In this case, the triangles of S meet property (2) above. Now assume that (3) all vertices of S are regular. A vertex $\mathbf{p}$ of S is regular if the edges opposite to $\mathbf{p}$ in the triangles of S having $\mathbf{p}$ as vertex form a simple polygon (Fig. 2). S is a manifold surface since it meets (2) and (3).

3.2 3d Delaunay

As suggested in Section 3.1, the first step is the 3d Delaunay triangulation of P. Remind that $\mathbf{p}_i$ has very bad accuracy if it is reconstructed in degenerate configuration: if $\mathbf{p}_i$ and all $\mathbf{c}_j, j \in V_i$ are (nearly) collinear [6]. This case occurs in part of the camera trajectory which is a straight line and if points reconstructed

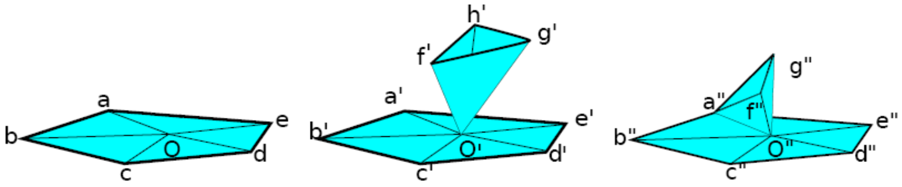


Fig. 2. O is regular since the edges opposite to O define a simple polygon $abcdea$. O' and O'' are not regular since polygons $a'b'c'd'e'a' - f'g'h'f'$ and $a''b''c''d''e''a''f''g''a''$ are not simple (the former is not connected, the latter has multiple vertex a'').

from this part are in the "neighborhood" of the straight line. Thus, P is filtered before Delaunay: we remove $\mathbf{p}_i$ from P if all angles $\mathbf{c}_j\widehat{\mathbf{p}_i}\mathbf{c}_k$ $(j, k \in V_i)$ are less than a threshold ϵ. It is also possible to improve the final S by adding "artificial points" in P which have empty visibility lists (more details are given in Section 4).

3.3 Ray-Tracing

Now we use the visibility information to segment T in "outside" and "inside" tetrahedra. Note that a tetrahedron is 100% inside or 100% outside since our target surface is included in the facets of all tetrahedra of the partition T. A tetrahedron which is intersected by ray (line segment) $\mathbf{c}_j\mathbf{p}_i, j \in V_i$ is outside since point $\mathbf{p}_i$ is visible from view point $\mathbf{c}_j$. In practice, all tetrahedra are initialized inside and we apply ray-tracing for each available ray to force outside all tetrahedra intersected by the ray.

In our catadioptric context where points are reconstructed in almost all directions around view point, the view points are in the convex hull of P. This implies that the region outside the convex hull of P can not be intersected by ray and this region is classified inside. For implementation convenience [3] (obtain a complete graph of tetrahedra), tetrahedra are added to establish connections between the "infinite point" and the facets of the convex hull. These tetrahedra are classified inside although they have no physical volume in 3d.

3.4 Manifold Extraction by Region Growing

At first glance, S could be defined by the list of triangles separating the inside and the outside tetrahedra. Unfortunately, the experiment shows that S is not manifold because it has vertices which are not regular (Section 3.1).

It is suitable to change the outside and inside definitions for the manifold extraction: *outside* becomes *intersected*, *inside* becomes *non-intersected*, and now the *outside* tetrahedra form a sub-list O of *intersected* tetrahedra such that its border S (a list of triangles) is manifold. The tetrahedra which are not *outside* are *inside*. These new definitions apply in the sequel of the paper.

Growing one tetrahedron at once. The manifold extraction is a region growing process: we progressively add to O new tetrahedra, which were inside and intersected, such that the border S of O is continually manifold. Region growing is a convenient way to guarantee the manifold constraint, but the final O depends on the insertion order of the tetrahedra in O. Indeed, a tetrahedron in O which has vertices in S enforces manifold constraints on these vertices, which are shared by other tetrahedra which are not (or not yet) in O.

To reduce the manifold constraints, we choose the new tetrahedron such that it has at least one facet included in S (i.e. it is in the immediate neighborhood of O). We also choose a priority score for each intersected tetrahedron to define the insertion order: the number of rays which intersect the tetrahedron. The tetrahedra in the neighborhood of O are stored in a priority queue for fast selection of the tetrahedron with the largest priority.

Growing several tetrahedra at once. Note that the region growing above does not change the initial topology of O. Here O is initialized by the tetrahedron with the largest score and it has the ball topology. This is a problem if the true outside space has not the ball topology, e.g. if the camera trajectory contains closed loop(s) around building(s). In the simplest case of one loop, the true outside space has the toroid topology and the computed outside space O can not close the loop (left of Fig. 3).

We correct this kind of problem with the following region growing. Firstly, we find a vertex on S such that all inside tetrahedra which have vertex S are intersected. Then, we force all these tetrahedra to outside. If all vertices of these tetrahedra are regular, the region growing is successful and O is increased. In the other case, we restore these tetrahedra to inside. In practice, we alternate "one tetrahedron at once" and "several tetrahedra at once" region growings until no tetrahedron may be added in O.

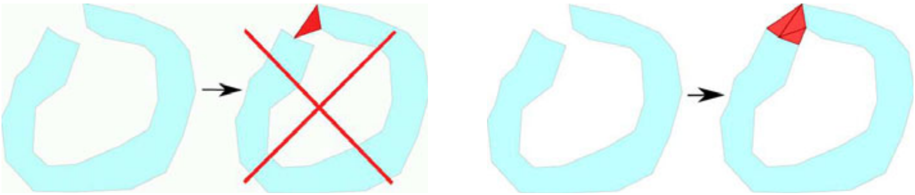


Fig. 3. Left: adding one tetrahedron at once in O (in light blue) can not close the loop due to non regular vertex. Right: adding several tetrahedra at once close the loop.

3.5 Post-Processing

Although the surface S provided by the previous steps is manifold, it has several weaknesses which are easily noticed during visualization. Now we list these weaknesses and explain how to remove (or reduce) them using prior knowledge of the scene.

A "peak" is a vertex $\mathbf{p}_i$ on S such that the ring of its adjacent triangles in S defines a solid angle w which is too small (or too large) to be physically plausible, i.e. $w < w_0$ or $w > 4\pi - w_0$ where w_0 is a threshold. We try to remove peak $\mathbf{p}_i$ from S as follows. We consider the acute side of the peak (inside or outside tetrahedra) and reverse its status (inside becomes outside, and vice versa). The removal is successful if all vertices of the reversed tetrahedra are regular. Otherwise, the reversed tetrahedra are reversed a second time to go back in the previous configuration and we try to remove an other peak. In practice, we go through the list of S vertices several times to detect and remove the peaks.

The surface S is smoothed to reduce the reconstruction noise. Let $\mathbf{p}$ be the concatenation vector of all $\mathbf{p}_i$ in S. We apply a mesh smoothing filter $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is a discrete laplacian defined on the mesh vertices [18].

Up to now, S is closed and contains triangles which correspond to the sky (assuming outdoor image sequence). These triangles should be removed since they do not approximate a real surface. They also complicate the visualization of the 3d model from a bird's-eye view. Firstly, the upward vertical direction $\mathbf{v}$ is robustly estimated assuming that the camera motion is (roughly) on a horizontal plane. Secondly, we consider open rectangles defined by the finite edge $\mathbf{c}_i\mathbf{c}_{i+1}$ and the two infinite edges (half lines) starting from $\mathbf{c}_i$ (or $\mathbf{c}_{i+1}$) with direction $\mathbf{v}$. A triangle of S which is intersected by an open rectangle is a sky triangle and is removed from S. Now S has hole in the sky. Lastly, the hole is increased by propagating its border from triangle to triangle such that the angle between triangle normal (oriented from outside to inside) and $\mathbf{v}$ is less than threshold α_0.

Last, the texture should be defined for each triangle of S. We use a simplified version of [9], which gets "as is" the texture of a well chosen view point $\mathbf{c}_j$ for each triangle t of S. In our case where the image sequence has hundreds of images (not dozens), we pre-select a list of candidate $\mathbf{c}_j$ for t as follows: t is entirely projected in the $\mathbf{c}_j$ image (large t are splitted), t is not occluded by an other triangle of S, $\mathbf{c}_j$ provides one of the k-largest solid angle for triangle t.

4 Experiments

The acquisition set up is described at the end of Section 2.1. Our sequence has 208 images taken along a full turn around a church. The trajectory length is about $(25 \pm 5\text{cm}) \times 208 = 52 \pm 10\text{m}$ (the exact step lengths between consecutive images are unknown). The radii of large and small circles of the catadioptric images are 563 and 116 pixels.

The geometry estimation step (Section 2) reconstructs 76033 3d points from 477744 2D points (inliers). The RMS error of the final bundle adjustment is 0.74 pixels. The estimated view field angles (Fig. 1) are $a0 = 41.5$ and $a1 = 141.7$ degrees; the angles provided by the mirror manufacturer are $a0 = 40$ and $a1 = 140$ degrees. Fig. 4 shows images of the sequence and the result of this step.

Then the surface is estimated (Section 3) using $\epsilon = 5$ degrees, $w_0 = \pi/2$ steradians, $\alpha_0 = 45$ degrees, and $k = 10$. Fig. 5 explains the advantages of the steps of our method. Row 1 shows that the reconstructed points can not be used "as is" as vertices of the surface. The surface fairing (peak removal and surface smoothing) is necessary. Remind that a 3d point may be inaccurate for several reasons: (1) if

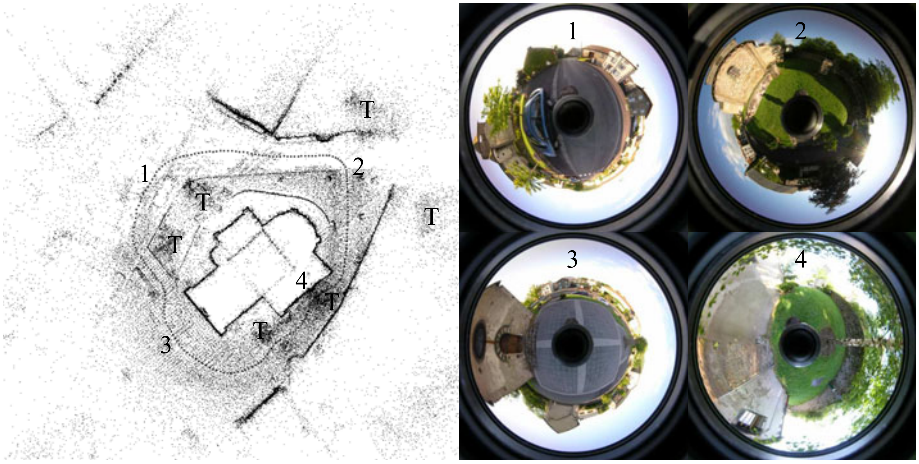


Fig. 4. Top view of the geometry estimation result (left) and four catadioptric images (right) for the church sequence. The top view includes 76033 points, 208 poses around the church, numbers for the locations of the four images, "T" for tree locations around the camera trajectory, and numbers for the image locations.

it has large depth (it is reconstructed from far camera poses), (2) if it has small depth (the central approximation of the true non-central camera model provides bad accuracy in our context) and (3) image noise. Row 2 shows top views of the surface before (left) and after (right) the sky removal. On the left, the surface is closed and we see the sky triangles. On the right, the surface can be visualized at a bird's eye view. Note that the current version of the sky removal process is very simple; it should be improved to remove large triangles on the top of the model (including those which connect trees and the church). The left of row 3 shows that the surface forms a blind alley if the "several tetrahedra at once" region growing is not used (the pedestrian can not go from location 3 to location 4, see Fig. 4). The right of row 3 shows that the loop is closed around the church if this region growing is used (the pedestrian can go from location 3 to location 4). Row 4 shows that the "several tetrahedra at once" region growing is also very useful to improve the outside space estimated by the "one tetrahedron at once" region growing. However, both region growings are not enough to avoid problems as the one shown on the left of row 5: there is an ark which connects a vertical surface to the ground, although all tetrahedra which define the ark are intersected by rays. We tried to solve this problem by changing the priority score of the tetrahedra, but arks always appear somewhere. Here we greatly reduce the risk of arks thanks to a simple method. In the 3d Delaunay step, "artificial points" are added in P such that the long tetrahedra potentially involved in arks are splitted in smaller tetrahedra. The artificial points are added in a critical region for visualization: the immediate neighborhood of the camera trajectory. Technical

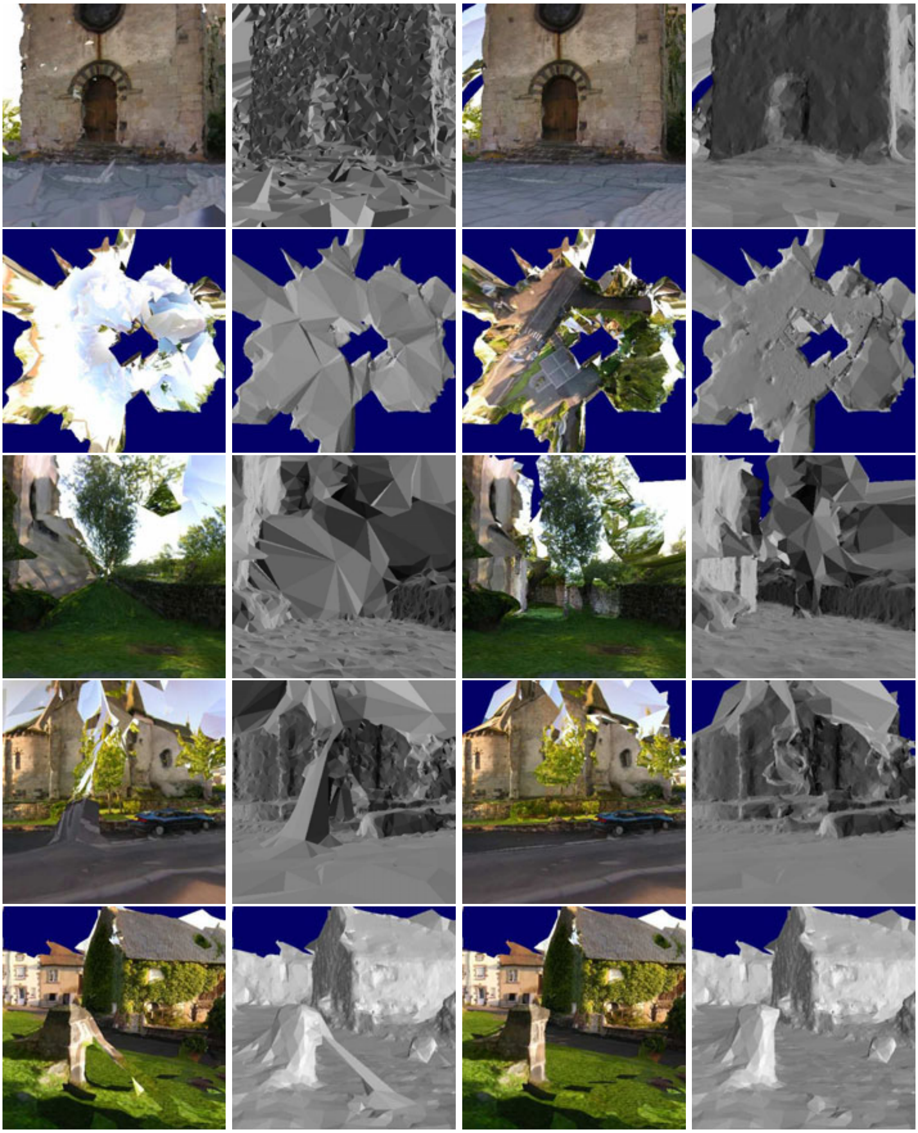


Fig. 5. Row 1: local view of the church 3d model without (left) or with (right) peak removal and surface smoothing (Section 3.5). Row 2: top view without (left) or with (right) the sky removal (Section 3.5). Rows 3 and 4: without (left) or with (right) the "several tetrahedra at once" region growing (Section 3.4). Row 5: without (left) or with (right) the artificial points (Section 3.2). In all cases, gray levels encode the triangle normals.

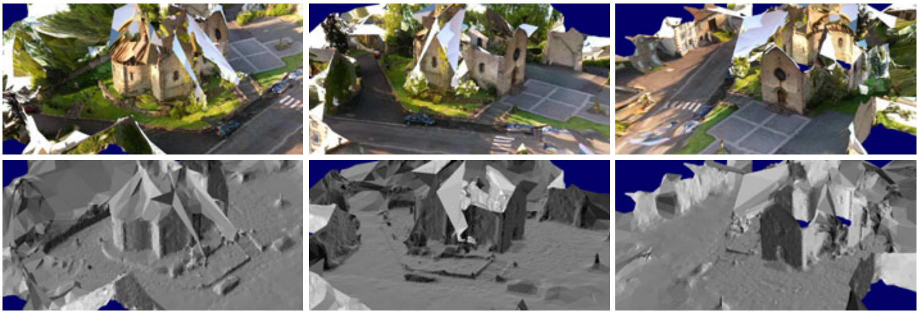


Fig. 6. Bird's-eye views of the church 3d model obtained with the complete method

details are described in the appendix for the paper clarity. The right of row 5 shows that ark is removed thanks to the artificial points. Fig. 6 shows other views of the church obtained with the complete method.

The 3d Delaunay has 59780 vertices and is reconstructed using the incremental 3d Delaunay of CGAL [3]. During the ray tracing step, 191947 tetrahedra are intersected by 398956 rays. The manifold extraction step by region growing provides a surface with 94228 triangles and 162174 outside tetrahedra. The ratio between outside and intersected tetrahedra is 86% (the ideal value is 100%). Lastly, 898 sky triangles are removed. The surface estimation (without texturing) takes about 30 seconds on a Core 2 Duo E8500 at 3.16 GHz. A complete walkthrough around the church is provided in a **mpeg video** available at www.lasmea.univ-bpclermont.fr/Personnel/Maxime.Lhuillier. This video is cyclic and includes the sky since the 3d model is viewed from a pedestrian's-eye view

5 Conclusion

The proposed method has two steps: geometry estimation and surface estimation. We briefly summarize the former and focus on the latter. Our results contrast with the previous ones since we are able to provide manifold surface (up to sky removal) from a reconstructed sparse cloud of points (instead of dense) in non trivial cases. The current system is able to reconstruct the main components of outdoor scene (ground, buildings, dense vegetation) and allows interactive walkthrough. Technical improvements are possible for several steps (surface fairing, sky removal and texture mapping). Future work includes the integration of edges in the 3d model, a real-time version of the surface estimation step, and a specific process for thin structures such as trunks and electric posts.

Appendix

The artificial points have empty visibility lists, their number is equal to 0.5% of the number of the reconstructed points, and they are randomly and uniformly added in the immediate neighborhood N of the camera trajectory. We define N by the union of half-balls (southern hemispheres such that the north direction is $\mathbf{v}$) centered on all $\mathbf{c}_j$ with radius $r = 10\ \text{mean}_j\ ||\mathbf{c}_{j+1} - \mathbf{c}_j||$.

References

1. http://www.kaidan.com
2. Cazals, F., Giesen, J.: Delaunay triangulation based surface reconstruction: ideas and algorithms. INRIA Technical Report 5394 (2004)
3. http://www.cgal.org
4. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH (1996)
5. Faugeras, O., Le Bras-Mehlman, E., Boissonnat, J.D.: Representing stereo data with the delaunay triangulation. Artificial Intelligence, 41–47 (1990)
6. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambridge (2000)
7. Hiep, V.H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo
8. Labatut, P., Pons, J.P., Keriven, R.: Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and Graph-Cuts. In: The International Conference on Computer Vision (2007)
9. Lempitsky, V., Ivanov, D.: Seamless Mosaicing of Image-Based Texture Maps. In: The Conference on Computer Vision and Pattern Recognition (2007)
10. Lhuillier, M., Quan, L.: Match propagation for image based modeling and rendering. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(8), 1140–1146 (2002)
11. Lhuillier, M.: Automatic scene structure and camera motion using a catadioptric system. Computer Vision and Image Understanding 109(2), 186–203 (2008)
12. Lhuillier, M.: A generic error model and its application to automatic 3d modeling of scenes using a catadioptric camera. International Journal of Computer Vision 91(2), 175–199 (2011)
13. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3d surface construction algorithm. Computer Graphics 21, 163–169 (1987)
14. Lovi, D., Birkbeck, N., Cobzas, D., Jagersand, M.: Incremental free-space carving for real-time 3d reconstruction. In: 3DPVT 2010 (2010)
15. Pan, Q., Reitmayr, G., Drummond, T.: ProFORMA: probabilistic feature-based on-line rapid model acquisition. In: The British Machine Vision Conference (2009)
16. http://www.ptgrey.com
17. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3D reconstruction from video. International Journal of Computer Vision 78(2), 143–167 (2008)
18. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH (1995)

Creating Chinkin Works in the Virtual Space

Shinji Mizuno

Faculty of Information Science, Aichi Institute of Technology
1247, Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392 Japan
s_mizuno@aitech.ac.jp

Abstract. In this paper, a method to enable people to experience "Chinkin" in the virtual space is introduced. Chinkin is a traditional artistic technique to draw designs on lacquer ware with fine lines and dots, and it was widely used to produce Daimyo's utensils in the Edo period during the 17th and the 19th century in Japan. The author develops a virtual Chinkin system based on the virtual sculpting method which is an interactive CG creating method developed by the author. Chinkin consists of some processes and each process is realized by virtual carving: an interactive deformation of solids, and virtual painting: an interactive generation of a 3D texture map. In this system, the user can experience each process of the Chinkin technique in the virtual space and can create virtual sculptures ornamented with Chinkin designs as CG. Both the operation and the result are similar to the real ones and it would be useful for introduction, education and preservation of the traditional Chinkin technique at museum and school.

1 Introduction

Computer technology has been progressed remarkably and it is widely used in many fields. Recently the field of heritage and museum is very interested in using computer technologies. The most principal duty of former museums for visitors is just to exhibit their collections and tell their brief background information. However, recent museums are expected to tell more background and related information of the collections and they have to treat huge information. Thus some museums start to use computer graphics (CG), virtual reality (VR), network technologies and robotics for dissemination and interaction about their collections, and many researches for digital museums are studied, such as a digital display case for museums [1], a museum guide system providing visual and auditory feedback via a sensor board [2], and a museum tour guide robot [3]. Information techniques are also used for digital archives of artistic works, artistic techniques, and heritage recently [4, 5].

One of the important duties of museums is to tell traditional and artistic techniques of creating museum arts and crafts. Real experience of such art-creating techniques is a good way to learn, but it is usually difficult to prepare places, tools, and guides for those techniques. Some of art-creating techniques use knives and chemicals, and they would be dangerous. One of the solutions for this problem is virtual experience of art-creating techniques. Interactive CG creating methods such as sketch-based operation is suitable for these purposes. A lot of methods to simulate drawing materials such as

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 157–168, 2011.

brushes and pencils to create 2D CG are developed [6, 7]. Some art-creating techniques such as clay work and embossing are simulated to realize interactive 3D CG creating system [8, 9, 10]. The operations and the results are similar to those of the real arts in those CG systems and the user could have experience of art-creating techniques in the virtual space, and they are useful for education, and preservation of the art creating technique at schools and museums. As one of these CG creating methods, the author has taken wooden sculpting and printing, and has developed a CG creating system [11, 12, 13]. In this system, the user can experience virtual wooden carving and woodblock printing, and can create works as 2D and 3D CG interactively.

The author is doing research on using information technology such as CG and Web for the Tokugawa Art Museum with other researchers and curators of the museum. The Tokugawa Art Museum is famous as a large Japanese traditional art collection created in the Edo period during the 17th and the 19th century. The principal collections of this museum are the Daimyo's utensils, such as furniture and dinner sets. Many of the Daimyo's utensils are lacquered and ornamented with beautiful designs. They are often produced with "Chinkin" and "Makie" techniques, and many visitors are interested in those traditional Japanese artistic techniques.

In this paper, a method to enable people to experience Chinkin in the virtual space is proposed. This method is implemented on the virtual sculpting system: an interactive CG creating system developed by the author, and a virtual Chinkin system is developed. The user can experience the Chinkin technique in the virtual space and can create works ornamented with Chinkin as 3D CG interactively. Each process of Chinkin is realized by improvement of virtual carving and virtual painting methods which the author has developed for the virtual sculpting system. In the proposed method, both the operation and the works created by the system are similar to the real ones and it would be useful to introduce this traditional art technique in the museum.

2 Overview of the Real Chinkin Technique

Lacquer ware is also called "Japan ware", and it is widely known as a traditional craft in Japan and some Asian countries [14]. Typical lacquer wares are furniture, dining utensils or sculptures made of wood, and they have been produced in many places in Japan. A type of lacquer ware: "Wajima-nuri" crafted around Wajima city in Japan is renowned for its beautiful lacquered surfaces and the ornamental designs drawn on them.

The ornamental design of Wajima-nuri is often drawn with fine gold lines or dots. This design is usually created with a "Chinkin" technique. Chinkin is a traditional artistic technique in Japan and some Asian countries to draw designs on lacquered surfaces as shown in Fig. 1.

The process of the Chinkin technique is as follows.

1. The design is carved into the lacquered surface with a very sharp chisel or a needle as shown in Fig. 2(a). This process is called "Subori".
2. Lacquer is inlayed into the carved track for gluing.

3. Gold powder or leaf is put on the lacquered surface as shown in Fig. 2(b) and the piece is placed in a damp environment. The gold powder sticks to the applied lacquer. This process is called "Kin-ire".
4. The lacquered surface is wiped with a cloth, the excess gold is wiped off, and only the gold in the carved track is left. The design comes out with gold lines and dots as shown in Fig. 2(c) and Fig. 2(d). This process is called "Shiage".

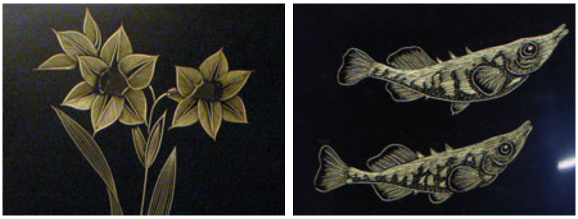
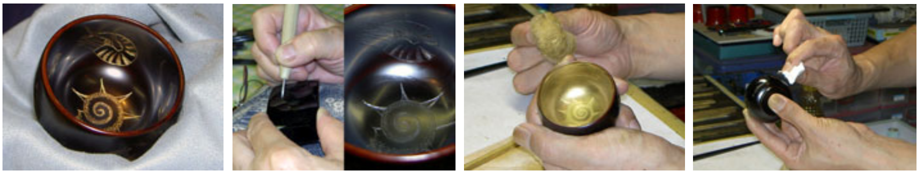


Fig. 1. Works of Chinkin



(a) "Subori" (b) "Kin-ire" (c) "Shiage" (d) A work with Chinkin

Fig. 2. The process of creating a Chinkin work

In creating Chinkin, it is necessary to carve a lacquered surface with a chisel first, so a failure is not allowed and elaborate skills are required. In addition, lacquer might cause a skin rash. So there are many people who are interested in creating Chinkin works, but it is not so easy for them to experience Chinkin actually.

3 Outline of Virtual Sculpting

The method for virtual Chinkin is realized based on the virtual sculpting method developed by the author. Virtual sculpting is an interactive 3D CG creation method based on carving and painting operation implemented with a pressure sensitive pen and a LCD display. The user can create a virtual solid work as a 3D CG by carving and painting a virtual object interactively.

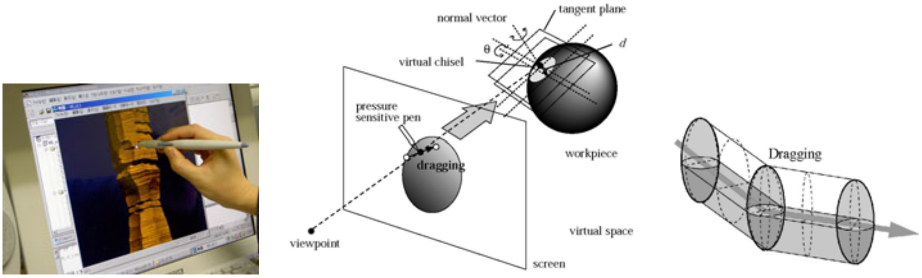
In virtual Chinkin creation, a lacquered virtual sculpture is ornamented by a virtual Chinkin technique: Subori, Kin-ire, and Shiage in the same way as a real Chinkin creation. The virtual Subori process is realized based on the virtual carving method, and the virtual Kin-ire process and the virtual Shiage process are realized based on the virtual painting method.

3.1 Virtual Carving

In virtual carving, an original workpiece and virtual chisels are prepared. The original workpiece is polyhedron by curved surfaces and the shape is expressed as a CSG (Constructive Solid Geometry) with quadric surfaces. Each virtual chisel is defined by a cube, an ellipsoid, a cylinder, and combinations of them.

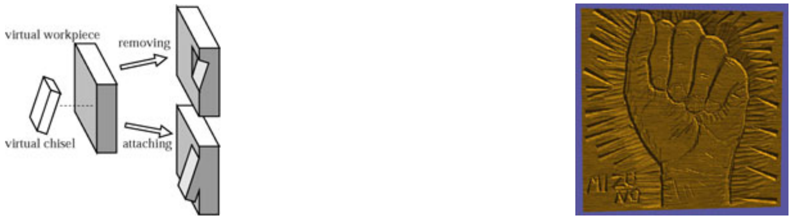
The user operates a virtual chisel on the surface of the virtual workpiece with a pressure sensitive pen (Fig. 3(a)). When the user drag a pen on a display, the position and the pressure of the operation decide the position of the chisel in the virtual space (Fig. 3(b)). The tilt of the virtual chisel is decided by the tilt of the pen and the transition of the pressure of operation. The carving track is created by combining the shape of virtual chisels automatically (Fig. 3(c)). The shape and the depth of carving track would be changed according to the operation of the pen, and the user can experience realistic virtual carving operation.

The surface is removed or attached by the shapes of chisels and deformed immediately (Fig. 4(a)). By performing these operations repeatedly, the user can create a virtual sculpture (Fig. 4(b)). The Shape of a virtual sculpture is also expressed by a CSG expression with quadric surfaces.



(a) Carving with a pen (b) Placing a virtual chisel (c) Creating a carving track

Fig. 3. Outline of virtual carving



(a) Removing/attaching a virtual chisel (b) An example of a virtual sculpture

Fig. 4. Deformation of a solid object by carving operations

In this system, lists of intersecting points are used to deform the workpiece expressed with CSG in real time [11]. A list of intersecting point is generated for each viewing line correspondence to all pixels of the screen. Each list stores all intersecting points of the viewing line with the surface of the workpiece, and arranges them in order of distance from the viewpoint. The head of each list is a visible intersecting point for each viewing line and used to compute the luminance and render an image. Lists for an original workpiece are generated first, and redrawing after each carving operation is performed by renewing the lists. Renewing process is enough fast and suitable for interactive deformation. The lists are also renewed when the viewpoint is changed by replaying the carving record from the beginning.

3.2 Virtual Painting

The user can paint ink on a virtual sculpture with a virtual brush directly. This is realized by generating and renewing a 3D texture map with painting operation. Some methods to create texture by painting operation have been developed [15]. The virtual painting method in this paper pays attention to small irregularities on the surface of a virtual sculpture and contact with a virtual pen. This method is similar to the way to check the local accessibility of the surface [16], and the similar effect is realized in some commercial CG software such as cavity masking of Zbrush [17]. The proposed method in this paper checks collision points between the surface of a virtual sculpture and a virtual brush and renew the 3D texture map at the same time in once painting operation. This method is used for Kin-ire and Shiage process in virtual Chinkin.

The tip of the virtual brush is defined by a sphere (the radius: *r*) as shown in Fig. 5. When the user operates a virtual brush with a pressure sensitive pen, a painting area on the surface of the virtual sculpture is decided and the system rolls the sphere of the virtual brush in the area.

The collision points between the sphere and the surface of the sculpture are calculated by using points of the surface stored in the lists of intersecting points. Fig. 6(a) shows the decision process of the contact points. The center point of the sphere is put on one viewing line first and it moves on the viewing line. Then the distance between the center point of the sphere and each intersecting point is calculated, and the position of the sphere is decided. Intersecting points near the surface of the sphere (the distance <*d*) are considered as contact points.

This process is done on every viewing line in the painting area. The collision points are judged as painted, corresponding pixels of the 3D texture map are renewed immediately, and the image of a painted virtual sculpture is synthesized.

The size of the virtual brush: the painting area and the radius of the sphere of the tip changes depending on the pressure of the operation, so the painting result would be changed according to the operation and the shape of the virtual sculpture as shown in Fig. 6(b). In this virtual painting method, tracks carved by a sharp chisel would not be painted as a real painting process (Fig. 5, Fig. 6).

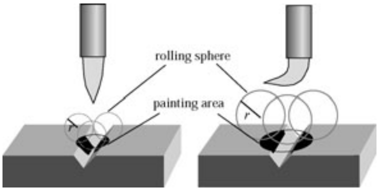
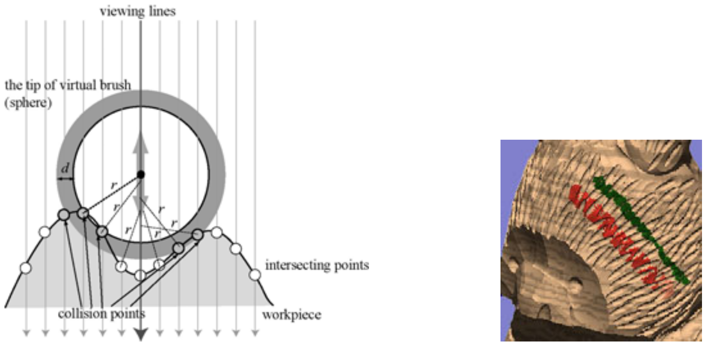


Fig. 5. A method for virtual painting



(a) The decision process of collision points

(b) Painting a virtual sculpture

Fig. 6. The decision process of collision points and an example of virtual painting result

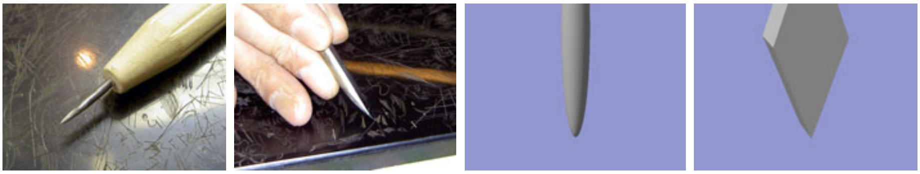
4 Implementing Virtual Chinkin

In this paper, Subori, Kin-ire, and Shiage processes in the Chinkin technique are studied and implemented on the system. Each technique of Chinkin is realized in the virtual space based on the virtual sculpting method. A carving process, a painting process, and a 3D texture map of the virtual sculpting system are improved for the virtual Chinkin method.

4.1 The Subori Process

In the real Subori process, a craftsman carves designs into lacquered surfaces with a shape chisel or a needle (Fig. 2(a)), and it is much the same as wooden carving process.

In the virtual Subori, a virtual lacquered sculpture is prepared in the system. The shapes of a chisel and a needle for Subori could be expressed as a triangle and an ellipsoid respectively (Fig. 7(a)(b)). The same shapes of virtual chisels are prepared (Fig. 7(c)(d)), and the user carves the surface of a virtual lacquered sculpture with virtual chisels in first step of the virtual Chinkin creation.



(a) A real needle (b) A real chisel (c) A virtual needle (d) A virtual chisel

Fig. 7. Chisels for "Subori" (real and virtual)

4.2 The Kin-Ire Process

The virtual Kin-ire process and the virtual Shiage process are realized by improving a 3D texture map and the painting process of the virtual sculpting. In the former virtual sculpting system, a 3D texture map consists of color values (RGB) and a moisture value, which expresses virtual ink and mainly used for creating virtual wooden prints [12]. In the Chinkin technique, a workpiece is painted both with lacquer and gold powder, and the properties of reflection of two materials are different very much. So, in this study the coefficient of specular reflection is added to the elements of the 3D texture map.

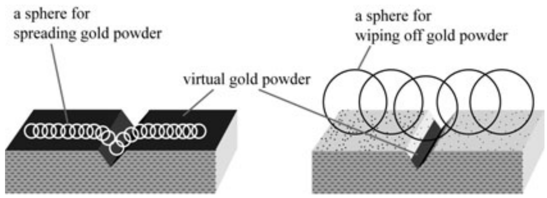
In the real Kin-ire process, the craftsman spread gold powder on the lacquered surface with a brush (Fig. 2(b)). The behavior of the gold powder is similar to ink, and the operation is similar to painting with a brush, so the virtual Kin-ire process is realized by improving the virtual painting method. When the user operates a virtual brush, the points where the brush reaches are considered that the gold powder is stuck to and the corresponding pixels of the 3D texture map are changed from lacquer to gold.

In the Kin-ire process, the gold powder sticks not only on the lacquered surface but also inside of the track carved by a chisel. To realize that in virtual Chinkin, the size of a sphere which expresses the tip of the virtual brush is determined small independently of the pressure of operation enough to go into the track (Fig. 8(a)).

4.3 The Shiage Process

In the real Shiage process, the craftsman wipes the gold powder off from the lacquered surface with a cloth, and only the gold powder in carved tracks is left because the gold powder in the carved tracks is adhered with lacquer and the cloth does not reach inside of the tracks (Fig. 2(c)(d)).

The virtual Shiage process is also realized by improving the virtual painting method. The cloth could be considered as a kind of a painting brush. The cloth would not reach inside of the carved tracks, so the sphere of the virtual cloth is determined large independently of the pressure of operation not to go into the track (Fig. 8(b)). The user operates a virtual cloth (brush), and the points where the cloth reaches are considered that the gold powder is wiped off and the corresponding pixels of the 3D texture map are changed from gold to lacquer.



(a)Virtual "Kin-ire" process (b)Virtual "Shiage" process

Fig. 8. "Kin-ire" and "Shiage" in virtual Chinkin

5 Point-Based Rendering for Virtual Sculptures

In the virtual sculpting method, lists of intersecting points are constructed for each viewing line as mentioned in Chapter 3, and they are used for interactive deformation and image rendering of a virtual sculpture. Renewing process is enough fast for each carving operation in real time. However it is necessary to reconstruct whole lists to change a viewpoint by replaying carving record from the beginning.

To change a viewpoint in real time, a point-based rendering method is adopted in the system. Point-based rendering is a technique that renders objects as a set of points and it is often used to render huge data obtained by a 3D scanner from a wide scene [18]. In the virtual sculpting method, the lists of intersecting points store whole intersecting points of viewing lines with the virtual sculpture correspondence to all pixels of the screen. All points are on the surface of the virtual sculpture and they are useful for point-based rendering in moving a viewpoint.

As far as seeing from the original viewpoint, all intersecting points look perfectly adjacent each other and the rendered image is equal to an image rendered by the former method (Fig. 9(a)). However, when the viewpoint is changed, holes between points would be appeared (Fig. 9(b)(c)). To fill such holes in point-based rendering method, a disk is often used to render each point [19]. This method is effective if points are distributed uniformly. However, points stored in the lists of intersecting points are generated by projection from a viewpoint, so densities of points at a surface of the virtual sculpture would be changed according to the angle between normal vectors of the surface and viewing lines.

Thus line segments are used to render points in this paper. In the lists of intersecting points, each two points always make a pair and a segment line between them shows inside of the virtual sculpture. The color of each segment line is decided to interpolate colors of two points linearly. Half pairs of points are rendered as segment lines and other points are rendered as points.

A virtual sculpture is rendered with the point-based rendering method during moving a viewpoint, and when a new viewpoint is fixed, lists of intersecting points at a new viewpoint are reconstructed to prepare additional deformation by carving operation, and a fine image is at a new viewpoint rendered by the former method.

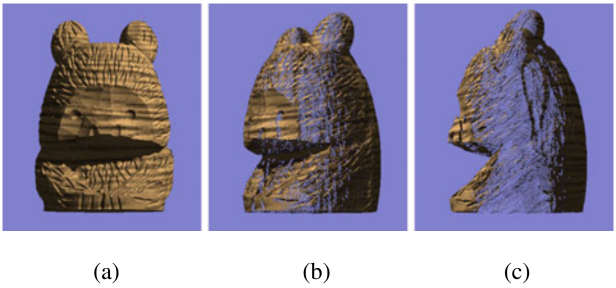


Fig. 9. Rendering a virtual sculpture only with a set of points. A virtual sculpture is seen from an original viewpoint (a), seen from an angle of 45 degrees (b), and seen from an angle of 90 degrees (c).

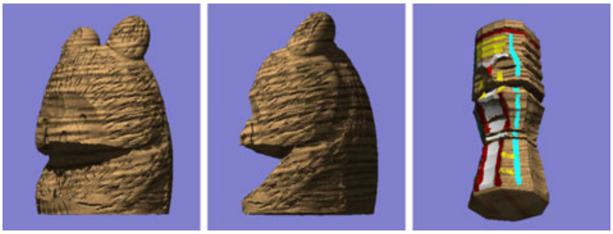
5 Experiments

5.1 Implementation

The proposed methods are implemented on a Windows PC (CPU: Intel Core2 Extreme 3.2GHz, memory size: 2GB, Graphics board: GeForce 9800GTX, 512MB), and a LCD pen display (Wacom Cintique), and a prototype CG system was built. The resolution of the window for an image is 512x512 (pixels), and the resolution of a 3D texture map is 512x512x512 (pixels). The resolution of the 3D texture map is limited by the memory size of the PC.

5.2 Results

Images in Fig. 10 are rendered as the proposed method using points and segment lines. About 210,000 points and 50,000 segment lines are used for the image, and the computation time is about 30(fps). Line segments could interpolate each pair of points and fill holes. Some details of the surface are lost by the interpolation. However this method is used only in moving a viewpoint and a fine image generated when a new viewpoint is fixed, so the quality is enough fine to see the shape of a virtual sculpture in moving a viewpoint.

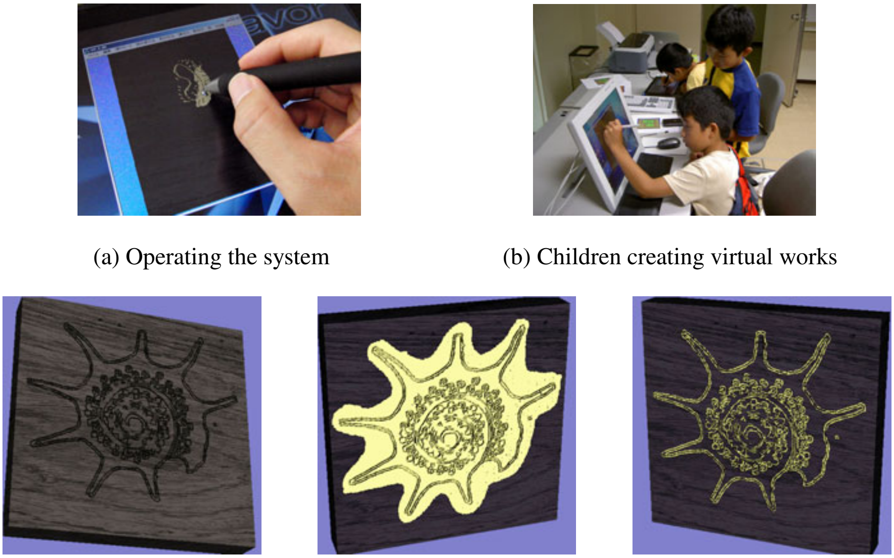


(a) (b) (c)

Fig. 10. Rendering virtual sculptures with points and line segments. A virtual sculpture is seen from an angle of 45 degrees (a), seen from an angle of 90 degrees (b), and another painted sculpture is seen from an angle of 45 degrees (c).

Fig. 11 shows the virtual Chinkin technique. The user uses a pressure sensitive pen for each process of virtual Chinkin as shown in Fig. 11(a). Fig. 11(b) shows a virtual lacquered board after the Subori process into which designs are carved with a virtual chisel. The user could carve the designs into the virtual board with a pressure sensitive pen interactively. Fig. 11(c) shows the virtual board after the Kin-ire process. The user could operate a virtual brush with a pen, and gold powder is stuck both on the surface and in the carved tracks. Fig. 11(d) shows the finished work after the Shiage process. The user could operate a virtual cloth with a pen to wipe off the gold powder, and only the gold powder in the carved track is left and the ornamented designs are appeared as gold lines and dots, which is the same way as real Chinkin.

The difference of the feel of materials between lacquered surfaces and the ornamental designs with gold powder could be expressed by adding the coefficient of specular reflection to the elements of the 3D texture map. Fig. 12 shows other examples of virtual Chinkin works.


(a) Operating the system

(b) Children creating virtual works

(c) After the Subori process

(d) After the Kin-ire process

(e) After the Shiage process

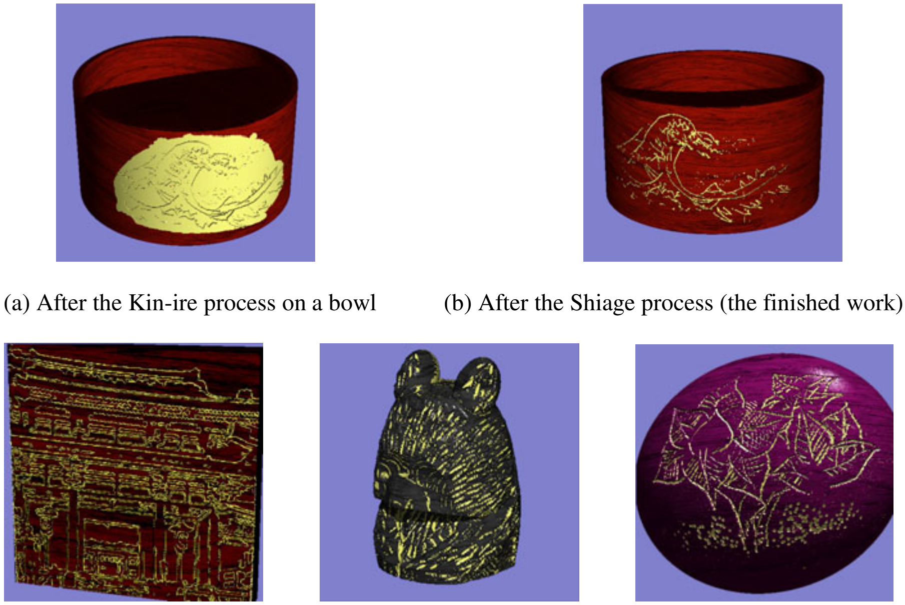
Fig. 11. Creating a virtual Chinkin work

5.3 Discussion

The prototype system enabled us to experience Chinkin creating processes in the virtual space. The computation time for carving (Subori) and painting (Kin-ire and Shiage) are fast enough to create virtual works interactively. The virtual works created with the system also have features of real Chinkin works such as the design with thin lines and small dots and a contrast between dark color of the lacquered surface and shining gold in the carved tracks.

I tested the system at a cultural event in Japan, and many children and aged people used the system (Fig 11(b)). The system had a good reputation and many people enjoyed to create virtual chinkin works.

However, some problems are shown. Some designs of Chinkin are composed of very fine lines and dots, and the resolution of the image and the 3D texture map are not enough to create such works. The resolution of the 3D texture map is limited by the memory size of the PC in the current system, and a method to reduce the size of the 3D texture map or a new technique to realize virtual painting with out 3D texture map is necessary.



(a) After the Kin-ire process on a bowl

(b) After the Shiage process (the finished work)

(c) Chinkin on a board

(d) Chinkin on a sculpture

(e) Chinkin on a sphere

Fig. 12. Examples of virtual Chinkin on virtual sculptures

6 Conclusion

In this study, the method to realize a Chinkin technique in the virtual space was proposed and it was implemented on PC. This method is based on the virtual sculpting method and user can create a 3D CG sculpture with Chinkin designs interactively. The operation of each process of the virtual Chinkin technique is similar to the real one, and the finished works of the virtual Chinkin technique are also similar to the real ones. A new rendering method that renders an image of a virtual sculpture with points and line segments enables the system to move a viewpoint in real time, and the usability of the system has been improved.

In the Subori process of the real Chinkin technique, the craftsman changes the tilt angle of a chisel during to change the shape of carving tracks, but the tilt angle of a virtual chisel is fixed in the present system. Consideration of the tilt angle of a chisel in virtual Chinkin is one of the future works. To render more realistic images, the microgeometry of carved tracks should be considered [20]. In the real Chinkin, some different size of gold powder are used to change the style of works, and it is necessary to consider the difference of property of virtual gold powder for more realistic virtual Chinkin works. Developing a virtual Chinkin system which can be used by visitors of an art museum is also expected. Testing of operation and results of the virtual Chinkin technique by craftsmen and curators would be necessary.

References

1. Kajinami, T., Hayashi, O., Narumi, T., Tanikawa, T., Hirose, M.: Digital Display Case: The Exhibition Sysytem for Conveying the Background Information. In: Proc. of SIGGRAPH 2010 Poster, DVD Proceedings (2010)
2. Kusunoki, F., Sugimoto, M., Hashizume, H.: Toward an Interactive Museum Guide with Sensing and Wireless Network Technologies. In: Proc. of WMTE 2002 (2002)
3. Shiomi, M., Kanda, T., Ishigro, H., Hagita, N.: Interactive Humanoid Robots for a Science Museum. IEEE Intelligent Systems 22(2), 25–32 (2007)
4. Oishi, T., Masuda, T., Ikeuchi, K.: Digital Restoration of the Cultural Heritages. In: Proc. of The Eighth Inter. Conf. on Virtual Systems and Multimedia (VSMM 2002), pp. 934–941 (2002)
5. Manmoto, M., Horioka, T., Yamamoto, S., Kurokawa, K.: Trends of Digital Archive in the Broadband Network Commerce. J. of the IIEEJ 33(3), 400–405 (2004)
6. Saitoh, S., Nakajima, M.: 3D Physicsbased Model for Painting. In: Proc. of SIGGRAPH 1999 Sketches, Conference Abstracts and Applications, vol. 226 (1999)
7. Murakami, K., Tsuruno, R., Genda, E.: Natural-looking strokes for drawing applications. The Visual Computer 22(6), 415–423 (2006)
8. Maeno, K., Okada, M., Toriwaki, J.: An Interactive and Intuitive Deformation System for Free Formed Curved Surface. J. of the Society for Art and Science 3(2), 168–177 (2004)
9. Matsumiya, M., Takemura, H., Yokoya, N.: A Virtual Clay Modeling System for 3D Free-form Design Using Implicit Surfaces. Transactions of Information Processing Society of Japan 42(5), 1151–1160 (2001)
10. Sourin, A.: Function Based Virtual Embossing. The Visual Computer 17(4), 258–271 (2001)
11. Mizuno, S., Okada, M., Toriwaki, J.: An Interactive Designing System with Virtual Sculpting and Virtual Woodcut Printing. Computer Graphics Forum 18(3), 183–193, 409 (1999)
12. Mizuno, S., Kobayashi, D., Okada, M., Toriwaki, J., Yamamoto, S.: Carving Painting, and Printing with a Pen Tablet. In: Proc. of EUROGRAPHICS 2005 Short Presentations, pp. 21–24 (2005)
13. Mizuno, S.: Improvement of Virtual Sculpting and Printing System with a Pressure Sensitive Pen. In: Proc. of VRSJ the 13th Annual Conference, vol. 1B5-3, pp. 177–180 (2008)
14. Digital Archive of Ishikawa Japan, Chinkin - Ornamental Beauty Created by the Art of Carving, `http://shofu.pref.ishikawa.jp/shofu/chinkin/`
15. Baxter, W., Scheib, V., Lin, M., Manocha, D.: DAB: Interactive Haptic Painting with 3D Virtual Brushes. In: Proc. of ACM SIGGRAPH 2001, pp. 461–468 (2001)
16. Miller, G.: Efficient algorithms for local and global accessibility shading. In: Proc. of ACM SIGGRAPH 1994, pp. 319–326 (1994)
17. Pixologic, ZBrush, http://pixologic.com/
18. Rusinkiewicz, S., Levoy, M.: Qsplat: a multiresoliton point rendering system for large meshs. In: Proc. of ACM SIGGRAPH 2000, pp. 343–352. ACM Press, New York (2000)
19. Pfister, H., Zwicker, M., Baar, J.V., Gross, M.: Surfels: Surface elements as rendering primitives. In: Proc. of ACM SIGGRAPH 2000, pp. 335–342 (2000)
20. Bosch, C., Pueyo, X., Merillou, S., Ghazanfarpour, D.: A Physically-based model for Rendering Realistic Scratches. Computer Graphics Forum 23(3), 361–370 (2004)

Real-Time Multi-view Human Motion Tracking Using 3D Model and Latency Tolerant Parallel Particle Swarm Optimization

Bogdan Kwolek[1,2], Tomasz Krzeszowski[2,1], and Konrad Wojciechowski[2]

[1] Rzeszów University of Technology
W. Pola 2, 35-959 Rzeszów, Poland
bkwolek@prz.edu.pl

[2] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warszawa, Poland
bytom@pjwstk.edu.pl

Abstract. This paper demonstrates how latency tolerant parallel particle swarm optimization can be used to achieve real-time full-body motion tracking. The tracking is realized using multi-view images and articulated 3D model with a truncated cones-based representation of the body. Each CPU core computes fitness score for a single camera. On each node the algorithm uses the current temporary best fitness value without waiting for the global best one from cooperating sub-swarms. The algorithm runs at 10 Hz on eight PC nodes connected by 1 GigE.

1 Introduction

Markerless 3D human motion tracking is an important problem in computer vision due to many potential applications, including, but not limited to, visual surveillance, recognizing human activities, clinical analysis and sport (biomechanics) [10]. Commercial systems for human motion capture are typically based on optical or magnetic markers and usually require laboratory environment and the attachment of markers on the body segment being analyzed. Thus, a technique for articulated human motion tracking that does not need markers attached to body would greatly extend the applicability of the motion capture.

Tracking articulated motion is difficult task because of generally unpredictable nature of human movements, high variability of human appearance, self-occlusions and depth ambiguities. The high-dimensional non-linear search space and the exponentially increasing computational overload are the main challenges in full articulated body tracking on the basis of markerless techniques. Three dimensional model based methods are generally more accurate in comparison to methods relying on learned mapping between pose exemplars and a set of image features. An articulated human body can be perceived as a kinematic chain consisting of at least eleven parts corresponding to body parts. Typically such a 3D human model consists of very simple geometric primitives like cylinders or truncated cones. On the basis of such geometrical primitives a lot of hypothetical body poses are generated

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 169–180, 2011.

and after projecting to the image space are compared with real images through a likelihood function.

Particle filtering is one of the most important and common tracking algorithms in non-intrusive human motion capture. In a particle filter each sample represents some hypothesized body pose. For a 3D model consisting of eleven geometric primitives we need around 26 parameters to describe the full body articulation. That means that tracking full articulated body is very computationally demanding task. For instance, in [2] processing for 5 seconds long video took about one hour using a particle filter with 200 samples and 10 annealing layers. In more recent work [6], the processing time of *Lee walk* sequence from Brown University is larger than one hour. Several attempts were proposed to mitigate the inherent limitations of particle filtering such as degeneracy, loss of diversity and course of dimensionality. Recently, Particle Swarm Optimization (PSO) was proposed as an alternative of particle filtering for full-body articulated motion tracking [5][8][13]. Some work has also been done in order to achieve real-time articulated body tracking [12][8].

In this work we propose a communication latency tolerant parallel algorithm for PSO based articulated motion tracking. The algorithm consists of multiple swarms that are executed in parallel on multiple computers connected via a peer-to-peer network. The computers exchange information about the location of the best particle and its corresponding fitness function of a sub-swarm. Next to each optimization iteration, information about the global particle location and the corresponding fitness score is sent asynchronously without blocking the sending thread. The message contains also data about the frame number and the iteration number. The computers receive the data in a separate thread. On the basis of arriving data the receiving threads are responsible for determining the best particles for each frame and each iteration. The best values are stored in a mutually exclusive memory. After each iteration, the processing thread checks if its global particle is better than the particle sent via other computers. If yes, it updates its own best particle and continues the optimization.

The contribution of our work is a parallel particle swarm optimization algorithm for real-time object tracking. The novelty of our work lies in the asynchronous exchange mechanism for the best particle location and its fitness score during the multiple calls of particle swarm optimization, which take place during object tracking. This results in a communication latency tolerant parallel algorithm for object tracking. The algorithm is fast and affective because it strongly relies on the stochastic nature of Particle Swarm Optimization algorithm. In particular, a sub-swarm, which as a first one finished tracking of the object in a given frame, it carries out the rediversification of the particles using its current global best particle, without waiting for the best locations of the remaining sub-swarms. In such circumstances the algorithm takes the best locations of the cooperating sub-swarms from the previous iterations, which were determined for the considered frame. The algorithm has been evaluated in multi-view based markerless full-body tracking. The tracking can be done at real-time frame rates using ordinary network of peers consisting of multi-core PCs.

2 Relevant Work

PSO was applied in a number of areas as a technique to solve large, non-linear optimization problems [11]. The applications of PSO in computer vision and graphics are still rather limited. The main applications of PSO in computer vision are connected with non-articulated object tracking. For example, [14] shows that in tasks consisting in tracking human face a variant of PSO, called sequential PSO behaves better than a particle filter in terms of tracking accuracy.

Existing algorithms for articulated motion tracking can be roughly divided into two categories, namely, discriminative and generative [9]. Discriminative approaches attempt to learn a direct mapping between image descriptors, such as edges or shapes to the 2D human pose. A major limitation is that their performance is considerably lower in circumstances in which is difficult to obtain reliable features, for instance in the cluttered scenes. Generative approaches generate a number of plausible pose hypotheses, which are then evaluated against the current image for evidence. The pose hypotheses are generated on the basis of a 3D model of the human body. Such a model is projected onto an image plane and an error function is calculated to indicate the quality of the match. The mentioned approaches are based on a rather coarse 3D models of the human body. In methods introduced in [3][4], realistic human body models were developed to accomplish tracking through analysis-by-synthesis. In such an approach the texture mapping is used to obtain a precise textured model of the person.

Very recently, PSO has been successfully applied to full-body articulated motion tracking [5][8][13]. In [5], the articulated pose is estimated through a hierarchical search. The articulated human body model is represented as a 3-D kinematic tree consisting of 13 nodes. The experiments were performed on *Lee walk* sequence, which was downsampled at frame rate of 30 Hz. On images of size 640 × 480 the average error distance between estimated pose and ground-truth pose is larger than 50 mm, whereas the processing time of the sequence with 75 images is larger than one hour. The above mentioned sequence has also been used in [13]. The average error on 15 virtual markers is about 40 mm. Our work differs from theirs in a number of ways, of which the most crucial is the focus on full body motion tracking in real-time. To the best of our knowledge, ours is the first near real-time system that is able to accomplish full-body articulated motion tracking. The quality of tracking on various number of computers was compared by analyses carried out both through qualitative visual evaluations as well as quantitatively through the use of the motion capture data as ground truth. The preliminary results demonstrate that the tracking accuracy is in the same range as the accuracy in work mentioned above.

Some parallel PSO algorithms were proposed to speed-up the optimization of complex engineering optimization problems but, to the best of our belief, so far, no parallel PSO algorithm for object tracking has been proposed. In particular, our algorithm executes not only the PSO iterations in parallel in a given frame, but being latency tolerant and asynchronous it starts processing the next frame without waiting for all best locations of the cooperating sub-swarms.

3 3D Body Model and Cost Function

The skeleton of the human body is modeled as a kinematic tree. The articulated 3D model consists of eleven segments with the limbs represented by truncated cones, which model the pelvis, torso/head, upper and lower arm and legs. The configuration of the model is defined by 26 DOF. It is parameterized by position and orientation of the pelvis in the global coordinate system and the relative angles between the connected limbs. In order to obtain the 3D human pose each truncated cone is projected into 2D image plane via perspective projection. In such a way we obtain an image with the rendered model in a given configuration. Such image features are then matched to the person extracted by image analysis.

The fitness function consists of two components: $f(x) = w_1 f_1(x) + w_2 f_2(x)$, where w_i stands for weighting coefficients that were determined experimentally. The function $f_1(x)$ reflects the degree of overlap between the body parts and the projected segments of the model into 2D image. It is expressed as the sum of two components. The first component is the overlap between the binary image and the considered rasterized image of the model. The second component is the overlap between the rasterized image and the binary one. The larger the degree of overlap is, the larger is the fitness value. The function $f_2(x)$ is calculated on the basis of distance transform based Chamfer matching.

4 Latency Tolerant Parallel PSO for Object Tracking

Particle swarm optimization is a population based optimization technique, which is stochastic in nature and makes use of the memory of each particles as well as the knowledge gained by the swarm as a whole. In the ordinary PSO algorithm the update of particle velocity and position is given by the following equations:

$$v_j^{(i)} \leftarrow w v_j^{(i)} + c_1 r_{1,j}^{(i)} (p_j^{(i)} - x_j^{(i)}) + c_2 r_{2,j}^{(i)} (p_{\mathrm{g},j} - x_j^{(i)}) \tag{1}$$

$$x_j^{(i)} \leftarrow x_j^{(i)} + v_j^{(i)} \tag{2}$$

where w is the positive inertia weight, $v_j^{(i)}$ is the velocity of particle i in dimension j, $r_{1,j}^{(i)}$ and $r_{2,j}^{(i)}$ are uniquely generated random numbers with the uniform distribution in the interval [0.0, 1.0], c_1, c_2 are positive constants, $p^{(i)}$ is the best position found so far by particle i, p_g denotes a best position, which can be:

- a global best that is immediately updated when a new best position is found by any particle in the swarm
- neighborhood best where only a specific number of particles is affected if a new best position is found by any particle in the sub-population

A topology with the global best converges faster as all the particles are attracted simultaneously to the best part of the search space. Neighborhood best allows parallel exploration of the search space by multi-swarm. Such configuration decreases the susceptibility of falling into local minima, however, it typically slows down the convergence speed.

The equation (1) has three main components. The first component, referred to as inertia, models the particle's tendency to continue the moving in the same direction. Thus, it controls the exploration of the search space. The second component, called cognitive, attracts towards the best position $p^{(i)}$ previously found by the particle. The last component is referred to as social and attracts towards the best position p_{g}. The fitness value that corresponds to $p^{(i)}$ is called local best $p^{(i)}_{\mathrm{best}}$, whereas the fitness value corresponding to p_{g} is referred to as g_{best}.

The PSO is initialized with a group of random particles (hypothetical solutions) and then it searches hyperspace (i.e. R^n) of a problem for optima. Particles move through the solution space, and undergo evaluation according to some fitness function f. Much of the success of PSO algorithms comes from the fact that individual particles have tendency to diverge from the best known position in any given iteration, enabling them to ignore local optima, while the swarm as a whole gravitates towards the global extremum. If the optimization problem is dynamic, the aim is no more to seek the extrema, but to follow their progression through the space as closely as possible. Since the object tracking process is a dynamic optimization problem, the tracking can be achieved through incorporating the temporal continuity information into the traditional PSO algorithm. This means, that the tracking can be accomplished by a sequence of static PSO optimizations to determine the best person's pose, which are followed by re-diversification of the particles to cover the possible state in the next time step. In the simplest case, the re-diversification of the particle i can be done as follows:

$$x_t^{(i)} \leftarrow \mathcal{N}(\hat{x}_{t-1}, \mathit{\Sigma}) \tag{3}$$

where $\hat{x}_{t-1}$ is the state estimate in time $t-1$. In the global best configuration the estimate $\hat{x}_{t-1}$ is equal to p_{g} determined in the last iteration. In the configuration with neighborhood best it is selected as the best position of any sub-swarm.

PSO is parallel in nature. To shorten the optimization time several studies on parallelizing the algorithm were done so far. In general, two parallelization strategies are considered, namely synchronous and asynchronous. In the synchronous algorithm at the end of each iteration all nodes communicate with each other to determine the global best fitness. In asynchronous parallelization the particles use the current temporary best fitness without waiting for the global best one. However, up to now all of the published literature reported parallel PSO algorithms for *static optimization*, where the particles are evaluated and evolved in parallel in several iterations until the global extremum is found out.

The latency tolerant parallel PSO uses asynchronous exchange mechanism for the best particle location and its fitness score during the multiple calls of particle swarm optimization, which take place during object tracking. In particular, subsequent to each iteration no barrier synchronization is executed as the algorithm strongly relies on the stochastic nature of PSO. Particularly, if a sub-swarm, which as a first one finished object tracking in a given frame, it carries out the rediversification of the particles using its current global best particle, without waiting for the global best optimum determined by the participating sub-swarms. It is worth mentioning that in such circumstances the estimate of the object state is determined using the global best locations of co-operating sub-swarms, which were available during determining in each iteration the global best location of the considered population. After each optimization iteration, information about the global particle location and the corresponding fitness score is sent asynchronously without blocking the sending thread. The frame number and the iteration number are included in the message for a control mechanism aiming at processing the same frame by all computers without large inter-frame delays. The threads receive the data in a separate thread. On the basis of arriving data the receiving threads are responsible for determining the best particles for each frame and iteration. The best values are stored in a mutually exclusive memory. After each iteration, the processing thread checks if its global particle is better than the particle sent via other computers. If yes, it updates its own best particle and continues the optimization.

5 Experimental Results

The algorithm was tested in two multi-camera systems consisting of synchronized and calibrated cameras. The first system consists of two calibrated and synchronized cameras. It acquires images of size 640 × 480 at frame rate of 15 Hz. Figure 1 depicts sample images that were acquired by the cameras. At the figure we can also see the projected and overlaid model on both input images.

In the second system the images were captured by four calibrated and synchronized cameras acquiring images of size 1920 × 1080 with rate 24 fps. Each pair of the cameras is approximately perpendicular to the other two, see the placement of video cameras in Fig. 2. A commercial motion capture (moCap) system from Vicon Nexus provides ground truth motion of the body at rate of 100 Hz. The system uses reflective markers and sixteen cameras to recover the 3D position of such markers. The synchronization between the moCap and multi-camera system is based on hardware from Vicon Giganet Lab. The digital cameras are capable to differentiate overlapping markers from each camera's view.

The precision of human motion tracking was evaluated experimentally in scenarios with a walking person. The analysis of gait is currently an active research area due to various applications in medicine, surveillance, etc. Although we

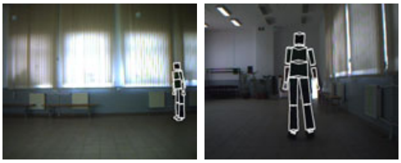


Fig. 1. Human motion tracking using two cameras. The images illustrate the initial model configuration overlaid on the image in first frame.

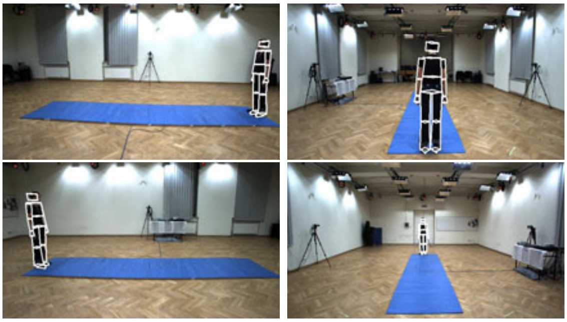


Fig. 2. Layout of the laboratory with four cameras. The images illustrate the initial model configuration, overlaid on the image in first frame and seen in view 1 and 2 (upper row), and in view 3 and 4 (bottom row).

focused on tracking of torso and legs, we also estimated the pose of both arms as well as of the head. The body pose is described by position and orientation of the pelvis in the global coordinate system as well as relative angles between the connected limbs. The results obtained on various number of computers were compared by analyses carried out both through qualitative visual evaluations as well as quantitatively by the use of the motion capture data as ground truth.

Figure 3 shows results obtained in the two camera system. The left images in each image pair depict the projected and overlaid model on the image from the first camera, whereas the right images are from the second one. The initialization of the system was done manually through fitting the 3D model onto the images, see Fig. 1. The tracking was done using 300 particles and 10 iterations.

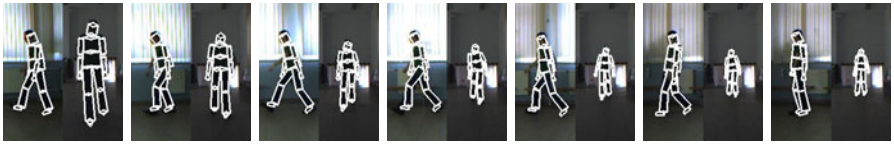


Fig. 3. Articulated 3D human body tracking in two camera setup. Shown are results in frames #10, 20, 30, 40, 50, 60, 70. The left sub-images are seen from view 1, whereas the right ones are seen from view 2.

Figure 4 demonstrates some results that were obtained in the four camera system. The quality of tracking is illustrated using images from first and second camera. The initialization of the tracking was done manually. Optionally, the tracking can be initialized on the basis of data from the moCap system. The same number of particles and iterations was utilized as in the previous experiment. In all experiments on image sequences from the four camera system we used images of size 480 × 270.

Figure 5 depicts the errors that were obtained during motion tracking using one and two desktop computers. The experiments were done on image sequences

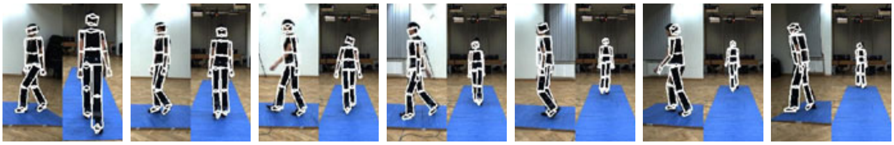


Fig. 4. Articulated 3D human body tracking in four camera setup. Shown are results in frames #20, 40, 60, 80, 100, 120, 140. The left sub-images are seen from view 1, whereas the right ones are seen from view 2.

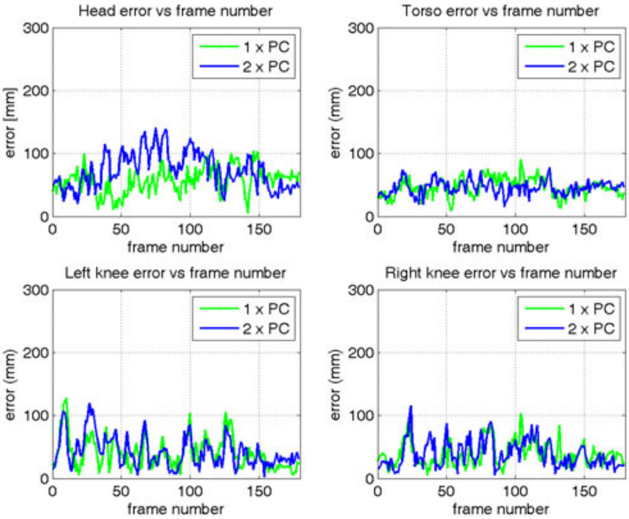


Fig. 5. Tracking errors [mm] versus frame number at 1 and 2 PCs

from the four camera system. The errors of tracking the head, torso and knee were calculated using moCap data as ground truth. In optimizations we used 300 particles and 10 iterations. In the configuration consisting of two computers the optimizations were achieved using 150 particles on each computer. As we can observe in the plots shown at Fig. 5, the difference between error estimates obtained by the ordinary algorithm and the parallel algorithm running on two computers is not significant.

In Fig. 6 are shown the error estimates that were obtained on single and eight computers. In a PC cluster with 8 nodes the optimizations were performed using 38 particles on each computer. As we can observe, the average error is far below 90 mm. It is worth noting here that something better results can be obtained using our GLPSO (Global-Local PSO) algorithm [7].

The experiments were conducted on desktop PCs with 4 GB RAM, Intel Core i5, 2.8 GHz. All measurements were conducted on a cluster that was composed of identical machines connected with a TCP/IP 1 GigE (Gigabit Ethernet)

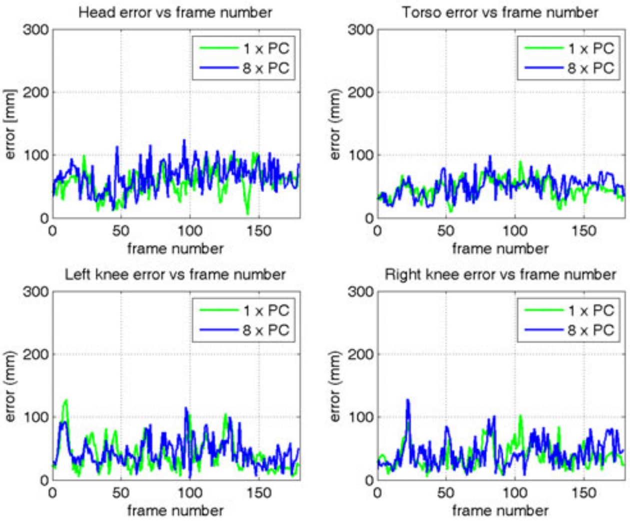


Fig. 6. Tracking errors [mm] versus frame number at 1 and 8 PCs.

local area network. The parallelization of the code was done using OpenMP directives. The parallel computations were realized on multi-core (4-core) CPUs.

Currently, OpenMP is widely utilized standard for parallelizing programs in a shared memory environment [1]. It consists of a set of directives (pragmas) and library routines that can be inserted into Fortran or C/C++ codes to enable use of more than one thread. OpenMP provides a fork-and-join execution model in which a program begins execution as a thread. The thread executes sequentially until a parallelization directive for a structured block of code is found. If this takes place, such a thread creates a set of threads and becomes the master thread of the new group of threads. Each thread executes the same code redundantly until the end of the parallel section and the threads communicate by sharing variables. The exit point of a structured block is an implicit synchronization point for the master thread and the threads created for the block. After the synchronization the master thread continues with the computation and the other threads end. In our system each CPU core is responsible for calculation of the fitness function for single camera.

Table 1 shows computation times and speeds-up that were obtained on our PC cluster. The depicted times are needed to extract single model configuration using images from four camera views. In the experiments we focused on efficiency of parallel particle swarm optimization algorithm and therefore the computation times do not comprise the image processing. The image processing was done in advance and all images needed to compute the fitness score were stored on local hard drives. It is worth mentioning that time needed for image processing is about 20% of the total processing time. Moreover, the code of image processing can be easily parallelized. Currently, the communication between the PC nodes

Table 1. Tracking time [ms] for single human pose (computed on the basis of images from 4 camera views) and speed-up

		Latency tolerant		Blocking	
#PCs	#particles	time [ms]	speed-up	time [ms]	speed-up
1	300	635.7	-	635.7	-
2	2 × 150	339.6	1.87	370.4	1.72
3	3 × 100	227.1	2.80	257.5	2.47
4	4 × 75	173.7	3.66	202.2	3.14
6	6 × 50	123.7	5.14	146.5	4.34
8	8 × 38	96.9	6.56	110.8	5.74

is realized using popular QT library. As we can see, the speed-up of our latency tolerant parallel PSO is considerable. Using a cluster consisting of 8 PCs and PSO with 300 particles and 10 iterations the human motion tracking can be done at about 10 fps. The tracking time of blocking version of the parallel PSO is considerably larger in comparison to our latency tolerant algorithm. When images from two camera system are used, we can perform full-body motion tracking together with image preprocessing in real-time with 10 fps.

In Tab. 2 are depicted the average errors that were obtained during a computations on different numbers of computers. The pose error in each frame was determined on the basis of $M = 39$ markers $m_i(x) \in R^3, \quad i = 1, \ldots, M$ expressing locations in the world coordinates. The pose error was expressed as the average Euclidean distance:

$$E(x, \hat{x}) = \frac{1}{M} \sum_{i=1}^{M} ||m_i(x) - m_i(\hat{x})|| \tag{4}$$

where $m_i(x)$ denotes for marker's position that was calculated using the estimated pose, whereas $m_i(\hat{x})$ stands for the position that was determined using data from our motion capture system. From the above set of markers, four markers were placed on the head, seven markers on each arm, 6 on the legs, 5 on the torso and 4 markers were attached to the pelvis. Given the discussed placement of the markers on the human body the corresponding virtual marker's were assigned on the 3D model. The position of such virtual markers was determined for each estimate of the human pose and then employed in calculating the average Euclidean distance expressed by (4). The ground truth was extracted on the basis of data stored in `c3d` files. Finally, the average errors shown in Tab. 2 were calculated on the basis of the following equation:

$$Err(x, \hat{x}) = \frac{1}{LM} \sum_{k=1}^{L} \sum_{i=1}^{M} ||m_i(x) - m_i(\hat{x})|| \tag{5}$$

where L denotes the number of frames in the utilized test sequences. The discussed results were obtained on $L = 180$ images, see also Fig. 4, and averaged

Table 2. Average errors [mm]

#PCs	#particles	error [mm]	std. dev. [mm]
1	300	75.8	45.8
2	2 × 150	72.2	38.1
3	3 × 100	71.9	34.1
4	4 × 75	74.3	40.3
6	6 × 50	73.9	37.9
8	8 × 38	72.7	36.9

over 5 runs of the algorithm. As we can observe, for a configuration with multiple nodes the average error is smaller than the error obtained on a single node. This means that multiple swarms PSO can generate better results in comparison to PSO based on single swarm.

The complete human motion capture system was written in C/C++. One of the future research directions of the presented approach is to explore multiple GPUs to further shorten the processing time [8].

6 Conclusions

We have presented communication latency tolerant parallel algorithm for particle swarm optimization. We demonstrated experimentally that the parallel PSO is especially well suited for real-time full-body articulated object tracking. To show its advantages we have conducted several experiments on walking sequences and realized computations on different numbers of computers connected with a TCP/IP 1 GigE local area network. The quality of tracking was compared by analyses carried out both through qualitative visual evaluations as well as quantitatively through the use of the motion capture data as ground truth.

Acknowledgment. This paper has been supported by the project “System with a library of modules for advanced analysis and an interactive synthesis of human motion” co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme - Priority Axis 1.

References

1. Chapman, B., Jost, G., van der Pas, R., Kuck, D.J.: Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press, Cambridge (2007)
2. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: IEEE Int. Conf. on Pattern Recognition, pp. 126–133 (2000)
3. Gavrila, D.M., Davis, L.S.: 3-D model-based tracking of humans in action: a multi-view approach. In: Proc. of the Int. Conf. on Computer Vision and Pattern Rec., CVPR 1996, pp. 73–80. IEEE Computer Society, Washington, DC (1996)

4. Grard, P., Gagalowicz, A.: Human body tracking using a 3D generic model applied to golf swing analysis. In: Int. Conf. on Computer Vision / Computer Graphics Collaboration Techniques and Applications (2003)
5. Ivekovic, S., John, V., Trucco, E.: Markerless multi-view articulated pose estimation using adaptive hierarchical particle swarm optimisation. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplicatons 2010. LNCS, vol. 6024, pp. 241–250. Springer, Heidelberg (2010)
6. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vis. Comput. 28, 1530–1547 (2010)
7. Krzeszowski, T., Kwolek, B., Wojciechowski, K.: Model-based 3D human motion capture using global-local particle swarm optimizations. In: Int. Conf. on Computer Recognition Systems. AISC, vol. 95, pp. 297–306. Springer, Heidelberg (2011)
8. Krzeszowski, T., Kwolek, B., Wojciechowski, K.: GPU-accelerated tracking of the motion of 3D articulated figure. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6374, pp. 155–162. Springer, Heidelberg (2010)
9. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Comput. Vis. Image Underst. 104, 90–126 (2006)
10. Muendermann, L., Corazza, S., Andriacchi, T.: The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications. Journal of Neuroengineering and Rehabilitation 3(1) (2006)
11. Poli, R.: Analysis of the publications on the applications of particle swarm optimisation. J. Artif. Evol. App., 4:1–4:10 (January 2008)
12. Schmidt, J., Fritsch, J., Kwolek, B.: Kernel particle filter for real-time 3D body tracking in monocular color images. In: IEEE Int. Conf. on Face and Gesture Rec., Southampton, UK, pp. 567–572. IEEE Computer Society Press, Los Alamitos (2006)
13. Zhang, X., Hu, W., Wang, X., Kong, Y., Xie, N., Wang, H., Ling, H., Maybank, S.: A swarm intelligence based searching strategy for articulated 3D human body tracking. In: IEEE Workshop on 3D Information Extraction for Video Analysis and Mining in Conjuction with CVPR, pp. 45–50. IEEE, Los Alamitos (2010)
14. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: IEEE Int. Conf. on CVPR, pp. 1–8 (2008)

Snap Image Composition[*]

Yael Pritch, Yair Poleg, and Shmuel Peleg

School of Computer Science, The Hebrew University, Jerusalem, Israel

Abstract. Snap Composition broadens the applicability of interactive image composition. Current tools, like Adobe's Photomerge Group Shot, do an excellent job when the background can be aligned and objects have limited motion. Snap Composition works well even when the input images include different objects and the backgrounds cannot be aligned. The power of Snap Composition comes from the ability to assign for every output pixel a source pixel in any input image, and from any location in that image. An energy value is computed for each such assignment, representing both the user constraints and the quality of composition. Minimization of this energy gives the desired composition.

Composition is performed once a user marks objects in the different images, and optionally drags them into a new location in the target canvas. The background around the dragged objects, as well as the final locations of the objects themselves, will be automatically computed for seamless composition. If the user does not drag the selected objects to a desired place, they will automatically snap into a suitable location. A video describing the results can be seen in www.vision.huji.ac.il/shiftmap/SnapVideo.mp4.

1 Introduction

Image composition is common in digital image editing, whose objective is to combine images from different shots into a single output image that looks natural and realistic. Three approaches are common for image composition: Matting, Blending, and Optimal Cuts. Matting [22] attempts to make an accurate segmentation of an object, allowing to place it in a new image. In image blending [6,15,13,10,20] a user builds a new image from patches taken from the input images, and the seams between these patches are eliminated by the blending. In Optimal Cuts [1], the seam between images to be combined is computed automatically within their overlap areas. In all of the above, the user placement of the objects is a hard constraint, and the geometry of both the background and the foreground do not change. Rearrangement of a single image is presented in [7,19,2,16]. Snap Composition allows the user to define approximate regions and target locations, letting objects snap into place during a single optimization process. Background rearrangement to match the objects is done as well.

Digital Photomontage [1] presented a pioneering approach to create a seamless composite from selected instances of objects in multiple aligned images. Their

[*] This research has been suppoted by a grant from the Israel Science Foundation.

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 181–191, 2011.

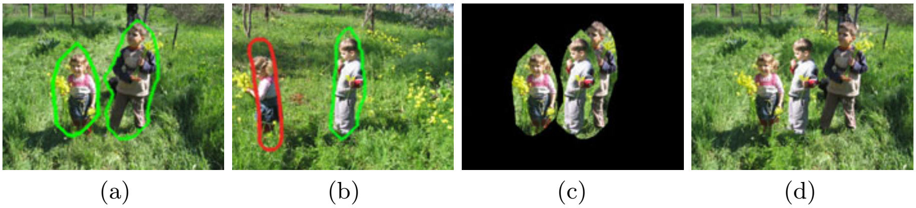


Fig. 1. Snap Composition. (a-b): input images with regions to include in composition are marked in green and regions to avoid are marked in red. (c) Initial canvas, where selected kids are in their original locations. (d) Snap Composition results. The kids from (a) were automatically spaced to allow the kid from (b) to snap in between, and background is rearranged for a seamless composition.

process selected optimal seams followed by gradient-domain fusion. It is assumed that the camera is in the same location in all images, and that objects move very little. When objects move significantly between images, or when camera motion causes parallax, a different approach is needed.

The issues of moving objects and parallax were also addressed for construction of panoramic images [21,9]. While panoramic stitching is based on global registration, the goal of image composition is to satisfy user requirements. This is done by using the concept of visual similarity between the output and the input images, rather than trying to achieve a true geometric consistency. Fig. 6 shows possible conflict between global alignment and user sketch.

The computation of individual shifts for every pixel using global energy minimization, as done in Snap Composition, follows the ShiftMap framework [16]. As in other composition approaches, the user constraints in ShiftMap are hard, and several attempts may be necessary until the user places the objects in the "right" locations. The energy terms in Snap Composition allow flexibility of object location, which is automatically determined during optimization.

The task shown in Fig. 1 is an example of the issues addressed in this paper. A user would like to insert the selected "green" kid from Fig. 1.b between the two kids in Fig. 1.a, even though there is not enough space there. Using existing methods, this task requires two steps: (i) Specify new locations for the two kids in Fig. 1.a with a larger gap between them, and compute a new image. (ii) Place the kid from Fig 1.b into its new location using image cloning, matting, or digital Photomontage. During this process the user has made three selections for the locations of the three kids, and many attempts may be needed until a good result is obtained. Snap-Composition determines these locations using a single global optimization, substantially reducing user interaction. Fig. 2 compares the result of Snap Composition to Adobe Photomerge Group Shot using same images.

Another contribution of this paper addresses the approximate optimization of graph labeling whose energy function has a data term and a smoothness term. We have observed that a better optimum is obtained when the weight of the

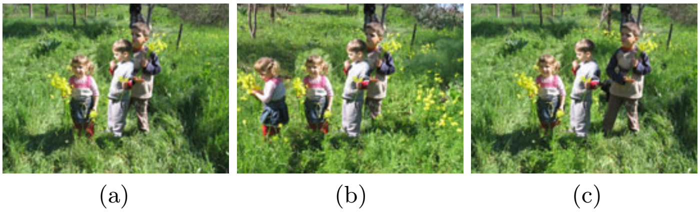


Fig. 2. Comparing the example in Fig. 1 to Adobe Photomerge Group Shot. This emphasizes the importance of individual displacements of objects, which is not one of the features supported by Photomerge. (a-b) Photomerge results obtained using different ordering of input images. (c): Snap Composition results.

smoothness term is gradually increased during iterations. This process, inspired by Lagrangian relaxation, was found especially helpful in cases that the graph has a very large number of labels.

2 Editing as Graph Labeling

Image composition is represented as graph labeling as done in ShiftMap image editing [16]. The composite image $R(u, v)$ is reconstructed from the input images $I(x, y, i)$ based on the shifts $M(u, v) = (t_x, t_y, i)$ as follows: $R(u, v) = I(u + t_x, v + t_y, i)$, where i is an index on the input images. In the graph labeling representation the nodes are the pixels (u, v) of the output image, where each output pixel (u, v) is labeled by a shift $M(u, v) = (t_x, t_y, i)$. The optimal shifts M minimize the new cost function below, tailored for image composition:

$$E(M) = \sum_{p \in R} E_d + \sum_{(p,q) \in N} \{E_r + \alpha E_s\} \tag{1}$$

The data term $E_d(M(p))$ is defined over single pixels, providing user constraints such as desired locations of objects or a preference that an area will not be used in the output. $E_r(M(p), M(q))$ is a pairwise term built from user constraints for preserving object integrity. $E_s(M(p), M(q))$ is a pairwise term addressing seamless composition. N of the pairwise term is defined over four spatial neighbors of each pixel, and α is a user defined weight for the smoothness term. In the optimization process we gradually increase α to get a better convergence as describe in Sec. 2.3. Each term will now be defined in detail. Once the graph is given, optimal labeling (i.e. optimal shifts) is computed using the alpha expansion algorithm for graph cuts [11,4,5].

2.1 User Constraints

The data term E_d indicates the user constraints such as the approximate location of an object in the output image. Specific pixels in the input image can be marked by the user as "do not use in output" or as "must appear in some approximate output location". Each object marked by the user is represented by a mask $S(x, y, i)$ over the input images. $S(x, y, i)$ will be non zero for the marked pixels where user has imposed a constraint. If the user dragged an object to a particular location on the target canvas, the mask is also assigned a desired approximate shift (P_x, P_y), otherwise it is assumed that the desired shift is zero.

The data term. $E_d(M(u, v))$ for an output pixel (u, v) with a shift $M(u, v) = (t_x, t_y, i)$ is assigned as follows:

1. If $(u + t_x, v + t_y)$ falls outside image boundary, or if $S(u + t_x, v + t_y, i)$ is marked as "must disappear", then $E_d(M(u, v)) = \infty$.
2. In case $S(u+t_x, v+t_y, i)$ is marked by the user to move by (P_x, P_y), if $|t_x - P_x, t_y - P_y| \leq LD$, then $E_d(M(u, v)) = -1$. Otherwise, $|t_x - P_x, t_y - P_y| > LD$, and we assign $E_d(M(u, v)) = 1$. LD is a parameter specifying permitted deviations from the location specified by the user. When LD is smaller than the size of its associated region, no region duplication is possible. We usually used LD values that are about 10 percent from the image dimensions to allow flexibility in object location, while avoiding unwanted duplications of the marked objects.
3. In all other cases $E_d(M(u, v)) = 0$.

The rigidity term. $E_r(M(p), M(q))$ verifies that the marked objects move coherently, and do not occlude each other. Let $M(p)$ point to $d_1 = (x_1, y_1, i_1)$ and $M(q)$ point to $d_2 = (x_2, y_2, i_2)$. If either d_1 or d_2 points to a pixel in a selected area (non-zero in $S(x, y, i)$), and $M(p) \neq M(q)$,we incur a cost setting $E_r(M(p), M(q)) = \infty$.

This term is verifying that two neighbors marked pixels in any of the input images must remain neighbors in the output image. Together with the smoothness term that penalizes stitching artifacts, it helps to avoid the situation where multiple marked objects are occluding each other. If rigidity term together with LD value of E_d, were not introduced, the high benefit of including the marked pixels in the output would have caused marked objects to be duplicated several times and create unwanted results.

2.2 The Smoothness Constraint

The smoothness term. $E_s(M(p), M(q))$ represents discontinuities introduced to the output image by discontinuities in the shifts: A shift discontinuity exists in the output image R between two neighboring locations, (u_1, v_1) and (u_2, v_2), if their shifts are different $(M(u_1, v_1) \neq M(u_2, v_2))$. The smoothness term $E_s(M)$ takes into account both color differences and gradient differences between corresponding spatial neighbors in the output image and in the input images. This term is similar to [1,17].

$$E_s(M) = \sum_{(u,v)\in R} \sum_j (R((u, v) + e_j) - I((M(u, v)) + e_j))^2 + \qquad (2)$$

$$\beta \sum_{(u,v)\in R} \sum_j (\nabla R((u, v) + e_j) - \nabla I((M(u, v)) + e_j))^2$$

where e_j are vectors representing the four immediate neighbors of a pixel, the color differences are Euclidean distances in RGB, ∇ represents the magnitude of the image gradients, and β is a weight combining these two terms. In our experiments we used $\beta = 10$.

As our pairwise energy terms are not a metric distance, the theoretical guarantees of alpha expansion are lost. In practice we have found that good results are still possible, as also observed in [12,1].

2.3 Relaxation in Energy Minimization

Energy minimization by graph labeling has very high complexity due to its non convex nature and the very large number of labels. The approximate optimization methods are not likely to reach the global minimum. However, we found that if we gradually increase the weight of the smoothness term during the iterations of the alpha expansion, and use in each iteration the labeling of the previous iteration as an initial guess, we converge to a better result. This heuristic solution has been inspired by Lagrangian Relaxation [8] and Graduated Non-Convexity [3].

We start our iterations with the relatively easy satisfaction of the user constraints (data term and rigidity term). The smoothness term is much harder to satisfy, and we avoid using it in the first iteration ($\alpha = 0$), but gradually increase the weight of the smoothness term during iterations until we reach the desired weight. In a set of experiments we found that this approach obtained a lower energy and a better result in comparison to the use of the desired weight from the first iteration. Figure 3 compares the minimum energy obtained by the two approaches.

2.4 Hierarchical Solution for Graph Labeling

We use a multi-resolution approach to reduce the complexity of finding the optimal graph labeling. Multi-resolution approaches to graph labeling were also done in[14,18,16]. We build a Gaussian pyramid for the input images, and coarse shifts are computed using the small input images to generate a small composite image. This operation is very efficient, as both the number of nodes and the number of possible labels (shifts) is small. In practice we select only shifts whose magnitudes are smaller than 25% of image size. Once coarse shifts are assigned, they are interpolated as an initial guess for the labels in the higher resolution level of the pyramid.

There are too many possible shifts when addressing higher resolutions in the pyramid. We limit our examination only to shifts that are popular in the initial interpolation from the lower resolution, and are used as labels of at least 50 pixels. We also add the nine closest shifts around each of these candidates. While we do not use all possible shifts, the smaller set still allows a pixel to get a more accurate shift, and also to switch from one group of pixels to another, improving the cut between regions.

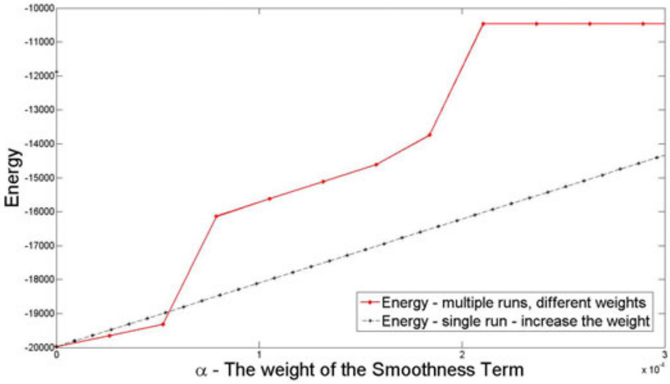
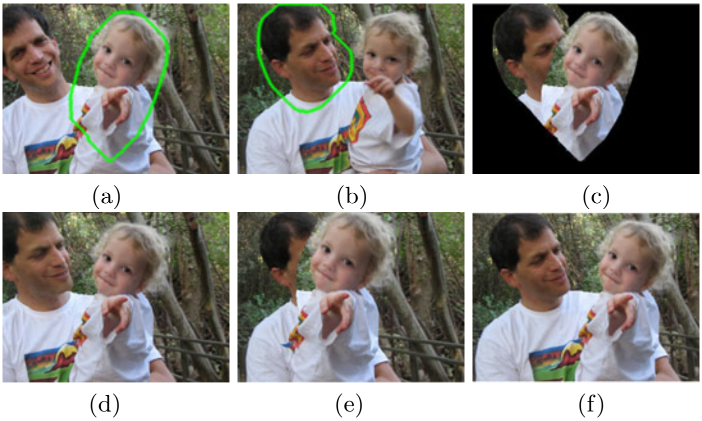


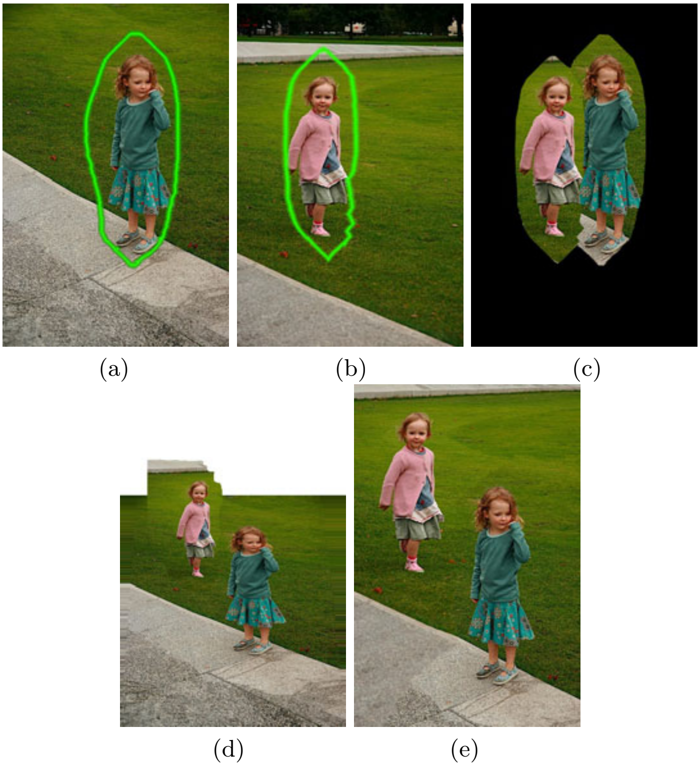
Fig. 3. Gradual Relaxation: The red curve represents the minimal energy obtained when minimization is performed on the desired weight of the smoothness term, a different run for each weight. The black curve represents the minimum obtained when iterations started with no smoothness term, and the weight of the smoothness term is gradually increased until its desired value is reached (single run). Gradual increase of the weight of the smoothness term gives a better minimum most of the times.

Our pyramid contains about 100 × 100 pixels in its smallest level. It took up to a minute to perform the composition on most images in this paper, and a GPGPU implementation is in progress with the goal of reaching interactive speeds.


(a) (b) (c)

(d) (e) (f)

Fig. 4. (a-b) Input images where selected regions are marked in green. (c) Canvas image with selected regions placed in their initial location. The overlap between the selected regions presents a special difficulty to other tools. (d) Microsoft Photo Fuse results. Note that the man's head is cropped. (e) Adobe Photomerge Group Shot results. (f) Snap Composition results.



(a) (b) (c)

(d) (e)

Fig. 5. (a-b) Input images where selected regions are marked in green. (c) Canvas image with selected regions placed in their initial locations. (d) Adobe Photomerge Group Shot results. While the background is nicely aligned, the "blue" girl is distorted. (e) Snap composition results.

3 Experimental Results

We tested Snap Composition against both Adobe's "Photomerge Group Shot" and Microsoft's "Photo Fuse". In most cases but two "Photo Fuse" failed to align, and no results are given for the failed cases. Fig. 4 compares Snap Composition against both methods. Snap composition creates a better composite image as it enables local modifications of both images after alignment, overcoming geometric misalignments that are not handled by the other methods. Fig. 5 compares to "Photomerge Group Shot", where the blue girl has been distorted. More examples are in Fig. 6, Fig. 8, and Fig. 9. We have found that that user marking by drawing a thick outline around the object is most convenient, and we used it in most of our examples. But the marking is very flexible due to the effect of the smoothness terms, and as demonstrated in Fig. 6 the marking does not need to include an entire object.

Snap Composition sometimes fails when filling the background, and the most common failure is duplication of regions that belong to the foreground, into the

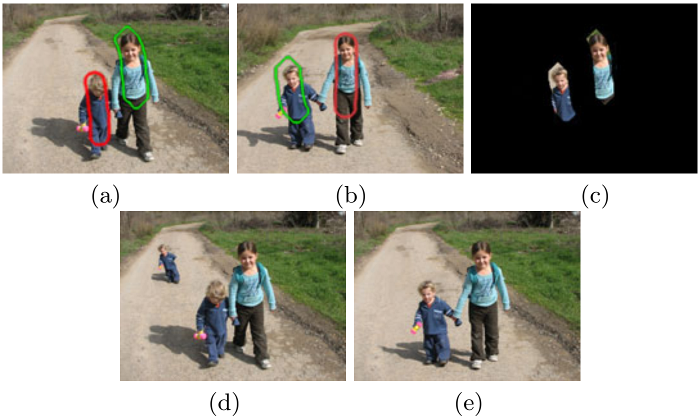


Fig. 6. (a-b) Input images. Regions to be included in composition are marked in green, unwanted regions marked in red. (c) Selected regions shown at their user selected location on the composition canvas.(d) Adobe Photomerge Group Shot places the kid on the road twice based on a global alignment. (e) Snap composition result, allowing rearrangement of both objects and background.

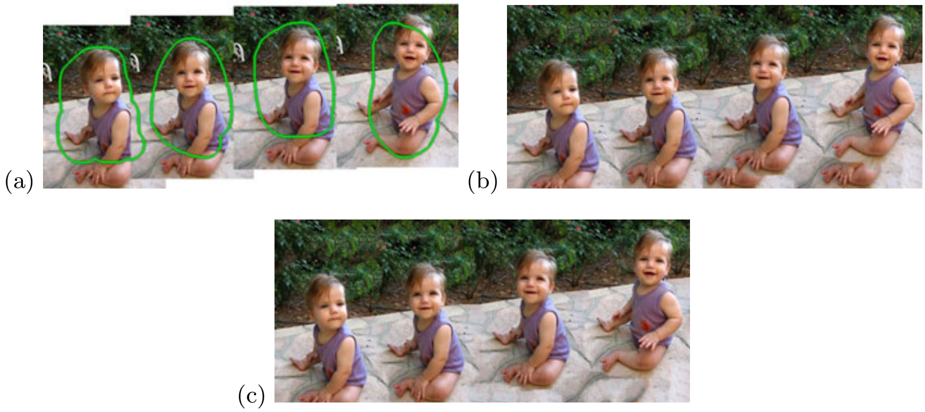
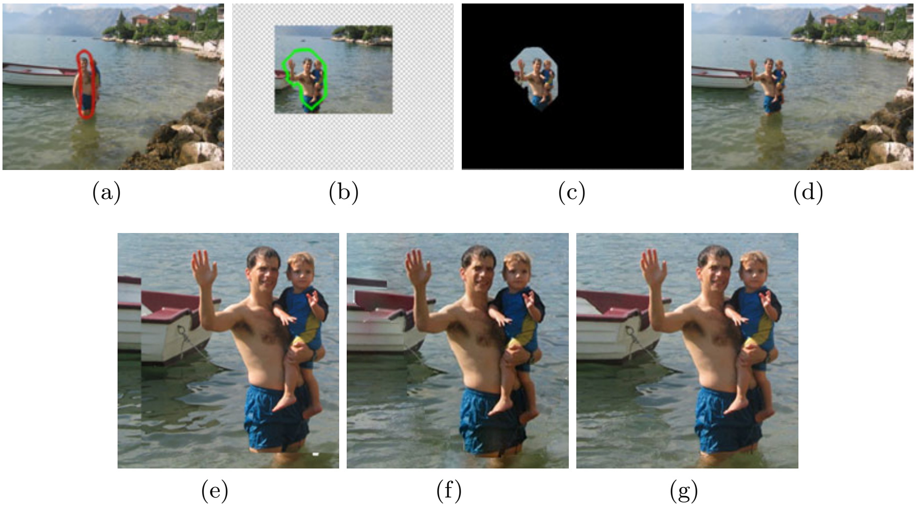


Fig. 7. Failure example: (a) Four similar input images placed in a canvas. Regions to keep are marked in green. (b) Initial Snap Composition. Some body parts are duplicated to create the background. (c) User interaction, giving higher costs to these duplications (as a second interaction phase), created a nice final composition image.

background. Such failure is shown in Fig. 7. Another possible failure case is the duplication of marked objects when the allowed deviation of object location (LD defined in 2.1) is very large. The user can recover from these failures by marking on the output image the undesired duplications. Pixels in the marked region will come from elsewhere, and this is done by increasing the cost on the undesired labeling. We used this feature only in the example of Fig. 7. All other examples did not need this feature.


(a) (b) (c) (d)

(e) (f) (g)

Fig. 8. (a-b) Input images after alignment, where wanted regions are marked in green and unwanted regions are marked in red. (c) Canvas image with selected objects placed in their initial location. (d) Snap Composition results. (e) Zoom in to composition by Adobe Photomerge Group Shot. (f) Zoom in to composition by Microsoft Photo Fuse. (g) Zoom in to Snap Composition.

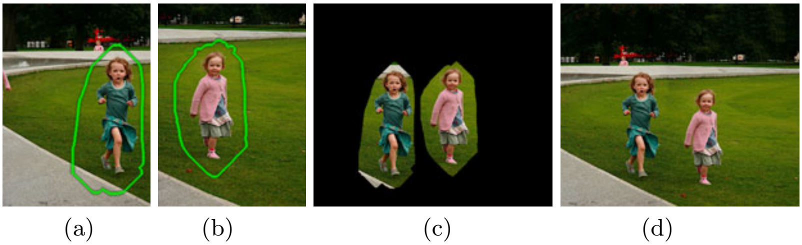

(a) (b) (c) (d)

Fig. 9. (a-b) Input images where wanted region are marked in green. (c) Canvas image with selected objects placed in their initial location. (d) Snap Composition results. Note that the "pink" girl has been shifted down automatically for better composition.

In all the experiments we avoided using photometric blending, in order to show the pure effect of Snap Composition.

The user interaction tool we have built includes the ability to (i) sketch an area that should be included in the output; (ii) sketch an area that should not be included in the output; (iii) set approximate locations of selected areas in output canvas; and (iv) sketch on the output image areas that should be changed. The use of this tool is demonstrated in the accompanied video.

4 Conclusion and Discussion

This paper presents Snap Composition, a method for image composition that extends existing image composition approaches by adding the possibility to automatically compute best locations of the objects and the rearrangement of the background for seamless composition. All Computations are done in a single global optimization step. These capabilities are not possible as a single automatic step in any available composition tool, and increase the applicability of interactive image composition.

While the examples shown in this paper do not include any photometric blending, it is recommended that blending such as gradient domain blending be applied at the seams of stitched regions.

In addition to the visual results, it was found that gradual increase of the smoothness term lets the process converge to a better minimum and a better result.

References

1. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Interactive digital photomontage. In: SIGGRAPH, pp. 294–302 (2004)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graph (July 28, 2009)
3. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press, Cambridge (1987)
4. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEET-PAMI 26(9), 1124–1137 (2004)
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEET-PAMI 23 (2001)
6. Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Trans. Graph. 2, 217–236 (1983)
7. Cho, T., Butman, M., Avidan, S., Freeman, W.: The patch transform and its applications to image editing. In: CVPR 2008 (2008)
8. Fisher, M.L.: The lagrangian relaxation method for solving integer programming problems. Manage. Sci. 50, 1861–1871 (2004)
9. Jia, J., Tang, C.: Image stitching using structure deformation. IEEE T-PAMI 30, 617–631 (2008)
10. Jia, J., Sun, J., Tang, C.-K., Shum, H.-Y.: Drag-and-drop pasting. ACM Trans. Graph. 25, 631–637 (2006)
11. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 65–81. Springer, Heidelberg (2002)
12. Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. In: SIGGRAPH 2003, pp. 277–286 (2003)
13. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: ECCV 2006 (2006)
14. Lombaert, H., Sun, Y., Grady, L., Xu, C.: A multilevel banded graph cuts method for fast image segmentation. In: ICCV 2005, vol. 1, pp. 259–265 (2005)
15. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: SIGGRAPH 2003, New York, NY, USA, pp. 313–318 (2003)
16. Pritch, Y., Kav-Venaki, E., Peleg, S.: Shift-map image editing. In: ICCV 2009, Kyoto, pp. 151–158 (September 2009)

17. Rother, C., Bordeaux, L., Hamadi, Y., Blake, A.: Autocollage. In: SIGGRAPH 2006, pp. 847–852 (2006)
18. Rubinstein, M., Shamir, A., Avidan, S.: Improved seam carving for video retargeting. ACM Trans. Graph. 27(3), 1–9 (2008)
19. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: CVPR 2008 (2008)
20. Tao, M.W., Johnson, M.K., Paris, S.: Error-tolerant image compositing. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 31–44. Springer, Heidelberg (2010)
21. Uyttendaele, M., Eden, A., Szeliski, R.: Eliminating ghosting and exposure artifacts in image mosaics. In: CVPR 2001, Hawaii, vol. II, pp. 509–516 (December 2001)
22. Wang, J., Cohen, M.: Image and Video Matting: A Survey. Foundations and Trends in Computer Graphics and Vision 3(2), 97–175 (2007)

Towards the Automatic Generation of 3D Photo-Realistic Avatars Using 3D Scanned Data

Thibault Luginbühl[1], Laurent Delattre[2], and André Gagalowicz[1]

[1] Inria Rocquencourt, Domaine de Voluceau BP105, 78153 Le Chesnay, France
{thibault.luginbuh,andre.gagalowicz}@inria.fr
[2] 3D Ouest, Technopole Anticipa, 22300 Lannion, France
delattre@3douest.com

Abstract. The use of 3D avatar scanned from real person 3D data has become more and more common in different fields such as video games and movies. The target applications for these models require different constraints on the result to be fulfilled. In this paper we propose to generate high-resolution closed meshes of a person that can be used for virtual try-on applications, therefore the 3D model should be precise enough for the customer to recognize him/herself wearing a virtual garment. Our approach uses a generic model which is deformed using fast editing method to fit a point cloud obtained from a 3d scanner. Our system is fully automatic and requires only an unoriented point cloud.

1 Context and Related Works

1.1 Virtual Try-on Applications

Virtual try-on applications are still in development, although some techniques already show impressive results [13] [1] none of them is widely used in large scale industrial and commercial process. The main steps leading to a virtual try-on application should be from the customer point of view :

- Acquisition of the 3D measurements of the body
- Cloth selection from a database
- Watching the simulation showing the cloth on his/her 3D avatar

Each one of these steps has focused the attention of the scientific community : building a cloth database requires the collection of garments 2D pattern designed by professionals as well as mechanical parameters of the textiles obtained from Kawabata machines. Then the 3D pattern garments need to be converted to 3D and positioned automatically on the avatar. Finally realistic simulation is achieved by solving the mechanical equations taking care of collision and buckling. [7]

The work described in this paper will focus on the first step. Acquisition of the 3D measurements of the customer is a crucial step to help him/her choose the best fitting size for the cloth. Therefore the avatar must be precise enough for the simulation to be as realistic as possible. Furthermore, the cloth simulation

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 192–203, 2011.

imposes constraints on the mesh to be valid, it must be a closed manifold of genus 0 to avoid cloth from penetrating a topological hole.

In our system acquisition is performed using a 3D laser scanner developed by the French company *3D Ouest* [2]. The customer is asked to wear only underwear not too loose so that the measures are as close as possible to the customer shape. The acquisition provides a point cloud that needs to be turned into a 3D mesh ready to use in the cloth simulation. Nevertheless, the use of 3D avatars should not limited to the pipeline of cloth simulation ; we think that the possibility to enrich automatically the features of the avatar is very important for animation or statistical analysis of the human body for instance. To provide more features on the final avatar we propose to use a generic model that can contain many informations computed offline and to fit it to the acquired point cloud. With this strategy all reconstructed models will be related to one single mesh topology allowing statistical analysis and a rough skinning can be used to make basic animations of all the generated models.

The process has to be fast : the customer should not wait too long to get his/her avatar, one current reference time is around one minute that is the time for the customer to get his/her cloth back after the scanning process. One minute is quite enough for a large variety of strategies. We propose to use some paradigms of surface editing methods that are able to run at interactive time speed to get our mesh quickly. To make fitting possible we must first analyse the point cloud to detect feature points and to segment it in order to guide the deformation process.

But for our process to be efficient we have to make some assumptions on the position of the customer. We choose a standard position that we ask the customer to keep during the scan. The person must be standing with the two feet on the ground and legs slightly apart. Arms must be slightly apart from the torso. See Fig 1.

1.2 Related Works

Interested readers can find an exhaustive survey on segmentation and modeling of 3D human body using scanned data in [24]

Feature Points Detection. We are interested only in automatic feature points detection without markers on the body. Manual processing should be avoided so that a client won't be too disturbed and manual placement of markers is itself time consuming. Automatic detection is still a complicated problem if general poses are considered but for a fixed position we can obtain enough points using dedicated detection algorithms. A lot of methods rely on the analysis of slices using various criteria to extract special points (extrema, angles, convexity or concavity etc.). Leong et al. [17] tried to find mathematical definitions of points defined in ASTM and ISO and detected them by combining image analysis and computational geometry. Those criteria are used along with statistical information about the human body proportions to limit the search area. Wang et al. [22] proposed a full pipeline to reconstruct and extract features from laser scanned data using fuzzy logic concepts.

Surface from Point Cloud. Retrieving a surface from a point cloud is a widely studied problem in computational geometry or reverse engineering. Methods can be split in two groups : constructive methods that are based upon triangulation of the point cloud using Voronoï diagrams or Delaunay triangulations, a description of these methods can be found in [9] or meshless methods that try to fit an implicit function to the point cloud. For the latter, the current state-of-the-art method would be the Poisson surface reconstruction [15]. It is also possible to combine both approaches [4]

Surface Deformation. Surface deformation methods have been studied for many years. A recent state of the art and course has been made by Sorkine and Botsch [21]. Two classes of deformations can be defined : surface deformations and space deformations. A focus on linear surface deformation has been made in [8], while surveys on space deformation can be found in [5] and [11]. The former ones consist in finding a displacement function defined over the surface whereas the latter look for displacement functions defined in the space in which the surface is embedded (that is $\mathbb{R}^3$ in most of cases). Each of these classes comprises linear or non-linear approaches. Surface deformation methods lead to solving a linear system, often symmetric definite positive, whose size depends on the representation of the surface (number of vertices in a mesh for example). Once the regions allowed to move are fixed the system can be pre-factorized off-line which enables fast deformations since only back-substitution is needed. Nevertheless not all surface representations can be used because differential properties need to be computable on the representation. On the other hand, space deformation methods are able to deal with a lot of surface representations and their complexity is independent of the surface representation. When using space deformation, one needs to define a control object that will be edited to compute the deformation. In order to achieve deformation of complex shapes like a human body, especially around joints, the best technique would be to use cage-based deformation such as proposed by [18] but it requires to build a complex control object and constraining the position of a specific vertex in the mesh is not as easy as with surface deformation. Among all the methods, the use of linear or non-linear approaches is related to the quality of the conservation of details (i.e. differential properties). In our system precise conservation of the details of the generic model are not useful since we want to fit a target model with a different shape so a linear method was chosen.

Model Based Approaches. Interactive model-based approaches were proposed for head [14] and complete body [20], they rely on the placement of feature points that were used to build a deformation field using RBF functions. A fully automatic process was performed by Allen et al. [3]. They used the CAESAR database and developed an optimization procedure to fit a template model to all the data. After that they used these reconstructions to make a PCA approach for reconstruction. The main limitation of this method is that it requires a large database of good quality acquisition and the fitting part relied on 74 markers that were put on the subjects which needs to be avoided for our application. Using a model to reconstruct the surface and add additional information for animation was proposed by Moccozet et al. [19]. Reconstruction of a large database is often a first step to build a statistical model of the human body including variations in shape and pose. [12] [3] [6].

2 Our System

In this part we present our system to generate a 3D model of the customer. The scanner is composed of 3 laser lights of different wavelengths filmed by 3 calibrated cameras. Each laser emits a planar beam whose equation is known at each time. The images of the camera are analyzed to find pixels of the laser beam and by combining the plane equation and the calibration information, 3D points are obtained.

Once this point cloud is obtained we start the reconstruction process. The different steps of this process are :

- Point cloud analysis to extract feature points and a rough segmentation
- Adjustment of the pose of the generic model with a linear surface deformation approach using information from the point cloud analysis
- Optimization of the vertices positions to be as close as possible to the point cloud

The following subsection will detail each one of these steps. Let us first define some elements that will be used in the subsections.

The point cloud provided by the scanner is a set $C = (p_{C_i})_{i=1..n_C}$ of n_C points in $\mathbb{R}^3$, we note $(p_{C_i}) = (x_{C_i}, y_{C_i}, z_{C_i})$. Our generic model is a triangulated mesh made by a computer graphics designer respecting all the constraints for cloth simulation. The 3D mesh is defined by $M = (V, F)$ and a function p where V is the set of vertices. We can assume that $V = \{0; ..; n_V - 1\}$ where n_V is the number of vertices. F is a set of n_F triplets of vertices of V defining the triangles of the manifold. p is the function of the 3D realization of the mesh associating a 3D position in space to each vertex of V:

$$p : \begin{cases} V \rightarrow \mathbb{R}^3 \\ v \ \mapsto p(v) = (x(v), y(v), z(v)) \end{cases}$$

To make it simpler we will write $p_i = p(i) = (x_i, y_i, z_i)$

2.1 Point Cloud Analysis

The first part of the process is to extract information from the point cloud in order to adjust the pose of the generic model. Feature point definition and detection can be a very complex task since, for a same body part such as elbow, wrist ... different users may not always choose the same point in the point cloud. Mathematical description of many feature points is not an easy task and supposes often a strong regularity of the surface to be computed which is hard to obtain since there are always holes due to occlusions or parts of the body that weren't

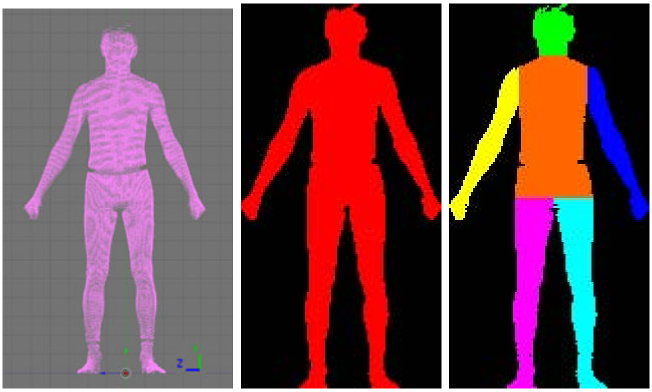


Fig. 1. Left : a point cloud provided by the scanner including axis orientations, the person stands in the standard position, middle : a binary image constructed by projecting the points in the Oyz plane and filtered by a median filter, right : results of the segmentation

reached by the light of the lasers. For these reasons we will only use feature points to adjust the pose of the model and provide a good initialization for an optimization procedure, we will not use these points as strong constraints on specific points of the point cloud.

We introduce some elements useful for the analysis of the mesh. Axis orientations are defined as in Fig 1. For $n \in \mathbb{N}$, let $I_n = \{i \in \mathbb{N}, 0 \leq i \leq n-1\}$. Let $x_CMax = \max_{i \in I_{n_C}} xC_i$ and $x_CMin = \min_{i \in I_{n_C}} x_{C_i}$. In a same manner we define y_CMax, y_CMin, z_CMax and z_CMin. We will use slices of the point cloud defined as follow :

$$slice_y(i) = \{p_{C_j} \in C, y_CMin + i.h_y \leq y_{C_j} < y_CMin + (i+1).h_y\}$$

We use a similar definition for z axis. h_y and h_z represent the thickness of the slices along y and z axes. The choice of these thicknesses is made as a compromise between the desired precision and the density of points provided by the scanner. In our case we found $h_z = 3mm$ and $h_y = 10mm$ to be good enough values.

Finally we construct a binary image that will help us in finding feature points and segmenting the points cloud. We project all the points of C in the Oyz plane. We digitize this plane into pixels and put a binary value 1 when a point of C is in the pixel and 0 otherwise. Our aim is to obtain a single component without holes as shown in Fig 1 on the right. Therefore we chose a fixed value of digitization along the z axis (100 pixels) the value along y axis is computed for each acquisition so that the aspect ratio of the point cloud is preserved. Such an image is shown in Fig 1 on the right.

Crotch Detection. To find the crotch position we use the slices along the z axis. Let us define the following sequence $(y_{min|z})$:

$$y_{min|z}(i) = \min_{p_{C_j} \in slice_z(i)} \{y_{C_j}\}$$

According to the reference position the customer is supposed to have, we can say that the crotch is located in a slice at a local maximum of this sequence whose positions are roughly around $\frac{z_C Min + z_C Max}{2}$. So we look for the highest maximum of the sequence around this value to find the crotch height.

Shoulder. Using a point around the armpit is a difficult problem because if the arm is too close to the torso the point detected will be too low and this will probably lead to deformation artefacts. To avoid this problem we use a feature point in the shoulder area. First, we use the binary image to find an approximate 2D position of the armpit. We start from a pixel in the center column of the image roughly at the neck height, this pixel belongs to the mask and has therefore a binary value of 1. To find the right armpit approximate position we start to move left from this pixel and we analyse the binary sequence, as long as the binary sequence is a list of 1's followed by 0's we move to the next line and start again from one pixel lower. Once we reach a line with a sequence of 1's followed by one or a few 0's and 1's again we can say that we found a transition between the torso and the right arm. This would be our rough armpit position. As we said this position is not reliable for pose approximation, so starting from this point we move up in the image (we have a sequence of 1's) until we reach a 0. This gives us a position on the shoulder juste above the armpit that will be our feature point. The left feature point position is found in a similar way.

Rough Segmentation. Now that we have crotch and armpit estimation we can obtain a segmentation of the image : points that are under the crotch are split between left and right and set to right and left leg. Then the armpit positions are used to separate the arms. See Fig 1 for the result. This segmentation can be used to segment the point cloud just by looking at the label of the pixel for a specific point.

Wrists and Ankles. Since in our reference position the subject has his two feet on the ground, to adjust the pose we only need a rough position around the ankle so we take the points in $slice_y(i_{ankle})$ with i_{ankle} such that $y_C Min + i_{ankle}.h_y \leq 12cm < y_C Min + (i_{ankle} + 1).h_y$. We split $slice_y(i_{ankle})$ between points that have a positive value of z and others (left and right legs actually) we take the center of gravity of these subsets as feature points for the left and right ankle.

Wrists positions are estimated as the barycenters of the slices with the smallest area in the lower part of each arm.

Chin and Nose. The only point we use as a feature point is the nose but for the detection we look at slices along the y axis and study the sequence :

$$x_{max|y}(i) = \max_{p_{C_j} \in slice_y(i)} \{x_{C_j}\}$$

Chin and nose can be regarded as local maxima in this sequence. Once the slice is found we take the point with the largest value of x in the slice as feature point.

2.2 Pose Adjustment

Once the feature points have been extracted, pose adjustment of the generic model can be performed. To do this task we propose to use a linear surface deformation method. Let's first recall some paradigm of surface editing. The main idea of surface editing is to define handle regions on the surface that the user can manipulate, fixed region that will not move and free regions that will be deformed in a smooth manner when the user manipulates handle regions. An easy way to compute such a deformation is to use a differential representation of the mesh and solve a minimization problem to preserve as much as possible the differential representation while enforcing some constraints on the position of a few vertices. In our framework we chose to use Laplacian coordinates. Laplacian coordinates are evaluated at each point of the mesh using a discrete Laplace operator. The discretization of the Laplacian operator has been well studied and has led to a variety of formulas. The general formulation is :

$$\delta_i = \Delta(p_i) = \sum_{j \in N(i)} w_{ij}(p_i - p_j)$$

where $N(i)$ is the 1-ring neighborhood of the vertex i. The choice of w_{ij} depends on the properties we want for the coordinates see [23] for a discussion, these values desribe the Laplacian matrix L. We look for new positions of the vertices p'_i, the minimization problem is formulated as :

$$\min_{p'} \sum \|\Delta({p'}_i) - \delta_i\|^2$$

This leads us to solve the bi-Laplacian equation

$$L^2 p' = L\delta$$

There are two ways of enforcing constraints to this system. The soft way is to add other energy terms in the minimization $\|{p'}_i - p_{i_{constraint}}\|^2$. The hard way is to remove rows and columns corresponding to constrained points and transfer the values to the right-hand side.

In this system we use a uniform discretization of the Laplacian (i.e. $w_{ij} = \frac{1}{|N(i)|}$) that means that δ_i represents the difference between p_i and the center of gravity of its neighbors. We also use soft constraints.

We defined 2 sets of handle regions on the generic model, see Fig 2. The first set is composed of points that will be moved using the feature points detected in the point cloud analysis. The second set is composed of slices along each member and will help to adjust the volume of the model to fit the point cloud in a first approximation. We pre-compute the matrices for these sets of constraints off-line, so that we only need to build the right-hand side.

In a first step we use only the set related to feature points. For each region of this set we use a feature point to find a transformation, we use only translation and scale transformations. Translation is often obtained by aligning the center of gravity (ankles, wrists) or an extreme point (nose, crotch) of the region in

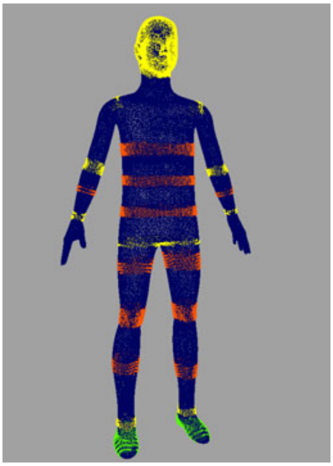


Fig. 2. Generic model with handle sets. In yellow, the handles related to a specific feature point, in red the ones used to retrieve volumetric information.

the generic model with the feature point detected. Scale is evaluated by taking a slice in the concerned member along the y axis in the point cloud around the feature point and computing its bounding box. This box is matched with the one of the handle region of the generic model.

In a second step we consider the deformed generic model after the first step, we pick slices in the segmented point cloud around each handle and find a scale deformation by matching the bounding boxes.

Now we have a deformed model in the same pose as the scanned customer and with roughly the same volume everywhere, we use it as the initialization of an optimization procedure.

2.3 Vertex Position Optimization

To get more precisely the shape of the customer we optimize the position of the vertices assuming we have a good initialization provided by the pose adjustment of the previous section.

We define an energy function over the mesh vertices with 2 terms, the first one E_{data} that moves the vertices towards the point cloud and the second one E_{smooth} that ensures smoothness :

$$E(V) = E_{data} + E_{smooth} = \sum_{i=0}^{n_C - 1} d(p_{C_i}, M)^2 + \sum_{v \in V} \|\nabla(v)\|^2$$

with $d(p, M)$ the distance between a 3D point p and the mesh M, we choose the distance to the projection of point p on its closest facet $f(p)$ in M. $f(p)$ is found using a fast search structure. Once we have it the distance can be expressed relatively the the vertices positions of M. Let us note $v_{f(p)1}$, $v_{f(p)2}$, $v_{f(p)3}$ the vertices of $f(p)$ and $n_{f(p)}$ the normal of the facet $f(p)$: $n_{f(p)} = (v_{f(p)2} - v_{f(p)1}) \wedge (v_{f(p)3} - v_{f(p)1})$. The distance is then :

$$d(p, M) = \frac{n_{f(p)}.(p - v_{f(p)1})}{\|n_{f(p)}\|}$$

Expressing $d(p, M)$ relatively to the vertices of M enables us to optimize E using a gradient descent approach.

Applying this method directly on the mesh yet leads to unsatisfactory results because the generic mesh is too dense and wrong displacements in the first iterations can lead to considerable artefacts. To deal with this problem we use a multiresolution strategy : we have a precomputed lower resolution of the generic model and its relation to the higher resolution. We first optimize the lower resolution mesh, then reconstruct the higher resolution with the new position and use it as a new initialization for the optimization.

3 Implementation, Results and Future Works

The whole process was implemented in C++, the sparse linear system for pose adjustment is solved using UMFPACK library [10].

Reconstruction of the point cloud of Fig 1 is shown in Fig 3.

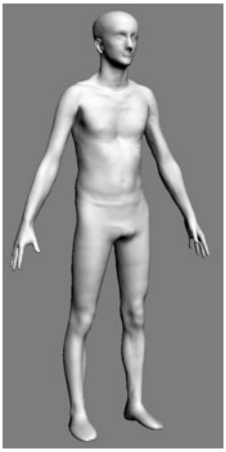


Fig. 3. Our reconstruction of the point cloud in fig 1

To measure the validity of the approach we compare measures on our reconstructed model with another state-of-the-art method for surface reconstruction from point cloud [15], we use both methods to reconstruct a computer-made 3D model of a human body by taking only the points of the mesh so that we

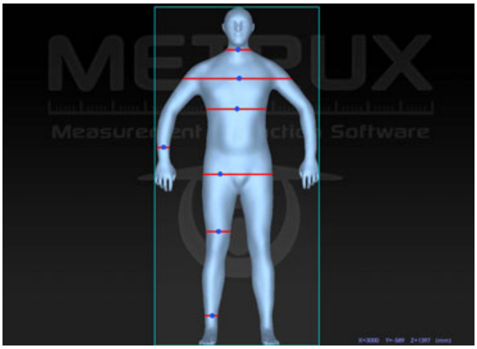


Fig. 4. The computer made model used as ground truth. Red lines show heights where measures are taken.

Table 1. Comparison of our method with a ground truth 3D model and Poisson surface reconstruction [15]. Circumferences are measured at different heights y on the models see Fig 4.

	Our approach	Poisson	Ground Truth	our % error	Poisson % error
y = 1540mm	425mm	432mm	422mm	0.71	2.37
y = 1390mm	1318mm	1325mm	1318mm	0.0	0.53
y = 1230mm	942mm	954mm	944mm	0.21	1.06
y = 1030mm	208mm	202mm	204mm	1.96	0.98
y = 890mm	1022mm	1033mm	1025mm	0.29	0.78
y = 590mm	425mm	440mm	426mm	0.23	3.29
y = 150mm	241mm	247mm	241mm	0.0	2.49

don't introduce errors due to the acquisition process and we compare the measures with the different reconstructions. We choose to measure circumferences at various heights since these are most likely the important values for a virtual try-on application. Percentage of error is calculated for each method relatively to the ground truth provided by the original mesh, see Table 1. Location of the measures on the original mesh are shown in Fig 4 For Poisson reconstruction we chose a depth parameter so that the generated mesh has a similar number of vertices as our generic model ($depth = 12$)

Processing time for the whole reconstruction is around 20 seconds on a modern laptop computer.

Our method provides results comparable to recent techniques of surface reconstruction from point cloud in terms of speed and precision. Yet some parts of the human body are still hard to reconstruct (hair and hands especially) for two reasons : the acquisition system has to be able to provide points for these area which is not always possible for hair with current technologies, and the reconstruction procedure has to be able to get fine details compared to the global size of the subject.

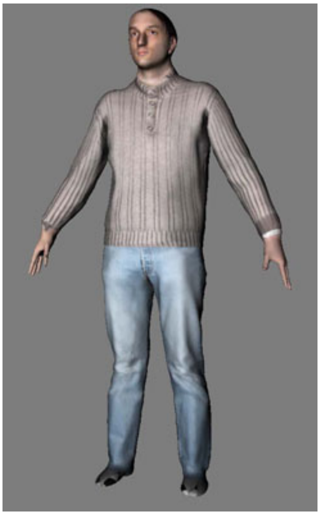


Fig. 5. Another reconstructed model with texture information

To complete the surface information we plan to add texture information that will be captured from calibrated cameras during the scan. Applying a state-of-the-art method such as [16] already gives promising results as shown in Fig 5, but further work needs to be done with illumination variations between images, and seam visibility in critical areas like the face.

Acknowledgements. The authors wish to thank Kévin Leduc for designing the generic model. We'd also like to thank Misha Kazhdan for making his software available for Poisson Surface Reconstruction.

References

1. http://www.optitex.com/
2. http://www.3douest.com/
3. Allen, B., Curless, B., PopovićThe, Z.: space of human body shapes: reconstruction and parameterization from range scans. In: SIGGRAPH 2003: ACM SIGGRAPH 2003 Papers, pp. 587–594. ACM, New York (2003)
4. Alliez, P., Cohen-Steiner, D., Tong, Y., Desbrun, M.: Voronoi-based variational reconstruction of unoriented point sets. In: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, pp. 39–48. Eurographics Association, Aire-la-Ville (2007)
5. Angelidis, A., Singh, K.: Space deformations and their application to shape modeling. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Courses, pp. 10–29. ACM, New York (2006)
6. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, SIGGRAPH 2005, pp. 408–416. ACM, New York (2005)

7. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, pp. 43–54. ACM, New York (1998)
8. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE Transactions on Visualization and Computer Graphics 14(1), 213–230 (2008)
9. Cazals, F., Giesen, J.: Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. Technical Report RR-5393, INRIA, 11 (2004)
10. Davis, T.A.: Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. 30, 196–199 (2004)
11. Gain, J., Bechmann, D.: A survey of spatial deformation from a user-centered perspective. ACM Trans. Graph. 27(4), 1–21 (2008)
12. Hasler, N., Stoll, C., Sunkel, M., Rosenhahn, B., Seidel, H.-P.: A statistical model of human pose and body shape. In: Dutr'e, P., Stamminger, M. (eds.) Computer Graphics Forum (Proc. Eurographics 2008), Munich, Germany, vol. 2 (March 2009)
13. Hilsmann, A., Eisert, P.: Tracking and retexturing cloth for real-time virtual clothing applications. In: Mirage 2009 - Computer Vision/Computer Graphics Collaboration Techniques and Applications, Rocquencourt, France (May 2009)
14. Kalinkina, D., Gagalowicz, A., Roussel, R.: 3D reconstruction of a human face from images using morphological adaptation. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 212–224. Springer, Heidelberg (2007)
15. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP 2006, pp. 61–70. Eurographics Association, Aire-la-Ville (2006)
16. Lempitsky, V., Ivanov, D.: Seamless mosaicing of image-based texture maps. In: CVPR 2007, pp. 1–6 (2007)
17. Leong, I.-F., Fang, J.-J., Tsai, M.-J.: Automatic body feature extraction from a marker-less scanned human body. Comput. Aided Des. 39, 568–582 (2007)
18. Lipman, Y., Levin, D., Cohen-Or, D.: Green coordinates. ACM Trans. Graph. 27(3), 1–10 (2008)
19. Moccozet, L., Dellas, F., Magnenat-thalmann, N., Biasotti, S., Mortara, M., Falcidieno, B., Min, P., Veltkamp, R.: Animatable human body model reconstruction from 3d scan data using templates. In: Proceedings of Workshop on Modelling and Motion Capture Techniques for Virtual Environments, CAPTECH 2004, pp. 73–79 (2004)
20. Quah, C.K., Gagalowicz, A., Roussel, R., Seah, H.S.: 3D modeling of humans with skeletons from uncalibrated wide baseline views. In: Gagalowicz, A., Philips, W. (eds.) CAIP 2005. LNCS, vol. 3691, pp. 379–389. Springer, Heidelberg (2005)
21. Sorkine, O., Botsch, M.: Tutorial: Interactive shape modeling and deformation. In: Eurographics (2009)
22. Wang, C.C.L., Chang, T.K.K., Yuen, M.M.-F.: From laser-scanned data to feature human model: a system based on fuzzy logic concept. Computer-Aided Design 35(3), 241–253 (2003)
23. Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E.: Discrete laplace operators: no free lunch. In: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, pp. 33–37. Eurographics Association, Aire-la-Ville (2007)
24. Werghi, N.: Segmentation and modeling of full human body shape from 3-d scan data: A survey. SMC-C 37(6), 1122–1136 (2007)

Content Based Image Retrieval Using Visual-Words Distribution Entropy

Savvas A. Chatzichristofis, Chryssanthi Iakovidou, and Yiannis S. Boutalis

Department of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi 67100, Greece
{schatzic,ciakovid,ybout}@ee.duth.gr

Abstract. Bag-of-visual-words (BOVW) is a representation of images which is built using a large set of local features. To date, the experimental results presented in the literature have shown that this approach achieves high retrieval scores in several benchmarking image databases because of their ability to recognize objects and retrieve near-duplicate (to the query) images. In this paper, we propose a novel method that fuses the idea of inserting the spatial relationship of the visual words in an image with the conventional Visual Words method. Incorporating the visual distribution entropy leads to a robust scale invariant descriptor. The experimental results show that the proposed method demonstrates better performance than the classic Visual Words approach, while it also outperforms several other descriptors from the literature.

1 Introduction

Over the years, a great number of approaches have been introduced in the field of content-based image retrieval (CBIR). Multiple features can be abstracted in order to obtain an efficient description of the visual content of an image. According to this approach, the visual content of the images is mapped into a new space named the feature space. Beginning with the so called global features, images can be described via a general single vector, conducing to a first rough classification. A feature is a set of characteristics of the image, such as color, texture and shape.

Trying to achieve successful content-based image retrieval exclusively via global features often proves to be rather challenging since the output depends on the image queries. CBIR with global features is notoriously noisy for image queries of low generality, i.e. the fraction of relevant images in a collection. In contrast to text retrieval where documents matching no query keyword are not retrieved, CBIR methods typically rank the whole collection via some distance measure [2]. If, for example, a query image depicts the plan of a white plate on a black background, due to the common features (round-shaped white foreground towards black background) that are met in a great number of images, the early ranked results may be dominated by non-plate depicting images.

Local-feature approaches provide a slightly better retrieval effectiveness than global features [1]. They represent images with multiple points of interest in a feature space

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 204–215, 2011.

in contrast to single-point global feature representations. While local approaches provide more robust information, they are more expensive computationally due to the high dimensionality of their feature spaces and usually need nearest neighbors approximation to perform points-matching [16]. High-dimensional indexing still remains a challenging problem in the database field. Thus, global features are more popular in CBIR systems as they are easier to handle and still provide basic retrieval mechanisms. In any case, CBIR with either local or global features does not scale up well to large databases efficiency-wise. In small databases, a simple sequential scan may be acceptable, however, scaling up to millions or billion images efficient indexing algorithms are imperative [11].

In order to surpass the aforementioned difficulty the Bag-of-visual-words (BOVW) [6] approach is adopted. BOVW is inspired directly by the bag-of-words model, a well-known and widely used method in text retrieval, where a document is represented by a set of distinct keywords. The same concept governs the BOVW model, in which an image is represented by a set of distinct visual words derived from local features. BOVWs are fast becoming a widely used representation for content-based image retrieval, for mainly two reasons: their better retrieval effectiveness over global feature representations on near identical images, and much better efficiency than local feature representations. However, experimental results of reported work show that the commonly generated visual words are still not as expressive as the text words [22]. When employing this approach, the extracted local features are clustered using k-means classifier and the computed cluster centers (i.e. the mean vectors) are called visual words. The set of visual words forms a visual vocabulary also known as codebook. For every new image added in the collection its local features must be extracted and assigned to the best fitting visual word from the existing codebook. By the end of that process a local feature histogram is composed for each image in the collection. The size of the codebook is directly related to the k-means clustering step and the filtering parameters that were set. Determining the appropriate size of the codebook is essential but very difficult to predict. Ideally, a small-sized codebook, which would allow fast identification and search tasks, is desired. However, while a small vocabulary may produce the expected results in some image collections, it proves to be inefficient in others due to its low discriminating abilities. On the other hand, a wider codebook often contains redundant visual words which results not only in increasing the computational cost caused by the high dimensionality of the produced local feature vectors but also in some cases forces the early ranked positions to be filled with spurious results. Multiple approaches to enhance the bag of words approach have been proposed in literature [10].

In several CBIR systems with global features, a feature can further be enriched with information about the spatial distribution of the characteristic, that it describes. In the following, a new techniques is proposed to incorporate to the BOVW the distribution state of each single visual word in the spatial dimension. This method revises BOVW approach by applying information about Distribution Entropy (DE) [20] of the visual words in the image. Fusing these two techniques allowed us to result in a promising model for CBIR, which is easy to implement and presents well ranked relative retrieval results. In Section 2 we briefly review relative literature about spatial distribution information in several low level features.

The rest of the paper is organized as follows. In Section 3 we present the proposed CBIR approach. Section 4 provides the details on the used image database, the similarity measure and the performance evaluation. The experimental results are described and compared with other methods and finally, Section 5 gives an overall conclusion.

2 Spatial Distribution Information

Due to the statistical nature of several global features, they can only index the content of images in a limited way. To make these features more effective for image indexing, spatial information should be considered. In [8] the authors proposed a technique of integrating color information with spatial knowledge to obtain an overall description of the image. This technique involves three steps: the selection of a set of representative colors, the analysis of spatial information of the selected colors, and the retrieval process based on the integrated color-spatial information. Stricker et al [19] partition an image into 5 partially overlapping, fuzzy regions. From each region in the image they extract the first three moments of the color distribution and store them in the index. The feature vectors in the index are relatively insensitive to small translations and rotations.

Pass et al [15] described the concept of color coherent vector (CCV) and use it to separate a color histogram vector into two parts: a coherent vector and a non-coherent vector. A pixel is called coherent if its connected component is large enough. A CCV of an image is the histogram over all coherent pixels of the image. In [9] a color correlograms method is proposed, which collects statistics of the co-occurrence of two colors. A simplification of this feature is the autocorrelogram, which only captures the spatial correlation between identical colors.

The MPEG-7 standard includes the Color Layout Descriptor [13], which represents the spatial distribution of color of visual signals in a very compact form. The CLD uses representative colors on an 8×8 grid followed by a Discrete Cosine Transform and encoding of the resulting coefficients. Spatial Color Distribution descriptor (SpCD) [4] is a recently proposed compact composite descriptor (CCD) which combines color and spatial color distribution information. In order to extract the color information, a fuzzy system is being used, which is maps the number of colors that are included in the image into a custom palette of 8 colors. The way by which the vector of the proposed descriptor is being formed, describes the color spatial information contained in images.

Rao et al [17] introduced annular color histogram. In this method the centroid C_i and the radius r_i of each color bin in the histogram are calculated. C_i serves as the center of N concentric circles with nr_i radius, where $1 \leq n \leq N$. This division allows us to count the number of pixels of a color bin in each n circle providing important spatial information which however is size variant since it is relative to the number of pixels of a color bin in the annular circles.

The Spatial-Chromatic Histogram (SCH) proposed by Cinque et al [5], describes how identical color pixels are distributed in an image and was found to be more efficient than the annular color histogram due to its smaller index. However, SCH uses the standard deviation σ to measure the square root of the average squared distance of pixels in a bin from the computed centroid of the bin. This means that σ is size variant and when this method is used in CBIR it can ultimately lead to falsely ranked results.

Sun et al [20] proposed a new Color Distribution Entropy (CDE) descriptor which describes the spatial information of an image and is based on the Normalized Spatial Distribution Histogram (NSDH) and the definition of entropy. This method presents low dimension indexes and is therefore very efficient in CBIR. Furthermore, the NSDH is size invariant because annular color histograms are normalized by the number of pixels of the color bins. Thus, CDE is also size invariant. More details about annular histograms and CDE are given in Section 3.

3 Visual Words Distribution Entropy

In this paper we propose a novel method for content-based image retrieval that is based on the BOVW method using the SURF [3] descriptors to produce the visual vocabulary and the CDE method to enhance the visual words histogram with a local spatial relationship component. In the proposed method a predefined number of Annular Visual Words are used to form the codebook for image classification and identification.

3.1 Annular Visual Words Histogram

In this section we present the structure of the proposed method. The block diagram in Figure 1 depicts the different implementation stages. Initially, in order to produce the visual codebook, we use a set of 237434 images from the ImageCLEF 2010 Wikipedia test collection. SURF descriptors are extracted from these images. Then, we randomly select 100000 descriptors to create the visual words that will form our codebook. SURF descriptors are clustered and the mean vector (using k-means) is used as a visual word. For our experiment we used two different scenarios with 128 and 256 sized codebooks respectively.

For every query image its SURF features are extracted. The local features are assigned to the best fitting visual word using the nearest neighbor method from the earlier created codebooks. A visual words histogram can now be computed for each image.

The spatial information of the visual words is incorporated using the following method: Based on the annular color histogram generation, we introduce the annular visual words histogram. Let A_i be the count of SURF descriptors belonging in the visual word i. Let $C_i = (x_i, y_i)$ be the centroid of the visual word i, x_i and y_i defined as:

$$x_i = \frac{1}{A_i} \sum_{(x,y)\in A_i} x; \tag{1}$$

$$y_i = \frac{1}{A_i} \sum_{(x,y)\in A_i} y \tag{2}$$

Let r_i be the radius of the visual word i:

$$r_i = \max_{(x,y)\in A_i} \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{3}$$

We divide each radius r_i into $N = 3$ and draw 3 concentric circles centered at C_i, creating three different image areas. A_{ij} is the count SURF features belonging to the

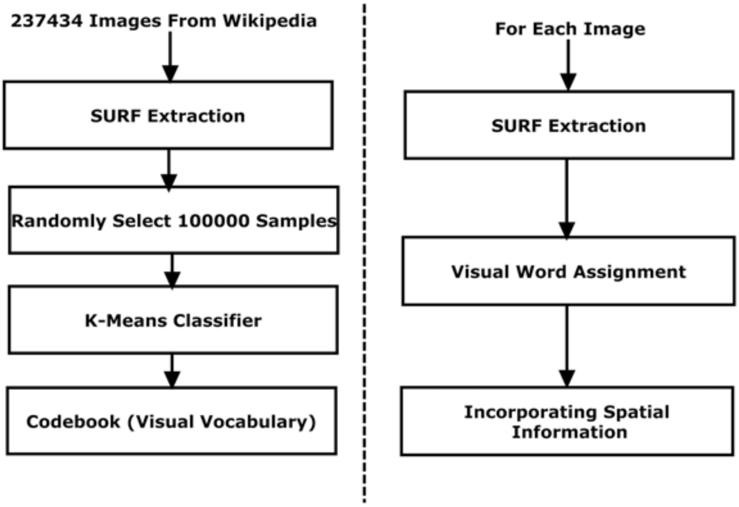

237434 Images From Wikipedia
SURF Extraction
Randomly Select 100000 Samples
K-Means Classifier
Codebook (Visual Vocabulary)
For Each Image
SURF Extraction
Visual Word Assignment
Incorporating Spatial Information

Fig. 1. Blog diagram of the Annular Visual Words implementation

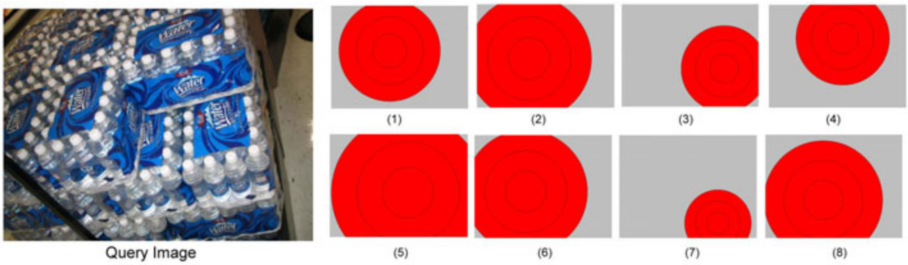


Fig. 2. The first 8 annular visual words for a query image

visual word i inside circle j. Figure 2 illustrates the first 8 annular visual words for an image.

At this point we incorporate the visual words distribution entropy (VWDE). This method is based on the normalized spatial distribution histogram (NSDH) according to which the annular color histogram (or in our case the annular visual words histogram) can be defined as P_i where:

$$P_i = (P_{i1}, P_{i2}, \ldots, P_{iN}) \tag{4}$$

where

$$P_{ij} = |A_{ij}|/|A_i| \tag{5}$$

The VWDE of a visual word i can be defined as:

$$E_i(P_i) = -\sum_{j=1}^{N} P_{ij} \log_2(P_{ij}) \tag{6}$$

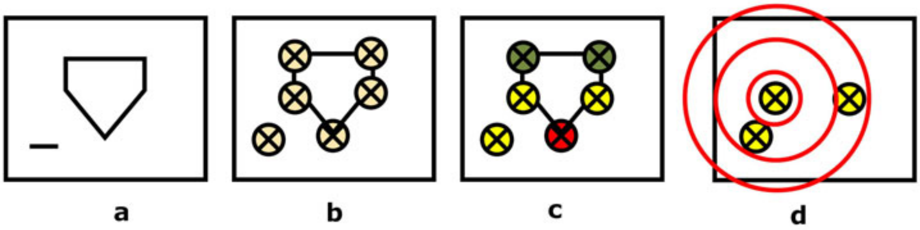


Fig. 3. A visual example of the proposed method a. input image b. extraction of the local features c. visual words assignment d. indexing of the ”yellow” visual word of the annular visual words histogram.

which gives a dispersive degree of the pixel patches of a bin in an image. The VWDE index for an image can be written as:

$$(A_1, E_1, A_2, E_2, \dots, A_n, E_n) \quad (7)$$

Where h_i is the number of the SURF features that belongs to the visual word i, E_i is the VWDE of these features i and n is the number of the visual words of the codebook.

Figure 3 depicts a quick visual example of the method. Figure 3.a is the input image, Figure 3.b depicts the extraction of the points of interest, Figure 3.c depicts the assignment of the key points to three different visual words (yellow, green, red) and finally Figure 3.d illustrates the indexing of annular visual words histogram of the yellow-assigned visual words of the image.

The SURF descriptors of the visual word i are more dispersed in the far annular circles than those closer to the centroid. The closer (in a spatial sense) the descriptors are found together the more possible it is for them to belong to the same object. This can and should be taken under consideration in order to strengthen the effect of descriptors found in the near to the centroid circles and, correspondingly, weaken the effect of those in the farther circles. We adopt the weight function $f(j)$ proposed in [20]. The weight function should satisfy $f(j_1) > f(j_2)$ when annular circle j_1 is out of j_2. With,

$$f(j) = 1 + \frac{j}{N} \quad (8)$$

equation 6 can be written as:

$$E_i(P_i) = -\sum_{j=1}^{N} f(j) P_{ij} \log_2(P_{ij}) \quad (9)$$

In order to remove the influence of the symmetrical property of entropy that forces perceptually dissimilar histograms to present the same entropy, Sun et al [20] proposed an improved formula for the computation of color distribution entropy (I-CDE). The improved formula is based on the observation that even though perceptually dissimilar histograms may have the same entropy they have different histogram areas. The area of histogram H is defined as,

$$A(H) = \sum_{i=1}^{n} (p_i \times i) \tag{10}$$

and

$$H = \{p_1, p_2, \ldots, p_n\} \tag{11}$$

And the weight function that allows the discrimination of the different areas is given by:

$$g(H) = 1 + \frac{A(H)}{n} \tag{12}$$

Applying this new weight function on equation 9 the I-VWDE function is described as:

$$E_i(P_i) = -g(P_i) \sum_{j=1}^{N} f(j) P_{ij} \log_2(P_{ij}) \tag{13}$$

Similar to our method, Ding et al [7] proposed a video annotation method based on an annular spatial partition scheme. In this approach the spatial partition scheme is based on the distribution of the overall points of interest. The centroid of the annular regions is computed according to the distribution of the keywords found in an image via the SIFT [12] descriptors. The centroid serves as the center of three concentric circles that define three annular regions. For each region the BOW histogram is computed and the three regional histograms are used to produce an overall feature vector. The main difference regarding our method is that we compute annular visual words histograms for the three regions that are formed per visual word centroid and then we employ the I-VWDE described by function 13, achieving a size invariant method which also removes the influence of the symmetrical property of entropy.

4 Experimental Results

In this section we present our first experimental results. In subsection 4.1 we define the database details, subsection 4.2 presents the similarity measure that was implemented in order to evaluate the image correlation.

4.1 Database Details

The Nister image database consists of K groups of four images each [14]. The image size is set to 640×480 pixels (VGA). Each group of four depicts an image of a single object captured from different angles and in some cases under different light conditions. The first image of every group is used as a query image and only the images from the same group are considered to be relevant. The first subset of 1000 images of the database with 250 queries was used in order to calculate the efficiency of the proposed method.

4.2 Similarity Measure

The distance $D(i, j)$ of two images i and j is defined as:

$$D(i, j) = t(n_i, n_j) \times E(E_i, E_j) \tag{14}$$

where $t(n_i, n_j)$ is the distance between the histogram of visual words calculated using the Tanimoto coefficient:

$$T_{ij} = t(n_i, n_j) = \frac{n_i^T n_j}{n_i^T n_i + n_j^T n_j - n_i^T n_j} \tag{15}$$

and where $E(E_i, E_j)$ is the Euclidean distance between the I-VWDE histograms.

4.3 Performance Evaluation

For the evaluation of the performance of the proposed image retrieval method one of the metrics we employed is the Averaged Normalized Modified Retrieval Rank (ANMRR) [13]. The average rank $AVR(q)$ for query q is:

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank(k)}{NG(q)} \tag{16}$$

Where $NG(q)$ is the number of ground truth images for query q, $K = \min(X_{NG} \times NG(q), 2 \times GTM)$, $GTM = \max(NG)$. If $NG(q) > 50$ then, $X_{NG} = 2$ else $XNG = 4$. $Rank(k)$ is the retrieval rank of the ground truth image. Consider a query and assume that the kth ground truth image for this query q is found at position R. If this image is in the first K retrievals then $Rank(k) = R$ else $Rank(k) = (K + 1)$. The modified retrieval rank is:

$$MRR(q) = AVR(q) - 0.5 \times [1 + NG(q)] \tag{17}$$

The normalized modified retrieval rank is defined as:

$$NMRR(q) = \frac{MRR(q)}{1.25 \times K - 0.5 \times [1 + NG(q)]} \tag{18}$$

and finally the average of NMRR over all queries is computed as:

$$ANMRR(q) = \frac{1}{Q} \sum_{q=1}^{Q} NMRR(q) \tag{19}$$

where Q is the total number of queries. The ANMRR has a range of 0 to 1 with the best matching quality defined by the value 0 and the worst by 1.

Apart from the ANMRR metric, we also evaluated the performance of the method using the Mean Average Precision (MAP) metric:

$$\text{Percision} = P = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} \tag{20}$$

$$\text{Recall} = R = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} \tag{21}$$

The average precision AP is:

$$AP(q) = \frac{1}{N_R} \sum_{n=1}^{N_R} P_Q(R_n) \tag{22}$$

where R_n is the recall after the nth relevant image retrieved and N_R the total number of relevant documents for the query. MAP is computed by:

$$MAP = \frac{1}{Q} \sum_{q \in Q} AP(q) \tag{23}$$

where Q is the set of queries q.

The last evaluation metric that we employ is the Precision at 10 (P@10) and Precision at 20 (P@20) metrics that describe the system's capability to retrieve as many relevant results as possible in the first 10 and 20 ranked positions, respectively. This evaluation of the system's performance is critical for web based retrieval systems where the users are particularly interested in the credibility of the first results.

4.4 Results

In this section we present our experimental results using the 1000 images from the Nister database with 250 queries, described in Subsection 4.1, evaluated by the ANMRR, the MAP, the P@10 and the P@20 metrics described in Subsection 4.3. We also give the corresponding results of the Visual Words method, for comparison reasons. Additionally, we calculate how significant is the performance deviation between the descriptors. Significance test tell us whether an observed effect, such as a difference between two means, or a correlation between two variables, could reasonably occur just by chance in selecting a random sample. This application uses a bootstrap test, one-tailed, at significance levels 0.05, 0.01, and 0.001, against a baseline run.

Through the proposed method we achieved to significantly improve the results of the Visual Words method. In particular, the ANMRR metric appears improved by a percentage of 36.37% in our method with a 128 word vocabulary compared to the corresponding 128 sized vocabulary and the Visual Word method. The improvement is more evident in the 256-visual words vocabulary. The ANMRR metric is improved by 65.69% using the proposed method compared to the corresponding Visual Word method. Figure 4.a and 4.b illustrate the retrieval results for the Annular Visual Words

Table 1. Retrieval effectiveness for Visual Words and Annular Visual Words. Significance-tested with a bootstrap test, one-tailed, at significance levels 0.05 (***), 0.01 (**), and 0.001 (*), against the visual words baseline.

Descriptor	MAP	P@10	P@20	ANMRR
Visual Words (128 Visual Words)	0.7581	0.3192	0.1656	0.2018
Annular Visual Words (128 Visual Words)	0.8115***	0.3396***	0.1792***	0.1480***
Visual Words (256 Visual Words)	0.7351	0.3044	0.1632	0.2293
Annular Visual Words (256 Visual Words)	0.8254***	0.346***	0.1808***	0.1384***

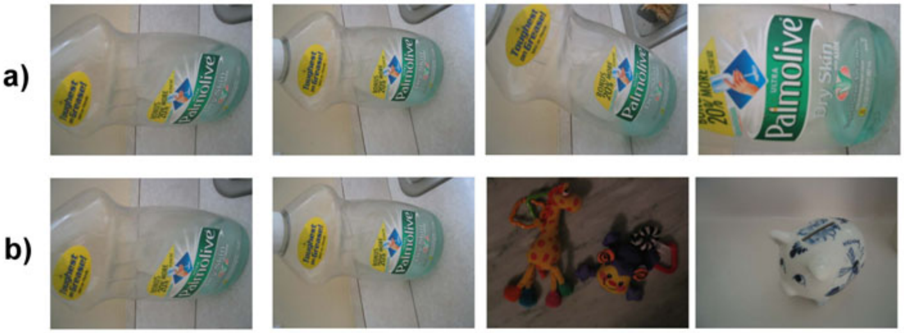


Fig. 4. Experimental results a. the retrieval results in the first four ranked positions using the proposed method b. the retrieval results in the first four ranked positions using the Visual Word method.

Table 2. Retrieval effectiveness for several low level features with spatial distribution information

Descriptor	MAP	P@10	P@20	ANMRR
SpCD	0.8178	0.3408	0.1772	0.1485
AutoCorrelograms	0.7616	0.3192	0.1692	0.1955
CLD	0.7258	0.3084	0.1648	0.2285
TOP-SURF	0.6177	0.2704	0.1498	0.3209

and the Visual Words, respectively. As shown in the example, the AVW methods manages to retrieve all four relevant results ranked in the first four position, while the VW method only retrieves two of the four relevant results in the first four positions.

The following table presents the results from three descriptors with spatial distribution information as well as the results of the recently proposed TOP-SURF[21] visual words descriptor for further comparison with the proposed descriptor.

The proposed method outperforms all four descriptors and according to the ANMRR evaluation metric, our method achieves a 7.30% improvement compared to the SpCD descriptor, a 41.26% improvement compared to the AutoCorrelograms descriptor, a 65.10% improvement compared to the CLD descriptor and an impressive 131.83% improvement compared to the TOP-SURF descriptor.

5 Conclusions

In this paper, we have presented a novel method that fuses the idea of inserting the spatial distribution relationship of the visual words in an image with the conventional Visual Words method. By locating the centroids of the 128 in the first scenario and the 256 in the second scenario different visual words distribution, we computed annular visual words histograms. Incorporation of the visual distribution entropy led to a robust scale invariant descriptor. The experimental results show that the proposed method demonstrates better performance than the Visual Words method, while it also outperforms descriptors such as SpCD, AutoCorrelograms and CLD. Our next step is to thoroughly examine and optimize the similarity matching method in order to further improve our results.

The proposed descriptor is implemented in the image retrieval system img (Rummager)[18] and is available online[1] along with the image databases and the queries.

References

1. Aly, M., Welinder, P., Munich, M.E., Perona, P.: Automatic discovery of image families: Global vs. local features. In: ICIP, pp. 777–780. IEEE, Los Alamitos (2009)
2. Arampatzis, A., Zagoris, K., Chatzichristofis, S.A.: Dynamic two-stage image retrieval from large multimodal databases. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 326–337. Springer, Heidelberg (2011)
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)
4. Chatzichristofis, S.A., Boutalis, Y.S., Lux, M.: SpCD—spatial color distribution descriptor. A fuzzy rule based compact composite descriptor appropriate for hand drawn color sketches retrieval. In: ICAART, pp. 58–63 (2010)
5. Cinque, L., Ciocca, G., Levialdi, S., Pellicanò, A., Schettini, R.: Color-based image retrieval using spatial-chromatic histograms. Image Vision Comput. 19(13), 979–986 (2001)
6. Cula, O.G., Dana, K.J.: Compact representation of bidirectional texture functions. In: CVPR (1), pp. 1041–1047 (2001)
7. Ding, G., Zhang, L., Li, X.: Video annotation based on adaptive annular spatial partition scheme. IEICE Electronics Express 7(1), 7–12 (2010)
8. Hsu, W., Chua, S.T., Pung, H.H.: An integrated color-spatial approach to content-based image retrieval. In: Proceedings of the Third ACM International Conference on Multimedia, pp. 305–313. ACM, New York (1995)
9. Huang, J., Kumar, R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: CVPR, pp. 762–768 (1997)
10. Kogler, M., Lux, M.: Bag of visual words revisited: an exploratory study on robust image retrieval exploiting fuzzy codebooks. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD 2010, pp. 3:1–3:6. ACM, New York (2010)
11. Li, X., Chen, L., Zhang, L., Lin, F., Ma, W.Y.: Image annotation by large-scale content-based image retrieval. In: ACM Multimedia, pp. 607–610. ACM, New York (2006)
12. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
13. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. IEEE Trans. Circuits Syst. Video Techn. 11(6), 703–715 (2001)
14. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR, Citeseer, vol. 5 (2006)
15. Pass, G., Zabih, R.: Histogram refinement for content-based image retrieval. In: IEEE Workshop on Applications of Computer Vision, pp. 96–102 (1996)
16. Popescu, A., Moëllic, P.A., Kanellos, I., Landais, R.: Lightweight web image reranking. In: ACM Multimedia, pp. 657–660 (2009)

[1] www.img-rummager.com

17. Rao, A., Srihari, R.K., Zhang, Z.: Spatial color histograms for content-based image retrieval. In: ICTAI, pp. 183–186 (1999)
18. Chatzichristofis, S.A., Boutalis, Y.S., Lux, M.: Img(rummager): An interactive content based image retrieval sytem. In: 2nd International Workshop on Similarity Search and Applications (SISAP), pp. 151–153 (2009)
19. Stricker, M., Dimai, A.: Color indexing with weak spatial constraints. In: SPIE Proceedings, vol. 2670, pp. 29–40 (1996)
20. Sun, J., Zhang, X., Cui, J., Zhou, L.: Image retrieval based on color distribution entropy. Pattern Recognition Letters 27(10), 1122–1126 (2006)
21. Thomee, B., Bakker, E.M., Lew, M.S.: Top-surf: a visual words toolkit. In: ACM Multimedia, pp. 1473–1476 (2010)
22. Zhang, S., Tian, Q., Hua, G., Huang, Q., Li, S.: Descriptive visual words and visual phrases for image applications. In: ACM Multimedia, pp. 75–84 (2009)

Video Summarization Using a Self-Growing and Self-Organized Neural Gas Network

Dim P. Papadopoulos, Savvas A. Chatzichristofis, and Nikos Papamarkos

Department of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi 67100, Greece
{dimipapa4,schatzic,papamark}@ee.duth.gr

Abstract. In this paper, a novel method to generate video summaries is proposed, which is allocated mainly for being applied to on-line videos. The novelty of this approach lies in the fact that the video summarization problem is considered as a single query image retrieval problem. According to the proposed method, each frame is considered as a separate image and is described by the recently proposed Compact Composite Descriptors(CCDs) and a visual word histogram. In order to classify the frames into clusters, the method utilizes a powerful Self-Growing and Self-Organized Neural Gas (SGONG) network. Its main advantage is that it adjusts the number of created neurons and their topology in an automatic way. Thus, after training, the SGONG give us the appropriate number of output classes and their centers. The extraction of a representative key frame from every cluster leads to the generation of the video abstract. A significant characteristic of the proposed method is its ability to calculate dynamically the appropriate number of clusters. Experimental results are presented to indicate the effectiveness of the proposed approach.

1 Introduction

In the last decades, observing the increasingly use of multimedia data, it is realized that they have penetrated in our everyday life. A characteristic example of multimedia data is the digital video, whose on-line use, especially the last years, has been increased dramatically.

This fact automatically entails that video web sites have become overcrowded and the amount of data has reached to an uncontrollable point. It is no coincidence that in August 2008 YouTube was considered to be the world's second search engine[1] while in 2010, more than 2 billion videos watched per day on-line[2]. Consequently, the situation necessitates the generation of a representative video abstraction with a view to facilitating the user to decide rapidly and easily whether or not he/she is interested in a video without the need to watch the entire video but only the essential content of it.

Over the last years a noteworthy amount of work in the field of video summarization has been observed (e.g. [22,29,21,18,4]). In the literature a lot of significant approaches

[1] http://tinyurl.com/yz5wb8x

[2] http://www.focus.com/images/view/48564/

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 216–226, 2011.

of this issue are demonstrated. Nevertheless, in a recent survey [12] the authors conclude that "video abstraction is still largely in the research phase". In [27] the authors conclude also that "practical applications are still limited in both complexity of method and scale of deployment". The main idea behind video summarization is to take the most representative and most interesting segments of a long video in order to concatenate it to a new, smaller, sequence.

Truong et al.[27] proposed two basic forms of video summaries: key frames and video skims. Key Frames, also called representative frames or R-frames is a collection of salient images extracted from the underlying video source. Video skims, also called a moving-image abstract, moving storyboard, or summary sequence consists of a collection of video segments (and corresponding audio) extracted from the original video. One popular kind of video skim in practice is the movie trailer. Both forms of generating a video summary are presented in a method that is based on clustering all the frames of a video and extracting the key frames of the most optimal clusters and then the preview is formed using the video shots that the key frames belong to [15]. It is a fact that the majority of techniques, in which the summarization of a video is aimed, are focused on the extraction of key frames instead of the preview of the video.

Video summarization methods can also be separated by the low-level features which are used for content analysis[20]. In general, video summarization is either performed by low level image features (e.g. [6]), audio features (e.g. [28]), textual elements (e.g. [10]), or a fusion of several features (multimedia/multimodal methods, e.g. [21]). Regarding low level image features, authors in [20,19] created a key frame selection tool, which implements summarization of video clips by key frame extraction based on several global and local image features.

In this paper, we propose a new key frame extraction approach using low level features from the visual content of the image that expands the problem of video summarization to a problem of single query image retrieval. More particularly, the method utilizes the recently proposed Compact Composite Descriptors (CCDs). The effectiveness of CCDs against to several global low level features for video summarization has been illustrated in [19]. Additionally, the proposed method utilizes a visual words (VW) histogram [11]. VWs are inspired directly by the bag-of-words model (BOVW), a well-known and widely used method in text retrieval, where a document is represented by a set of distinct keywords. The same concept governs the BOVW model, in which an image is represented by a set of distinct visual words derived from local features. In [20] the authors conclude that histogram of visual words produces more stable results than the ones based on global image features. CCDs are described in Section 2, whereas BOVW and visual-word histograms are described in details in Section 3.

According to the proposed method, video is considered as a sequence of frames. Each frame is considered as a separate image and is described by CCDs and from a histogram of visual-words. Additionally, the whole video is described by an artificially generated image, which is generated dynamically from the video. Afterwards, the distance of the low level features of each frame with the low level features of the artificially generated image is calculated, in order to extract the video summary. These distances are inserted as input in a powerful Self-Growing and Self-Organized Neural Gas (SGONG) network[3]. The SGONG network has the ability to calculate the optimal number of output neurons and finally to classify each frame of the video in the appropriate cluster. More details about the SGONG are given in Section 4. The total of the clusters sets the video summary. The frame that corresponds to the center of each cluster is considered as the frame that is able to describe the cluster. A significant characteristic of the proposed method is its ability to calculate dynamically the appropriate number of clusters. Consequently, a video summary is generated. The entire procedure is given in details in Section 5 while the experimental results are shown in Section 6. Finally the conclusions are drawn in Section 7.

2 Compact Composite Descriptors

The family of Compact Composite Descriptors (CCDs) includes the following four descriptors:

1. the Color and Edge Directivity Descriptor (CEDD) [24]
2. the Fuzzy Color and Texture Histogram (FCTH) [24],
3. the Brightness and Texture Directionality Histogram (BTDH) descriptor [8] and
4. the Spatial Color Distribution Descriptor (SpCD) [9]

The Color and Edge Directivity Descriptor (CEDD) and the Fuzzy Color and Texture Histogram (FCTH) are used to describe natural color images. CEDD and FCTH use the same color information, since two fuzzy systems are applied to them, resulting in reducing the scale of the colors of the image to 24. These 2 descriptors demand a small size for indexing images. The CEDD length is 54 bytes per image while FCTH length is 72 bytes per image. The early fusion of CEDD and FCTH leads to a new descriptor, called Joint Composite Descriptor (JCD) [7].

The Brightness and Texture Directionality Histogram (BTDH) descriptor combines brightness and texture characteristics in order to describe grayscale images. A two unit fuzzy system is used to extract the BTDH descriptor; the first fuzzy unit classifies the brightness value of the images pixels into clusters in order to extract the brightness information using Gustafson Kessel [14] fuzzy classifier and the other one is used to extract texture information suggested by the Directionality histogram in [26].

The Spatial Color Distribution Descriptor (SpCD) is used for artificially generated images combining color and spatial color distribution information. This descriptor uses a fuzzy linking system that reduces the scale of the image to 8 colors. SpCD captures the spatial distribution of the color by dividing the image into sub-images not to mention the fact that its length does not exceed 48 bytes per image.

3 Bag of Visual Words

Content based image retrieval with global features is notoriously noisy for image queries of low generality, i.e. the fraction of relevant images in a collection [2]. On the other hand, local-feature approaches provide a slightly better retrieval effectiveness than global features [1] but are more expensive computationally [23].

In order to surpass the aforementioned difficulty the Bag-of-visual-words (BOVW) approach is adopted. When employing this approach, the extracted local features are clustered using k-means classifier and the computed cluster centers are called visual words. The set of visual words forms a visual vocabulary also known as codebook. For every new image added in the collection its local features must be extracted and assigned to the best fitting visual word from the existing codebook. By the end of that process a local feature histogram is composed for each image in the collection. Multiple approaches to enhance the bag of words approach have been proposed in literature [16].

In this paper, a visual word histogram with a universal vocabulary/codebooks is used. SURF [5] local features are extracted from all the 237434 images from the ImageCLEF 2010 Wikipedia test collection. Then, we randomly select 100000 descriptors to create the visual words that will form our codebook. SURF descriptors are clustered in 256 clusters and the mean vector (using k-means) is used as a visual word.

For every frame the SURF features are extracted. The local features are assigned to the best fitting visual word using the nearest neighbor method from the earlier created codebooks.

4 Self-Growing and Self-Organized Neural Gas Network

The Self-Growing and Self-Organized Neural Gas (SGONG) Network[3] is an unsupervised neural classifier. SGONG network combines the advantages both of the Kohonen Self-Organized Feature Map (SOFM) [17] and the Growing Neural Gas (GNG) [13] neural classifiers according to which, the learning rate and the radius of the neighboring domain of neurons is monotonically decreased during the training procedure. The SGONG network has been used in [3] in order to reduce the colors of an image. It has also been utilized by a new method for hand gesture recognition[25]. It has the ability to cluster the input data, so as the distance of the data items within the same class (intra-cluster variance) is small and the distance of the data items stemming from different classes (inter-cluster variance) is large. A significant characteristic of this classifier is that it adjusts the number of created neurons and their topology in an automatic way. To achieve this, at the end of each epoch of the SGONG classifier, three criteria are introduced. These criteria are able to improve the growing and the convergence of the network. A main advantage of the SGONG classifier is its ability to determine the final number of clusters.

The SGONG consists of two layers, the input and the output layer. It has the following main characteristics:

- Is faster than the Kohonen SOFM in its convergence.
- In contrast with GNG classifier, a local counter is defined for each neuron that influences the learning rate of this neuron and the strength of its connections. This local counter depends only on the number of the training vectors that are classified in this neuron.
- The dimensions of the input space and the output lattice of neurons are always identical.
- Criteria are used to ensure fast convergence of the neural network. Also, these criteria permit the detection of isolated classes.

The coordinates of the classes' centers are defined by the corresponding coordinates of the output neurons. Each output neuron is described by two local parameters. The first parameter is related to the training ratio and the second one refers to the influence by the nearby neurons. At the beginning of the training, the SGONG network consists of only two neurons . As the training procedure progresses, the network inserts new neurons in order to achieve better data clustering. Its growth is based on the following criteria:

- A neuron is inserted near the one with the greatest contribution to the total classification error, only if the average length of its connections with the neighboring neurons is relatively large.
- The connections of the neurons are created dynamically by using the Competitive Hebbian Learning method.

The main characteristic of the SGONG is that both neurons and their connections approximate effectively the the topology of the input data.

5 Implementation- Method Overview

A detailed description of the method is demonstrated in the following steps:

To begin with, the video is decomposed into its frames. Each frame corresponds to independent image. The first step of the proposed method includes the dynamic construction of an artificial image. In order to determine the value of each pixel of the artificially generated image, it is executed a uniform color quantization in the frames of the video with 216 unique colors. Thus, every pixel of the artificially generated image is the corresponding most frequent used pixel of all the color quantized frames. In other words, as artificially generated image is defined an image whose each pixel is described by the following equation:

$$F(R,G,B)_{x,y} = \sum_{F=1}^{N} p_F(R,G,B)_{x,y} \tag{1}$$

$$p(R,G,B)_{x,y} = p_{Max(F(R,G,B)_{x,y})}(R,G,B)_{x,y} \tag{2}$$

Where $F(R,G,B)_{x,y}$ is the number of pixels that can be found in the position x, y and their values is $p_F(R,G,B)_{x,y}$. The (R,G,B) value of the pixel of the artificial image in a position x, y equals to the value (R,G,B) of the pixels that have the higher $F(R,G,B)_{x,y}$.

In order to avoid out of memory problems and to make the algorithm more efficient and quicker, all the frames of the video are resized into a smaller size. This procedure is taking place using tiles for each frame, and not the entire frame. For the calculation of the tiles of each frame is used the bicubic method and the final size of each tile is set to be 128×128 pixels. This number is chosen as a compromise between the image detail and the computational demand.

In the next step for each frame of the video the Compact Composite Descriptors (CCDs) and the visual words histogram are calculated. Note that, the descriptors are calculated from the original frames and not from the color quantized and resized ones. The CCDs descriptors that are extracted are the Joint Composite Descriptor (JCD), the Brightness and Texture Directionality Histogram (BTDH) descriptor and the Spatial Color Distribution Descriptor (SpCD).

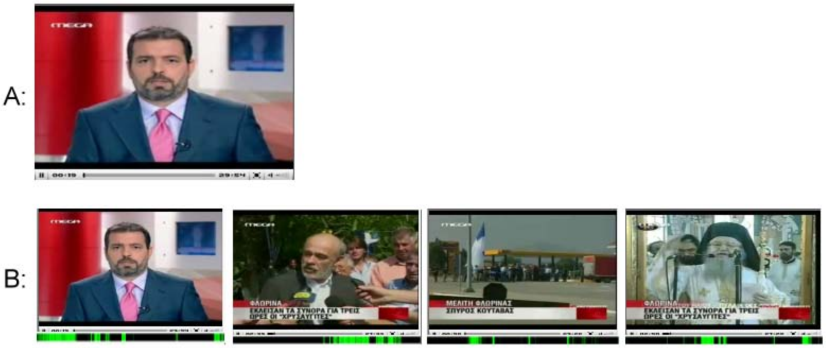


Fig. 1. (A) Original Video, (B) Key Frames and Timeline

Table 1. Evaluation Videos

Title	Type	URL
Waka-Waka	Video Clip	http://www.youtube.com/watch?v=pRpeEdMmmQ0
Al Tsantiri News	Tv Show	http://www.youtube.com/watch?v=KjbA3L6kQa8
Mickey Mouse	Animation	http://www.youtube.com/watch?v=jOvFIoBoxag
Gummy Bear	Animation	http://www.youtube.com/watch?v=astISOttCQ0
Radio Arvila	Tv Show	http://www.youtube.com/watch?v=UHbjy9k53cU

As it has already mentioned, the problem of video summarization is expanded to a single query image retrieval problem. The artificial image is used as the query image in order to retrieve and sort the frames of the video to ranking lists. This sorting is accomplished by calculating the distance between the descriptors of the artificial image and the descriptors of each frame. The distance is calculated by using the Tanimoto coefficient:

$$D(i,j) = T_{ij} = t(x_i, x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j} \tag{3}$$

where x^T is the transpose vector of the descriptor x.

In the absolute congruence of the vectors, the Tanimoto coefficient takes the value 1, while in the maximum deviation the coefficient tends to zero.

The procedure is repeating for every descriptor (JCD, BTDH, SpCD, visual words histogram) and in the end four ranking lists are constructed.

The next step includes the classification of the frames, which is implemented by the Self-Growing and Self-Organized Neural Gas (SGONG) network. The SGONG is fed by the distances of all frames from the artificially generated image. At this point it is worth mentioning that it is required the setting of some important parameters. The setting of these parameters are significant for the correct operation of SGONG network and regard the adding and the removing of the neurons. Moreover, another parameter that should be considered is the maximum number of neurons. This number should be

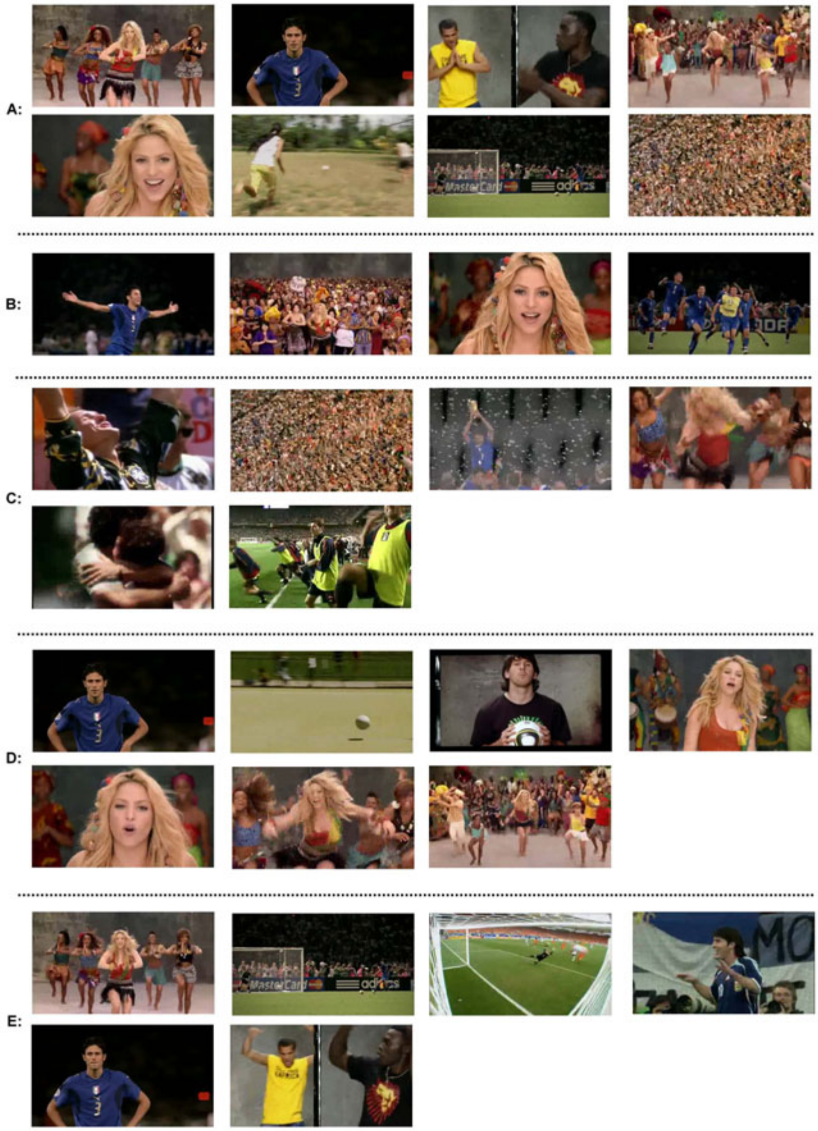


Fig. 2. Key Frames Extracted Per Method. (A) Proposed Method Produced 8 Key Frames, (B) BTDH Descriptor Produced 4 Key Frames, (C) JCD Descriptor Produced 6 Key Frames, (D) SpCD Descriptor Produced 7 Key Frames and (E) Visual Words Approach Produced 6 Key Frames.

chosen appropriately in order to the successful convergence of the SGONG network into a number of neurons less than this threshold. Also, the right choice of the learning rate of the network is an absolute necessity.

After training, the weights of the output neurons define the centers of the clusters. Each cluster corresponds to a "scene". The total of the "scenes" describes the whole video. For each cluster there is a representative key frame, which describes the cluster. This key frame is the nearest one of all the corresponding frames to the center of the cluster as it results from the SGONG classifier. Thus, for each cluster a key frame is extracted. These key frames are considered as the most significant frames of the cluster. A significant characteristic of the proposed method is its ability to calculate dynamically the appropriate number of clusters, which is based in the main advantage of the SGONG to adjust the number of created neurons and their topology in an automatic way.

In order to be illustrated the participation of every frame in every scene/cluster visually, is used a timeline. For every key frame, which has been calculated according to the proposed method, there is a timeline. The green color (see Fig.1) corresponds to the parts of the video that participate in this scene.

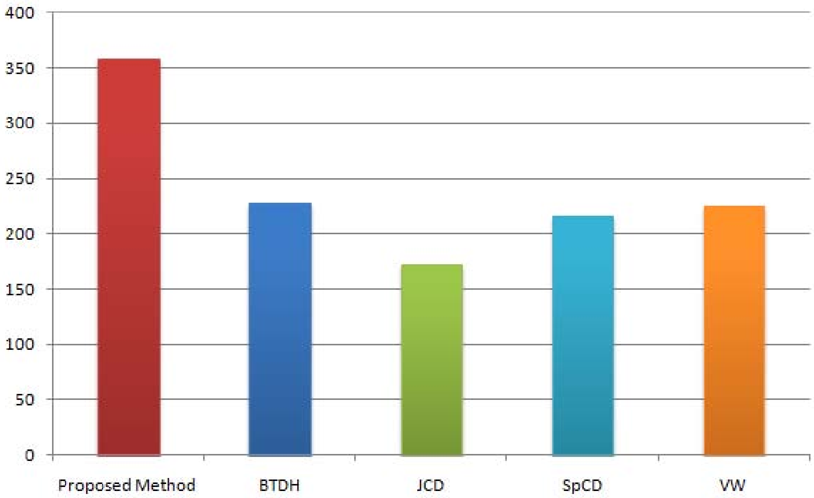


Fig. 3. User ratings

Table 2. Number of Key Frames Per Method

Title	Length	Proposed Method	BTDH	JCD	SpCD	Visual Words
Waka-Waka	211 s	8	4	6	7	6
Lazopoulos	196 s	8	5	6	4	6
Mickey Mouse	84 s	11	5	6	3	4
Gummy Bear	164 s	8	5	5	5	5
Radio Arvila	209 s	4	5	3	4	6

6 Experimental Results

In order to indicate the effectiveness of the proposed method, a user study was held. The proposed method that utilizes the combination of four descriptors was compared with the single utilization of each descriptor. So, users had to choose their favourite summary between five different summaries for each video. In this study five videos are analysed. Each participating user had to mark the five summaries for each video with a degree(5 points for the best down to 1 point for the worst). Sixteen users were participated in the study. Figure 2 shows the five video summaries for the Waka-Waka video extracted from the proposed method and from the four pre-mentioned techniques.

According to the results of the study illustrated in Figure 3, the proposed method reached the highest rating comparing with the other four techniques. More particularly, the score of the proposed method was 358 points with a clear difference from the other four approaches. The method that utilizes the BTDH descriptor came second with 228 points, while the method based on the visual word histogram and the method that utilizes the SpCD descriptor followed with 226 and 216 points respectively. The method that uses the JCD descriptor was found in the last place with 172 points.

The number of key frames extracted by all methods for each video is depicted in Table 2. It can easily be understood, the proposed method generated much more key frames than the other methods. A striking example is the generation of eleven key frames by the proposed approach in the Micky Mouse video, while the average number of the extracted key frames of the other methods is 4.5.

7 Conclusions

In this paper, a novel approach to summarize a video, based on a new Self-Growing and Self-Organized Neural Gas network is proposed. The proposed method utilizes the combination of four descriptors in order to describe the frames of the video. Our approach appears to have quite good results according to the user study held for the purposes of this paper. The method seems to be the best out of the other four techniques for each one of which only one descriptor was utilized. In addition, it has the advantage to determine the optimal number of the extracted key frames of a video.

References

1. Aly, M., Welinder, P., Munich, M.E., Perona, P.: Automatic discovery of image families: Global vs. local features. In: ICIP, pp. 777–780 (2009)
2. Arampatzis, A., Zagoris, K., Chatzichristofis, S.A.: Dynamic two-stage image retrieval from large multimodal databases. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 326–337. Springer, Heidelberg (2011)
3. Atsalakis, A., Papamarkos, N.: Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas. Eng. Appl. of AI 19(7), 769–786 (2006)
4. Bailer, W., Dumont, E., Essid, S., Merialdo, B.: A collaborative approach to automatic rushes video summarization. In: 15th IEEE International Conference on Image Processing, 2008. ICIP 2008, pp. 29–32 (2008)

5. Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)
6. Borth, D., Ulges, A., Schulze, C., Breuel, T.M.: Keyframe extraction for video tagging and summarization. In: Proc. Informatiktage, pp. 45–48 (2008)
7. Chatzichristofis, S.A., Arampatzis, A., Boutalis, Y.S.: Investigating the behavior of compact composite descriptors in early fusion, late fusion, and distributed image retrieval. Radioengineering 19 (4), 725–733 (2010)
8. Chatzichristofis, S.A., Boutalis, Y.S.: Content based radiology image retrieval using a fuzzy rule based scalable composite descriptor. Multimedia Tools Appl. 46(2-3), 493–519 (2010)
9. Chatzichristofis, S.A., Boutalis, Y.S., Lux, M.: Spcd - spatial color distribution descriptor - a fuzzy rule based compact composite descriptor appropriate for hand drawn color sketches retrieval. In: Filipe, J., Fred, A.L.N., Sharp, B. (eds.) ICAART (1), pp. 58–63. INSTICC Press (2010)
10. Ciocca, G., Schettini, R.: An innovative algorithm for key frame extraction in video summarization. Journal of Real-Time Image Processing 1(1), 69–88 (2006)
11. Cula, O.G., Dana, K.J.: Compact representation of bidirectional texture functions. In: CVPR (1), pp. 1041–1047 (2001)
12. Dumont, E., Merialdo, B.: Sequence alignment for redundancy removal in video rushes summarization. In: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop, pp. 55–59. ACM, New York (2008)
13. Fritzke, B.: Growing grid - a self-organizing network with constant neighborhood range and adaptation strength. Neural Processing Letters 2(5), 9–13 (1995)
14. Gustafson, D.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes, vol. 17 (1978)
15. Hanjalic, A., Zhang, H.J.: An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on Circuits and Systems for Video Technology 9(8), 1280–1289 (1999)
16. Kogler, M., Lux, M.: Bag of visual words revisited: an exploratory study on robust image retrieval exploiting fuzzy codebooks. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD 2010, pp. 3:1–3:6. ACM, New York (2010)
17. Kohonen, T.: The self-organizing map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
18. Lie, W.N., Hsu, K.C.: Video summarization based on semantic feature analysis and user preference. In: 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 486–491. IEEE, Los Alamitos (2008)
19. Lux, M., Schoffmann, K., Marques, O., Boszormenyi, L.: A novel tool for quick video summarization using keyframe extraction techniques. In: Proceedings of the 9th Workshop on Multimedia Metadata (WMM 2009). CEUR Workshop Proceedings, vol. 441, pp. 19–20 (2009)
20. Kogler, M., del Fabro, M., Lux, M., Schoffmann, K., Boszormenyi, L.: Global vs. local feature in video summarization: Experimental results. In: SeMuDaTe 2009, 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies, SeMuDaTe 2009 (2009)
21. Matos, N., Pereira, F.: Using mpeg-7 for generic audiovisual content automatic summarization. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2008, pp. 41–45 (2008)
22. Money, A.G., Agius, H.: Video summarisation: A conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation 19(2), 121–143 (2008)

23. Popescu, A., Moellic, P.A., Kanellos, I., Landais, R.: Lightweight web image reranking. In: Proceedings of the Seventeen ACM International Conference on Multimedia, pp. 657–660. ACM, New York (2009)
24. Chatzichristofis, S.A., Zagoris, K., Boutalis, Y.S., Papamarkos, N.: Accurate image retrieval based on compact composite descriptors and relevance feedback information. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 2, 207–244 (2010)
25. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Eng. Appl. of AI 22(8), 1141–1158 (2009)
26. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. IEEE Transactions on Systems, Man and Cybernetics 8(6), 460–473 (1978)
27. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 3(1), 1551–6857 (2007)
28. Xu, M., Maddage, N.C., Xu, C., Kankanhalli, M., Tian, Q.: Creating audio keywords for event detection in soccer video. In: Proc. of ICME, vol. 2, pp. 281–284 (2003)
29. Zhang, D., Chang, S.F.: Event detection in baseball video using superimposed caption recognition. In: Proceedings of the tenth ACM international conference on Multimedia, pp. 315–318. ACM, New York (2002)

Real-Time Upper-Body Human Pose Estimation Using a Depth Camera

Himanshu Prakash Jain[1], Anbumani Subramanian[2],
Sukhendu Das[1], and Anurag Mittal[1]

[1] Indian Institute of Technology Madras, India
[2] HP Labs India, Bangalore

Abstract. Automatic detection and pose estimation of humans is an important task in Human-Computer Interaction (HCI), user interaction and event analysis. This paper presents a model based approach for detecting and estimating human pose by fusing depth and RGB color data from monocular view. The proposed system uses Haar cascade based detection and template matching to perform tracking of the most reliably detectable parts namely, head and torso. A stick figure model is used to represent the detected body parts. The fitting is then performed independently for each limb, using the weighted distance transform map. The fact that each limb is fitted independently speeds-up the fitting process and makes it robust, avoiding the combinatorial complexity problems that are common with these types of methods. The output is a stick figure model consistent with the pose of the person in the given input image. The algorithm works in real-time and is fully automatic and can detect multiple non-intersecting people.

1 Introduction

Motion capture for humans is an active research topic in the areas of computer vision and multimedia. It has many applications ranging from computer animation and virtual reality to human motion analysis and human-computer interaction (HCI) [2] [18]. The skeleton fitting process may be performed automatically or manually, as well as intrusively or non-intrusively. Intrusive manners include, for example, imposing optical markers on the subject [11] while non-automatic method could involve manual interaction to set the joints on the image, such as in [4]. These methods are usually expensive, obtrusive, and not suitable for surveillance or HCI purposes. Recently, due to the advances on imaging hardware and computer vision algorithms, markerless motion capture using a camera system has attracted the attention of many researchers. One of the commercial solutions for markerless motion capture includes Microsoft's Kinect system [17] for console systems. Kolb et al. [14] gives an account of recent developments in Time-of-Flight (ToF) technology and discusses the current state of the integration of this technology into various vision and graphics-related applications.

Since the application domain is less restrictive with only a monocular view, human pose estimation from monocular image captures has become an emerging

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 227–238, 2011.

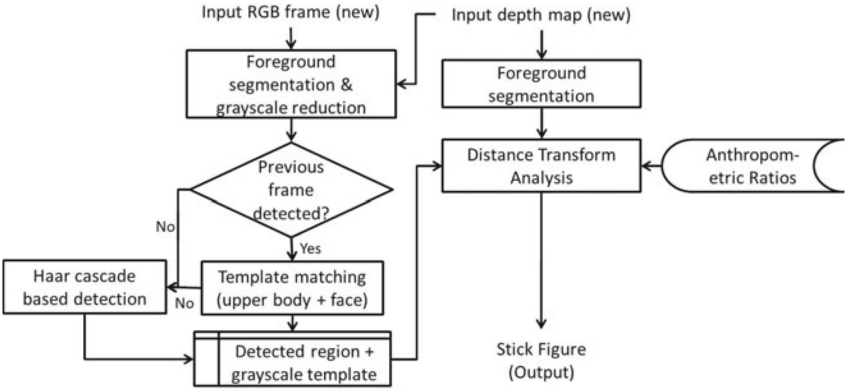

Input RGB frame (new)
Input depth map (new)
Foreground segmentation & grayscale reduction
Foreground segmentation
Previous frame detected?
Distance Transform Analysis
Anthropom-etric Ratios
No
Yes
Haar cascade based detection
No
Template matching (upper body + face)
Detected region + grayscale template
Stick Figure (Output)

Fig. 1. Flowchart of the proposed system for upper-body human pose estimation

issue to be properly addressed. Haritaoglu et al. [10] tries to find the pose of a human subject in an automatic and non-intrusive manner. It uses geometrical features to divide the blob and determine the different extremities (head, hands and feet). Similarly, Fujiyoshi and Lipton [8] have no model but rather determine the extremities of the blob with respect to the centroid and assume that these points represent the head, hands and feet. Guo et al. [9] attempts to find the exact positions of all body joints (like the neck, shoulder, elbow, etc.) by minimizing the distance based criterion function on the skeletonized foreground object to fit the stick model. Neural networks [19] and genetic algorithms [22] have also been used to obtain the complete position of all the joints of the person. Jensen et al. [12] tries to estimate the pose based on an articulated model, for gait analysis using calibrated ToF camera.

The simplest representation of a human body is the stick figure, which consists of line segments linked by joints. The motion of joints provides the key to motion estimation and recognition of the whole figure. This concept was initially considered by Johansson [13], who marked joints as moving light displays (MLD). Along this vein, Rashid [20] attempted to recover a connected human structure with projected MLD by assuming that points belonging to the same object have higher correlations in projected positions and velocities.

The organization of the paper is as follows: Section 2 discusses the proposed approach with subsections giving details about each module used in the system. Section 3 extends the discussion towards the implementation details about the proposed prototype. Finally, Section 4 concludes the paper and delineates possible directions for future research.

2 Overview of the Entire System

In this work, we assume that a depth-camera is static and is positioned at human height. It is also assumed that users' interaction spaces are non-intersecting and

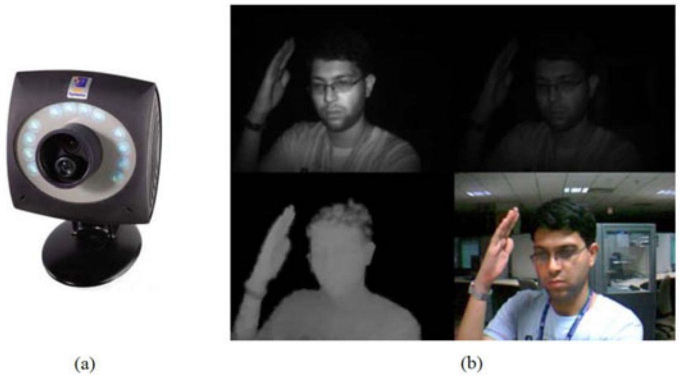


Fig. 2. (a) ZCam from 3DV Systems; (b) Data output from ZCam - top row: primary and secondary infrared images, bottom row: a depthmap and a RGB color image

upper-body and face are visible without any occlusion. A block diagram of the human detection and pose estimation approach used in our work is shown in Fig. 1. The following subsections provide details of each module incorporated in the system.

2.1 Depth Camera

We use ZCam from 3DV Systems [1] (shown in Fig. 2(a)) in our work done in mid-2010. The technology of this device is similar to Kinect systems currently available in the market. This camera uses active illumination for depth sensing - it emits modulated infra-red (IR) light and based on the time-of-flight principle, the reflected light is used to calculate depth (distance from camera) in a scene. This camera provides both RGB (640 x 480 resolution, VGA size) image and a grayscale depthmap (320 x 240 resolution, QVGA size) image at 30 frames per second (fps). Figure 2(b) shows a sample of four images obtained from the camera. The top row shows active brightness (left) and passive brightness (right) IR images and the bottom row shows the depthmap (left) and the RGB (right) image respectively. It can be observed in the depthmap, that the depth values of objects near the camera appear bright while those of objects that are farther appear darker.

2.2 Foreground Segmentation

We use the RGB image and the depthmap as inputs to the system (see Fig. 3). A threshold is used to remove noise (with low values) from the raw depth map information, obtained from ZCam without any calibration. These foreground pixels are then segmented into regions by a linear-time component labeling algorithm [6]. The extracted connected components or blobs, obtained from the depth map using 8-connectivity of pixels, are thresholded based on area. The

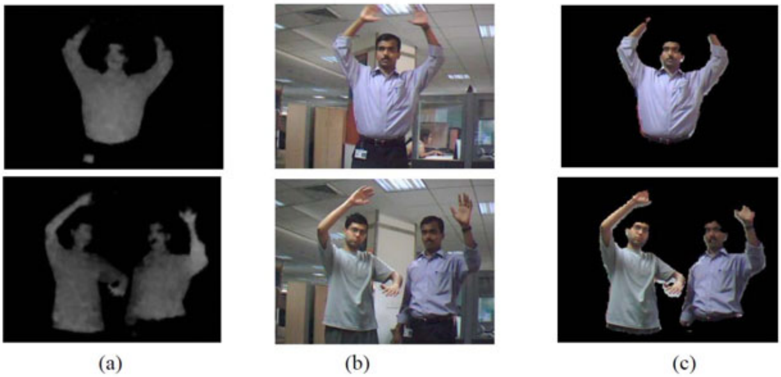


Fig. 3. (a) Input depthmaps; (b) Input RGB images; (c) Foreground segmented RGB images obtained using (a) and (b)

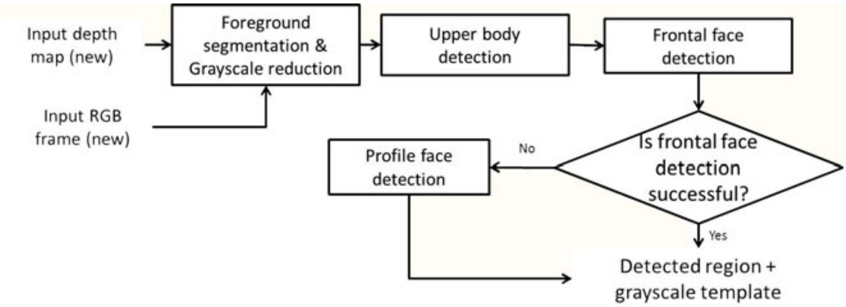

Input depth map (new)
Input RGB frame (new)
Foreground segmentation & Grayscale reduction
Upper body detection
Frontal face detection
Is frontal face detection successful?
No
Yes
Profile face detection
Detected region + grayscale template

Fig. 4. Haar cascade based detection logic

above blob analysis helps in pruning out small blobs and background noises in the input image. The processed depthmap is then used as a binary mask to obtain the foreground object in the RGB image.

2.3 Haar Cascade Based Detection

The object detector [16] based on Haar classifiers is used for detecting humans in the foreground segmented RGB images. Grayscale based object detector is used instead of an RGB based object detector, since it reduces the time complexity of the system by operating on a single channel. Human detector helps in differentiating humans from non-human objects present in the segmented foreground grayscale image (non-trivial using depth mask detection). For upper body detection, the classifier trained for upper-body (head + torso) [15] is used. The detected regions are then passed on to frontal face detector classifier (see Fig. 4). In case, the frontal face detection fails, a profile face detector [5] is used to detect faces. If either upper body detector or the profile face detector fails to produce any positive results then the frame is completely rejected and the next

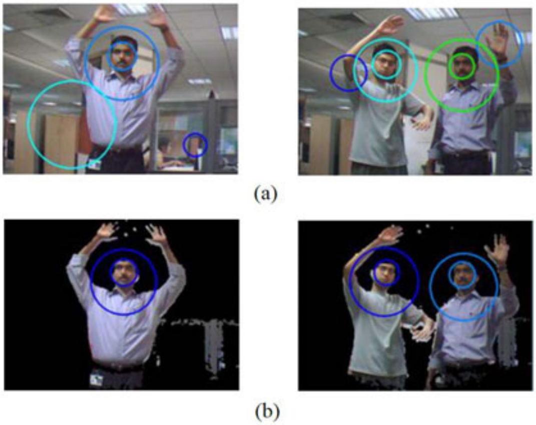


Fig. 5. Haar cascade based detection for upper-body and face. Circumscribed circles denote successful face (inner circle) and upper body detection (outer circle), whereas a single circle denotes a successful upper-body (either false positive or true positive) detection along-with unsuccessful face detection (either false negative or true negative). (a) Haar cascade based detection on original grayscaled RGB images. (b) Haar cascade detection on foreground segmented grayscaled RGB images.

frame is analyzed for any possible upper-body detection. If no face is detected in the identified upper body region, then it is assumed to be a false positive and the detection is rejected for further analysis. This successive detection logic helps in reliably determining the positive detections and pruning out the false positive detections. In order to reduce the computation time as well as the false positives, Haar detection is done on the foreground segmented image (see Fig. 5).

2.4 Template Matching Based Tracking

The template-based approach determines the best location by matching an actual image patch against an input image, by "sliding" the patch over the input search image using normalized cross-correlation, defined as:

$$R(x,y) = \frac{\sum_{x',y'}(T_{RGB}^{G'}(x',y') \cdot I_{RGB}^{G'}(x+x',y+y'))}{\sqrt{\sum_{x',y'} T_{RGB}^{G'}(x',y')^2 \cdot \sum_{x',y'} I_{RGB}^{G'}(x+x',y+y')^2}} \tag{1}$$

$$\text{where,} \qquad T_{RGB}^{G'}(x,y) = T_{RGB}^{G}(x,y) - \overline{T_{RGB}^{G}}$$

$$I_{RGB}^{G'}(x,y) = I_{RGB}^{G}(x,y) - \overline{I_{RGB}^{G}}$$

T_{RGB}^{G} is the grayscaled RGB template image and I_{RGB}^{G} is the grayscaled RGB input image. Since template-based matching requires sampling of a large number of points, we can reduce the number of sampling points by reducing the

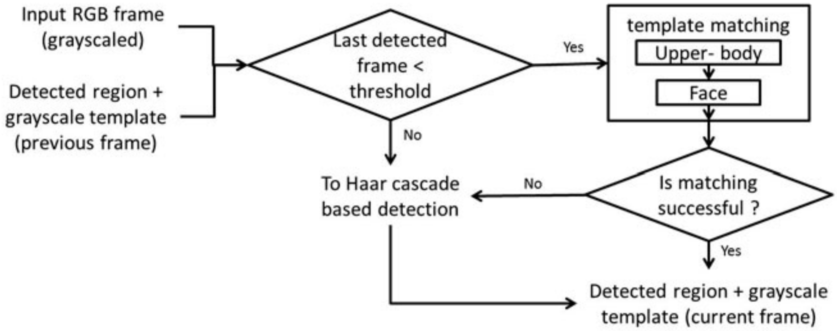

Input RGB frame (grayscaled)
Detected region + grayscale template (previous frame)
Last detected frame < threshold
Yes
No
template matching
Upper- body
Face
To Haar cascade based detection
No
Is matching successful ?
Yes
Detected region + grayscale template (current frame)

Fig. 6. Template Matching based tracking logic

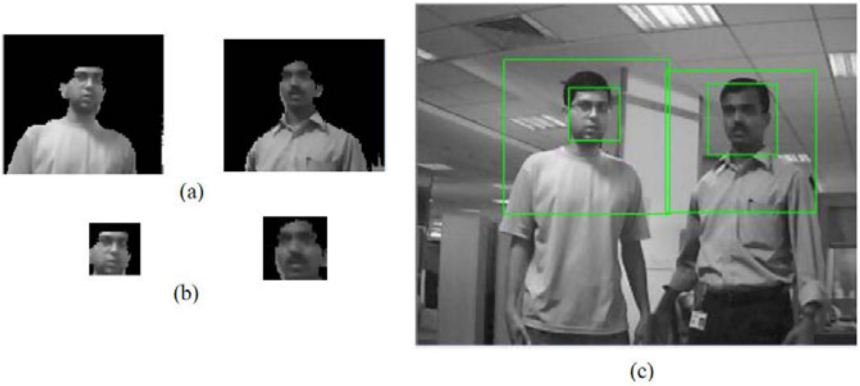


Fig. 7. Results for template matching based tracking. Templates are grayscaled and down-sampled to QVGA to reduce computation time. Similarly, input RGB image is also grayscaled and down-sampled to QVGA: (a) upper-body template identified in previous frame; (b) face templates identified in previous frames; (c) input image grayscaled and down-sampled with marked rectangular regions denoting successful template based tracking.

resolution of the search and template images by the same factor (in our case, down-sampled by a factor of 2) and performing the operation on the resultant downsized images. The template images/patches are obtained from the successful detection in the previous frame; either by Haar cascade based detection or by template based matching (see Fig. 6). Advantages of using template matching, over Haar cascades, is reduced computation time and higher true positives, since a Haar cascade misses variations in object orientation and pose. Template matching is successful in handling large pose variances of the object, if the inter-frame variance is low, since consecutive frames are used for matching. The system may fail for humans not facing (non-frontal pose) the camera. Haar cascade based detection is used only when there are no templates to perform matching or when

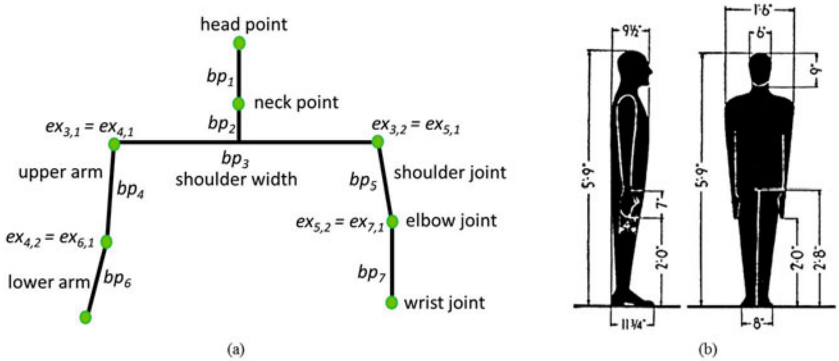


Fig. 8. (a) The stick model used for human upper-body skeleton fitting; (b) Anthropometric ratios of typical human body [3]

the template matching fails to track the template in the input image. Haar cascade based detection is forced after certain empirically chosen time-lapse/frames, to handle drifting errors and appearance of new person into the scene. Figure 7 shows examples of the template matching on the input images.

2.5 Stick Human Body Model

The skeleton model is represented by a vector of 7 body parts (bp_1 to bp_7) as shown in Fig. 8(a). The proportions between the different parts are fixed and were determined based on NASA Anthropometric Source Book [7] and [3](see Fig. 8). Each body part has its own range of possible motion. Each body part (bp_i) is composed of two extremities ($ex_{i,1}, ex_{i,2}$), representing the coordinates of the body part in the image plane:

$$bp_i = \{ex_{i,1}, ex_{i,2}\} \tag{2}$$

where, $ex_{i,j} = (x_{i,j}, y_{i,j})$. $x_{i,j}$ is the x coordinate of extremity j of the body part i and $y_{i,j}$ is the coordinate of the extremity j of the body part i.

The head, neck and shoulder (both left and right) joints are estimated based on detected upper-body and head region. The centroid of the detected head template is taken as head point. The shoulder joints are taken as the lower extremities of the detected upper body region in the input image. Based on the anthropometric ratios, the neck point is estimated to be at 2/3 of the vertical distance from head to shoulder points. Similarly, length of upper arms is taken as 2/3 of shoulder width and 5/9 of shoulder width in case of lower arms. This helps to detect head, neck and shoulder points of the detected humans from the foreground segments of the input grayscale image.

2.6 Limbs Fitting

In order to estimate the remaining joints (elbow and wrist, both left and right) and limb inclinations (upper and lower arm, both left and right), linear regression

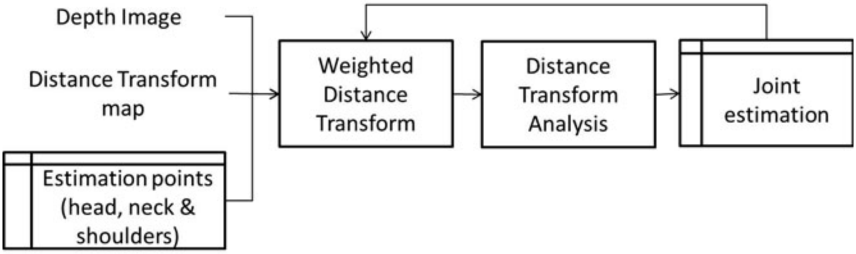

Depth Image
Distance Transform map
Estimation points (head, neck & shoulders)
Weighted Distance Transform
Distance Transform Analysis
Joint estimation

Fig. 9. Flowchat of limbs fitting method, based on linear regression of sampled weighted distance transform map

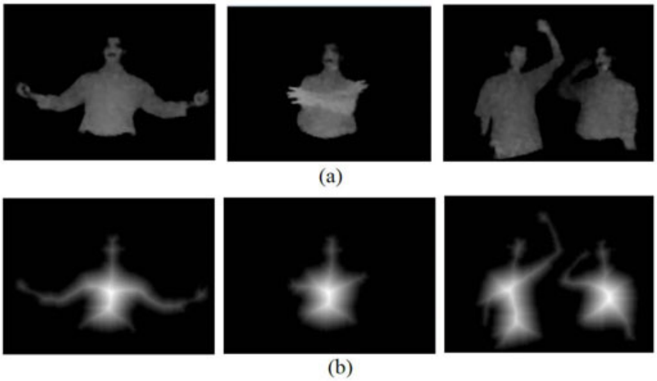


Fig. 10. DT on foreground segmented depthmap normalized from 0 to 255 range for visualization: (a) foreground segmented depthmap; (b) distance transform map

on sampled weighted-distance transform map (distance transform analysis) is performed (see Fig. 9). Once the elbow joints are estimated (as discussed in Sec. 2.5), weighted-distance transform w.r.t. these joints are computed for estimating wrist joints and 2D inclinations for lower arms. The Distance Transform (DT) maps each image pixel into its smallest distance to regions of interest [21]. Figure 10 shows some examples of DT on input images. Limb movements for human body can be out of the image plane, which DT fails to capture in the depthmap. In order to take into account the projected lengths of the limbs weighted-distance transform is calculated. The distance map of the image is multiplied with variance factor representing the variance ratio of the point w.r.t. the reference point (parent joint) in the direction orthogonal to the image plane. This variance can easily be calculated from the input depthmap. The weighted-distance transform $D^w(p, c)$ for point p w.r.t. c in depth image (I_d) is defined as:

$$D^w(p, c) = D(p) \cdot (1 + \frac{|I_d(p) - I_d(c)|}{I_d(c)}) \qquad \forall \quad I_d(c) \neq 0 \tag{3}$$

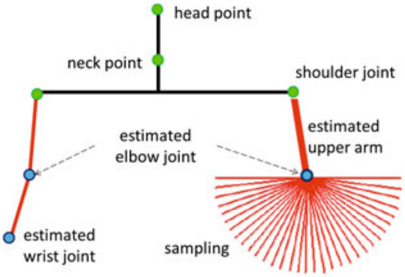


Fig. 11. Sampling of weighted distance transform map for left lower arm pose estimation. The green color points have already been estimated based on upper body and head region detection. The blue colored joints are estimated by sampling followed by linear regression.

where, $D(p)$ is DT value at point p in the input depth map image I_d. c is the reference point (parent joint) for estimating the angles for upper and lower arms. e.g. for estimating the inclination of upper left arm, reference point (c) is left shoulder joint and similarly for estimating the lower right arm, reference point (c) is right elbow joint. Sampling of the Weighted-Distance Transform map is done upto length l from the reference point (parent joint) c in an angular region varying from 0 to 2π, and with a predefined sampling angle. Temporal information can be incorporated to improve computational efficiency by imposing range constraints on the angular region for sampling the map (see Fig. 11). The length l of arms is estimated based on anthropometric ratios as discussed in Sec. 2.5. The step size for sampling (search for optimal value in 1-D) the orientation angle influences the robustness and speed of the technique. If it's too large, a good solution could be overlooked. However, the whole process might take too long if the step size is chosen small. It then becomes possible to sample points along and for each candidate solution. In estimation of both upper arms and lower arms, a second global maximum is taken as the estimated pose of the limb. In case of upper arms, the global maxima always denotes the angle from left or right shoulder joint towards torso's center region; since weighted-distance transform map value is always maxima along this path (see Fig. 10). Similarly for lower arms, a global maximum denotes the angle connecting the elbow joints to shoulder joints, as the physical structure of human body, upper arms are broader in width compared to lower arms. Due to these reasons second maxima is universally chosen to represent the estimated limb's inclination.

The sampling rate is an adjustable parameter that also influences the robustness and speed of the method. Indeed, the more points there are along a line to validate a solution, the more robust the system is if a part of a limb has been poorly extracted. However, the fitting process becomes more time consuming. A local method such as the one presented above also increases the robustness of the whole system in the following way. If some region of the blob has been poorly extracted, it is likely that only this part will be poorly fitted, while the

Table 1. Computational time for various modules in our system

Modules	**Time/frame (in ms)**
Haar cascade based upper-body & face detection	$\sim 57ms/frame$
Skeleton fitting	$\sim 11ms/frame$
Total time using detection	**∼ 68 ms/frame**
Template matching based tracking	$\sim 3ms/frame$
Skeleton fitting	$\sim 5ms/frame$
Total time using tracking	**∼ 8 ms/frame**
Average Running Time (Threshold = 15 frames/sec)	**∼ 14 ms/frame**

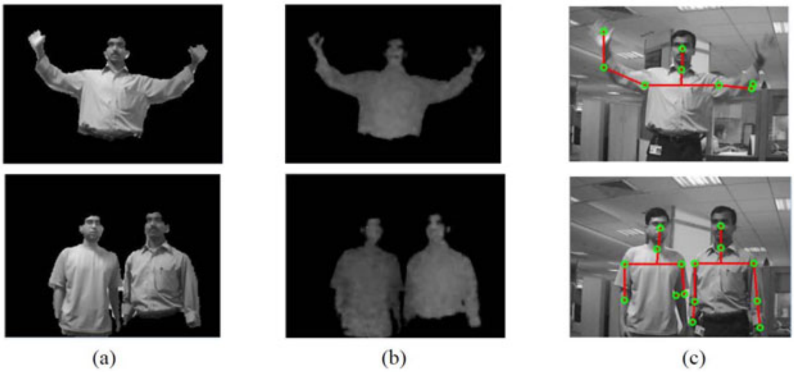


Fig. 12. (a) Foreground segmented grayscaled RGB image; (b) Input depthmap; (c) Estimated upper body human stick figure overlaid upon the grayscaled RGB image

other limbs will be successfully fitted if the upper body detection is successful. In the case of a global method, a small error can lead to the failure of the whole fitting module. However, because of the local fitting method, even if one part is missed, the overall fitting is often acceptable. The fitting process for the right arm is independent from that of the left arm, therefore, the error in the estimation process of the former will not affect the later, and vice-versa. This makes our proposed local approach more robust.

3 Experimental Results

We have developed a working prototype of our human detection and pose estimation logic. The prototype was implemented using C/C++ and OpenCV library, on a windows platform. The prototype works in real-time using live feeds from 3DV camera mounted on top of a personal computer. We have tested the above prototype for single as well as multiple (upto 3) non-intersecting people with appearance and disappearance of people at random and for various different upper body poses. The input RGB stream is of 640 x 480 resolution (VGA) at 30 fps and the depth stream is of 320 x 240 resolution (QVGA) at 30 fps. For foreground segmentation, blob with size less than 400 pixels (empirically chosen) are considered as non-humans. Haar cascade based detection is done on VGA size grayscaled RGB image to increase true positive detections. Template matching based tracking is done on a QVGA size grayscaled RGB image to reduce computation time. Threshold used for enforcing Haar cascade based detection is empirically chosen as 15 frames. Since foreground segmentation is the most critical step in pose estimation, poor foreground segmentation can sometimes lead to incorrect pose estimation. Figure 12 shows a few examples of our analysis done on input frames of humans interacting in various poses. Table 1 gives the time taken (on a machine with Intel Core 2 Extreme processor, 3 GHz and 3 GB RAM) for various processes in the prototype. The average running time of the entire process is less than the total time used for detection (∼68 ms/frame) since Haar cascade based detection is enforced only once in every 15 frames while for the rest of the frames, template matching based tracking (∼8ms/frame) is used. A rigorous performance analysis for measuring the scalability and robustness of our approach can be a possible scope of future work.

4 Conclusions

In this paper, we have presented a viable vision-based human pose estimation technique using RGB and depth streams from a monocular view. An articulated graphical human model is created for pose estimation of upper-body parts for HCI applications. Our technique uses a balance of Haar cascade based detection and template matching based tracking. Haar based detection handles appearance of humans and drifting errors in tracking, while template matching based tracking is able to handle variations in object pose and makes the approach computationally light. Limbs fitting is performed progressively, one limb at a time, instead of globally. This way, the process is faster and robust. We have demonstrated the technique for various real-world input data. Some improvements are possible in this framework. Incorporating skin detection and edge detection would reduce false positive configurations for lower arms. Occlusion handling and comparative studies with published work form nice scope of work in the future.

References

1. Zcam from 3dv systems (2009), http://3dvzcam.com
2. Aggarwal, J., Cai, Q.: Human motion analysis: A review. In: Proceedings of the Nonrigid and Articulated Motion Workshop, pp. 90–102 (1997)
3. Badler, N.I., Phillips, C.B., Webber, B.L.: Simulating Humans: Computer Graphics, Animation, and Control. Oxford University Press, Oxford (1993)
4. Barrón, C., Kakadiaris, I.A.: Estimating anthropometry & pose from a single uncalibrated image. Computer Vision and Image Understanding 81, 269–284 (2001)
5. Bradley, D.: Profile face detection (2003), http://opencv.willowgarage.com

6. Chang, F., jen Chen, C., jen Lu, C.: A linear-time component-labeling algorithm using contour tracing technique. Computer Vision and Image Understanding 93, 206–220 (2004)
7. Churchill, E., McConville, J.T., Laubach, L.L., Erskine, P., Downing, K., Churchill, T.: Anthropometric source book. A handbook of anthropometric data, vol. 2. NASA (1978)
8. Fujiyoshi, H., Lipton, A.J.: Real-time human motion analysis by image skeletonization. In: Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision (WACV 1998), pp. 15–21 (1998)
9. Guo, Y., Xu, G., Tsuji, S.: Tracking human body motion based on a stick figure model. Journal of Visual Comm. and Image Representation 5(1), 1–9 (1994)
10. Haritaoglu, I., Harwood, D., Davis, L.: W4: Who? when? where? what? A real time system for detecting and tracking people. In: Proceedings of the Third IEEE Int. Conf. on Automatic Face and Gesture Recog., pp. 222–227 (1998)
11. Herda, L., Fua, P., Plänkers, R., Boulic, R., Thalmann, D.: Skeleton-based motion capture for robust reconstruction of human motion. In: Proceedings of the Computer Animation, pp. 77–83. IEEE Computer Society, Los Alamitos (2000)
12. Jensen, R.R., Paulsen, R.R., Larsen, R.: Analyzing gait using a time-of-flight camera. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 21–30. Springer, Heidelberg (2009)
13. Johansson, G.: Visual motion perception. Scientific American 232(6), 76–89 (1975)
14. Kolb, A., Barth, E., Koch, R., Larsen, R.: Time-of-flight cameras in computer graphics. Computer Graphics Forum 29, 141–159 (2010)
15. Kruppa, H., Santana, M.C., Schiele, B.: Fast and robust face finding via local context. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (October 2003)
16. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: Proceedings of the International Conference on Image Processing, vol. 1, pp. 900–903 (2002)
17. Microsoft: Kinect for xbox 360 (2010), http://www.xbox.com/en-US/kinect
18. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104(2-3), 90–126 (2006)
19. Ohya, J., Kishino, F.: Human posture estimation from multiple images using genetic algorithm. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Image Processing. vol. 1, pp. 750–753 (1994)
20. Rashid, R.F.: Towards a system for the interpretation of moving light display. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(6), 574–581 (1980)
21. Rosenfeld, A., Pfaltz, J.: Distance function on digital pictures. Pattern Recognition 1(1), 33–61 (1968)
22. Takahashi, K., Uemura, T., Ohya, J.: Neural-network-based real-time human body posture estimation. In: Proceedings of the IEEE Signal Processing Society Workshop Neural Networks for Signal Processing X, vol. 2, pp. 477–486 (2000)

Image Matting with Transductive Inference

Jue Wang

Adobe Systems, Seattle, WA 98103, USA
juewang@adobe.com

Abstract. Various matting methods have been proposed to isolate objects from images by extracting alpha mattes. Although they typically work well for images with smooth regions, their ability to deal with complex or textured patterns is limited due to their inductive inference nature. In this paper we present a *Transductive Matting* algorithm which explicitly treats the matting task as a statistical transductive inference. Unlike previous approaches, we assume the user marked pixels do not fully capture the statistical distributions of foreground and background colors in the unknown region of the given trimap, thus new foreground and background colors are allowed to be recognized in the transductive labeling process. Quantitative comparisons show that our method achieves better results than previous methods on textured images.

1 Introduction

Using image matting techniques for creating novel composites or facilitating other editing tasks has gained considerable interests from both professionals and consumers. In the matting problem, an observed image I is modeled as a convex combination of a foreground image F and a background image B as $I = \alpha F + (1 - \alpha)B$, and matting techniques try to estimate the alpha matte α (and sometimes with F) from I with the help of additional constraints provided by the user. Once estimated, the alpha matte can be used as a soft mask for applying a variety of object-based editing operations.

Recently proposed matting techniques are capable of generating fairly accurate mattes for images with smooth regions and homogeneous color distributions, as demonstrated in the quantitative studies conducted in [1], [2] and [3]. The test images used in these studies usually contain a single or few dominant foreground colors which remain stable towards the foreground boundary, along with significantly blurred backgrounds. In this case the smoothness assumption on image statistics made in these approaches typically holds, leading to satisfying results.

Unfortunately, as we will demonstrate later, for images containing textured foreground and/or background regions, the performance of these approaches degrades rapidly. The reason is twofold. First, most approaches assume foreground and background colors remain constant or vary smoothly in a local window. This assumption will not hold over strong edges inside the foreground or background region. Second, alpha values are often estimated in an aggressive way in previous approaches. In order to fully capture the fine details of fuzzy objects such as hair

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 239–250, 2011.

and fur, previous methods try to estimate fractional alpha values for all pixels under consideration, which often leads to erroneous mattes. We argue that both limitations come from the inductive inference nature of these approaches.

One way to solve this problem is to always require the user to provide an accurate trimap where most pixels are marked as either foreground or background, and only transparent pixels are marked as unknown. However this is often a labor-intensive process. To improve matting performance over complex images with less accurate trimaps, we treat the matting task as a transductive statistical inference, under the assumption that new foreground and background regions may exist in the unknown region of the given trimap (see Figure 5). These new regions are close, but not equal to user-marked foreground and background regions in some feature spaces. With transductive inference, our method is able to identify these regions and mark them correctly as either definite foreground or background, and only estimate fractional alpha values for real mixed pixels, which is not possible for an inductive inference setting. To the best of our knowledge our method is the first to explicitly solve the matting problem as a transductive inference.

A quantitative evaluation is conducted on different data sets. Experimental results suggest that our algorithm outperforms previous approaches on highly-textured images in terms of both accuracy and robustness.

2 Related Work

Recent image and video matting approaches have been well summarized in a comprehensive survey in [1]. They are classified into three categories, sampling-based, affinity-based, and combined approaches.

Given a user-specified trimap, sampling-based approaches collect a set of nearby known F and B colors, and use them as close approximations of the true F and B colors of unknown pixels, which leaves alpha estimation to be relatively straightforward. Earlier representative sampling-based techniques include Ruzon and Tomasi's method [4] and Bayesian matting [5]. The recent Robust matting algorithm [2] proposes an improved color sampling procedure to selectively evaluate color samples, which is further improved in [6]. All these methods use color samples in an inductive way: user-specified known pixels are used as training data to build parametric or nonparametric models, and then the models are applied to unknown pixels for alpha estimation. For complex images, if the sampled colors do not represent the true B and F colors of unknown pixels, these methods tend to produce large errors.

Affinity-based approaches define constraints on the gradient of the alpha matte based on local image statistics. Poisson matting [7] estimates the matte by solving a set of Poisson equations. The random walk matting algorithm [8] uses the classic exponential affinity for matting. The geodesic matting technique [9] measures the weighted geodesic distances that a random walker will travel from an unknown pixel to reach the foreground and the background, and use the distance ratio as the alpha value. The closed-form matting [3] derives a matting Laplacian by assuming that F and B colors are a linear mixture of two colors in a small window, which is used also in the automatic Spectral matting approach [10] and the mylti-layer matting system [11]. For complex images with large local color variations, the smoothness assumptions often do not hold, leading to less accurate results.

Combined approaches integrate sampling methods and matting affinities together through an optimization process. Representative techniques include the iterative matting approach [12], Easy matting [13], Robust Matting [2], and the high-res matting system [14]. Although combined approaches often generate higher quality mattes [1], the inductive nature of these approaches limits their performance on complex images, as we will demonstrate later.

Our work is also inspired by recent success on applying transductive inference for image segmentation [15]. This method is based on the Laplacian graph regularizer, and segmentation is modeled as finding a labeling function (alpha matte) which is only allowed to vary in low density areas in the feature space. Although it estimates continuous alpha values in the intermediate step, this algorithm does not accurately model the shape of the matte in the foreground-to-background transition area, thus is not able to generate accurate mattes.

3 Transductive vs. Inductive Matting

In machine learning tasks, transductive inference is often employed in cases where both labeled (training) and unlabeled (test) data is presented. Since all the data is available, transductive inference algorithms will take this advantage and produce a mapping function in such a way that the statistics of unlabeled data is also respected. In inductive inference, the test data is unknown beforehand, thus the mapping function is designed solely on the training data to map any possible input data to the output space. Obviously, inductive inference has a higher requirement on the "quality" of the training data, or in other words, how well the limited training data can represent the statistical characteristics of the test data. In areas of the feature space where test data may exist but no training data has been collected, inductive inference tends to make mistakes, as visualized in Figure 2 in [15].

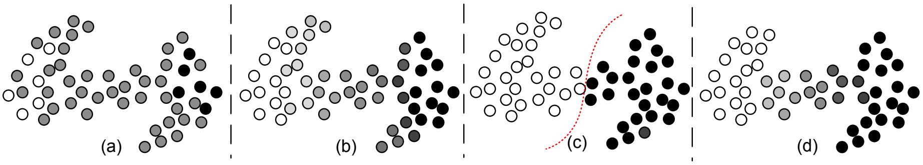


Fig. 1. (a). F(white), B(black) and U(gray) for matting. (b). Previous matting approaches will generate fractional αs for both mixed points and new F and B points. (c). Transductive segmentation generates a binary classification. (d). Our algorithm generates fractional αs for mixed points and also labels new Fs and Bs correctly.

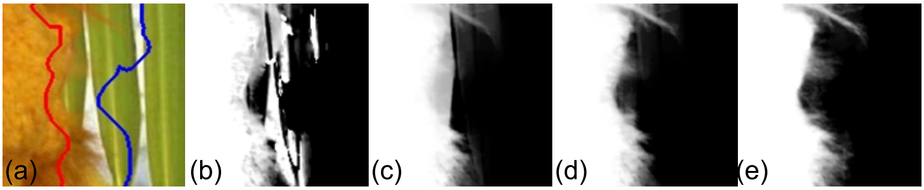


Fig. 2. (a). Input image with trimap boundaries. (b) Bayesian matting. (c). Closed-form matting. (d). our method. (e). ground-truth.

In the matting task, we assume the user has already provided a relatively loose trimap where the majority of pixels have been marked as either F or B as training data, and the rest are marked as U as test data. Since U is known, matting can be treated naturally as a transductive inference, which has a number of advantages over inductive inference when the color distribution in U is complicated and is not fully captured by both F and B. An analytic example is shown in Figure 1. Suppose F, B and U are distributed as in 1(a) in the feature space, note that mixed points (points between the two clusters) as well as new Fs and Bs are unmarked. Previous matting approaches tend to generate fractional αs aggressively, thus will label new Fs and Bs with fractional αs (1(b)). Transductive segmentation can label new Fs and Bs correctly, but is not able to generate correct αs for mixed points (1(c)). Our proposed method can deal with unlabeled data correctly as shown in 1(d).

A real example is shown in Figure 2, which is generated using one of the ground-truth foreground objects in the data set proposed in [2]. In this local region shown in 2(a), the background is highly textured and the white region between green leafs in U is not marked as B (in local sense). Consequently, previous approaches have difficulties to deal with the white region and its edges, as shown in 2(b)-(d). Our method is able to correctly identify the white pixels as new background colors, and generate a matte that is much closer to the ground-truth.

4 The Algorithm

4.1 Optimization Formulation

Our algorithm is designed to explicitly meet the following three objectives:

1. it should be able to identify new F or B colors presented in U;
2. it should be able to correctly label new F and B colors;
3. it should be able to accurately estimate αs for real mixed pixels in U.

Previous approaches mostly ignore Objective 1 and 2 and only focus on Objective 3. We show here how additional transductive inference can be added to meet all three objectives.

Recall that in the transductive segmentation work [15], the labeling functions f is only allowed to vary in low-density regions in the feature space, and segmentation is modeled as the following optimization problem:

$$\min_{f} \sum_{i\in\{F,B\}} c_i \left[Y_i - f(X_i)\right] + \int_U \|\Delta f\|^2 p^s dV, \tag{1}$$

where the summation is over all known pixels, X_i are feature vectors of unknown pixels and Y_i are user-provided labels. c_i is a positive weight controlling how much we want to trust the known labels, which typically is set to be $+\infty$. The integral term is a *s-weighted Laplacian operator* which only allows f to vary where the density estimation p is low. Given the fact that a direct solution of this optimization cannot be obtained, graph Laplacian methods are used to solve for its discrete approximation:

$$\min_{\boldsymbol{\alpha}\in\Re^n} \sum_{i\in\{F,B\}} c_i \left[Y_i - f(X_i)\right] + \boldsymbol{\alpha}^t L\boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{\alpha}$ is the vector of α values of all pixels, and L is the Laplacian matrix whose coefficients are determined by a kernel function $k(X_i, X_j)$ (for instance a Gaussian kernel), which measures the similarity between two feature vectors X_i and X_j.

However, this approach cannot be directly applied to the matting problem as for fuzzy objects, a large number of pixels may present fractional αs, thus the density estimation p does not necessarily correspond to where the alpha matte should vary. In other words, mixed pixels may form high density regions in the feature space, and directly optimizing Equation 1 will force the alpha matte to stay constant is these regions, resulting in matting errors.

In our algorithm we force f to vary not in low density regions, but in *high density regions of real mixed pixels*. This can be achieved if we have a mixed pixel detector, which for each X_i calculates a probably γ_i, indicating how likely this pixel has a fractional alpha value. With this detector X_i can thus be decomposed into two components: $X_i = \gamma_i X_i + (1-\gamma_i)X_i$. Let $X_i^m = \gamma_i X_i$ and $X_i^b = (1-\gamma_i)X_i$ be two subsets, applying density estimation on subset X_i^b will allow f to vary in the correct regions. Denoting p_b as density of X_i^b, we then replace p in Equation 1 with p_b. Furthermore, If we relax the kernel function and allow each X_i to be associated with a weight w_i, and define the weighted kernel function as $k(X_i, X_j, w_i, w_j) = w_i w_j k(X_i, X_j)$, the discrete approximation of the modified optimization problem becomes:

$$\min_{\boldsymbol{\alpha}\in\Re^n} \sum_{i\in\{F,B\}} c_i \left[Y_i - f(X_i)\right] + \boldsymbol{\alpha}^t L^b\boldsymbol{\alpha}, \tag{3}$$

where in the Laplacian matrix L^b the similarity between two pixels is computed as:

$$k^b(X_i, X_j) = \tilde{k}^b(X_i, X_j, 1-\gamma_i, 1-\gamma_j). \tag{4}$$

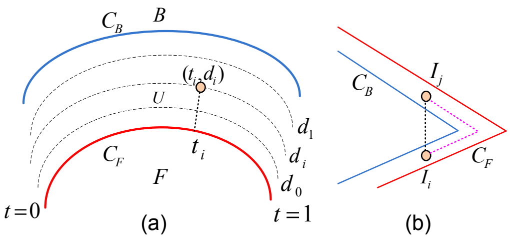


Fig. 3. (a). Parametrization of the unknown region. (b). Respecting sharp corners when computing distance between I_i and I_j.

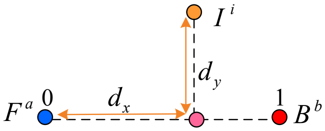


Fig. 4. Linearity analysis among F^a, B^b and I^i

The solution of the updated optimization problem in Equation 3 will satisfy the Objective 1 and 2, however not 3, since the exactly shape of f is not characterized for mixed pixels. To achieve this we add another term to the optimization problem as

$$\sum_{i \in U} \lambda_i \gamma_i^2 \|\alpha_i - \hat{\alpha}_i\|^2 + \boldsymbol{\alpha}^t L^m \boldsymbol{\alpha}, \tag{5}$$

where $\hat{\alpha}_i$ is the prior alpha value for X_i (data term), L^m is a matting Laplacian defined by a matting kernel function $k^m(X_i, X_j) = \tilde{k}^m(X_i, X_j, \gamma_i, \gamma_j)$. λ_i is a weight to balance the two terms.

Combining 3 and 5, the final optimization problem is defined as

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \Re^n} \sum_{i \in \{F,B\}} c_i \left[Y_i - f(X_i)\right] + \\ \sum_{i \in U} \lambda_i \gamma_i^2 \|\alpha_i - \hat{\alpha}_i\|^2 + \boldsymbol{\alpha}^t (L^b + L^m) \boldsymbol{\alpha}. \end{aligned} \tag{6}$$

In our system we usually set $c_i = +\infty$ as we treat the input trimap as a hard constraint, and fix $\lambda_i = 0.1$ as it generates good results in our tests. In next sections we will describe how we compute γ_i, $\hat{\alpha}_i$, and the two kernel functions $\tilde{k}^b$ and $\tilde{k}^m$.

4.2 Kernel Functions

In our system for a pixel i the feature vector X_i contains three components: the RGB color vector I_i, the color level of a local 3×3 patch I_i^p as the texture feature, and a geometric position G_i which is not defined by the absolute (x, y) coordinates of the pixel, but by its relative location in the unknown region parameterized using level set curves, as shown in Figure 3a. We first parameterize the boundary curve of the F region as $C_F(t)$, then apply a distance transform in U to parameterize the region using level set curves. for pixel i, G_i is parameterized as (t_i, d_i).

This parametrization allows us to compare relative locations of two points in U instead of on the image lattice, thus shape corners of the foreground boundary can be respected. An illustration is shown in 3b, where two pixels I_i and I_j are on the two sides of a sharp corner. If we use absolute coordinates, the two pixels have a short distance and a strong affinity value, which will encourage α_i and α_j to be the same, thus statistical changes along C_F will not be respected. Using our parametrization these two points will have a much smaller affinity value.

The primary goal of the Laplacian L^b is to classify new Fs and Bs in a binary sense. We use a weighted Gaussian kernel for it as

$$\begin{aligned}\tilde{k}^b(X_i, X_j, a_i, a_j) = a_i a_j \exp(-(\|I_i - I_j\|^2/2h_c^2 + \\ \|I_i^p - I_j^p\|^2/2h_p^2 + \|G_i - G_j\|^2/2h_g^2)).\end{aligned} \tag{7}$$

To reduce the complexity, similar to [15], we use a truncated version of the Gaussian kernel for the geometric distance, by setting $h_g = 1.25$ and applying a threshold at 0.05. In this way L^b becomes a sparse matrix. h_c and h_p are color and texture variances which can be either fixed as user-specified constants, or computed dynamically using local image statistics as proposed in [16]. We found the latter usually works better when the input image contains both smooth regions and textured regions.

Recall that in [15], the kernel is further normalized as

$$k(X_i, X_j) = \frac{\tilde{k}(X_i, X_j)}{[\tilde{d}(X_i)\tilde{d}(X_j)]^\tau}, \tag{8}$$

where $\tilde{d}(X_i) = \sum_{j=0}^n \tilde{k}(X_i, X_j)$, and $\tau = 1 - s/2$ (s is the free parameter in Equation 1). However, in our system we do not want this normalization to happen since each X_i is associated with a weight, and normalizing the kernel will undesirably cancel out the effects of the weights. We thus set $s = 2$ and $\tau = 0$ for both L^b and L^m.

Denoting W^b as the $n \times n$ matrix where $W_{ij}^b = k^b(X_i, X_j)$ (see Equation 4 and 7), D^d as the diagonal $n \times n$ matrix where $D_{ii}^b = \sum_{j=0}^n k^b(X_i, X_j)$, then L^b is defined as $L^b = D^b - W^b$.

The goal of L^m in the optimization problem 6 is to accurately estimate alpha values for real mixed pixels in U, which have been extensively studied in previous matting approaches. Although the same weighted Gaussian kernel can be defined for L^m as described in [8], the recently proposed matting Laplacian [3] has been shown to be able to generate the most accurate mattes among affinity-based matting approaches [1]. In our system we use this affinity and define $\tilde{k}^m(X_i, X_j, a_i, a_j) = a_i a_j \mu(i, j)$, where $\mu(i, j)$ is the matting Laplacian coefficient defined in Equation 12 in [3]. Similarly, we define W^m as the $n \times n$ matrix where $W_{ij}^m = k^m(X_i, X_j) = \tilde{k}^m(X_i, X_j, \gamma_i, \gamma_j)$, D^m as the diagonal $n \times n$ matrix where $D_{ii}^m = \sum_{j=0}^n k^m(X_i, X_j)$, and L^m as $L^m = D^m - W^m$.

4.3 Estimation of γ_i and $\hat{\alpha}_i$

Recall the convex combination assumption of the matting problem: $I = \alpha F + (1 - \alpha)B$. Under this assumption, and given a relatively tight input trimap (compared with a few scribbles), we assume that if a pixel I_i can be well approximated as a linear combination of a known foreground color $\hat{F}$ and background color $\hat{B}$, then it has a higher probability to be a mixed pixel. Similar to the sampling scheme proposed in [2], for an unknown pixel I_i, we sample a relatively large number of nearby foreground and background colors $F^k, B^k, i = 1, ..., M$, and try to find a good linear approximation of I_i among them.

Specifically, for a sample pair $(F^a, B^b)(a, b \in [1, M])$, we first normalize the distance $|F^a - B^b|$ to 1 and align B^b to $(0, 0)$ and F^a to $(1, 0)$ in the 2D plane defined by the three points F^i, B^j and I_i in the 3D color space, as shown in Figure 4. We then compute the coordinates of I_i in this plane as (d_x, d_y), and compute an estimated α and a mixture probability γ as

$$\hat{\alpha}(F^a, B^b, I_i) = \varGamma(d_x), \tag{9}$$

$$\gamma_{a,b,i} = P(\hat{\alpha}) \cdot exp\left(-\frac{\delta(d_y - \varepsilon_y)(d_y - \varepsilon_y)}{\sigma_y}\right), \tag{10}$$

where $\varGamma(x)$ is a truncation function whose output is 1 if $x > 1$, 0 if $x < 0$, and x otherwise. $\delta(x)$ is a standard step function where $\delta(x) = 1$ for $x \geq 0$ and $\delta(x) = 0$ otherwise. ε_y and σ_y are two constants which are empirically chosen as $\varepsilon_y = 0.1$ and $\sigma_y = 0.025$, which generate good results in our tests. Intuitively, if I_i is closer to the line, which means d_y value is smaller, then γ is higher, indicating the three points can be better approximated using a line in the color space. $P(\hat{\alpha})$ is a weighting function defined as

$$P(\hat{\alpha}) = 4\hat{\alpha}(1 - \hat{\alpha}). \tag{11}$$

$P(\hat{\alpha})$ has its maximal value of 1 at $\hat{\alpha} = 0.5$ and gradually goes to 0 as $\hat{\alpha}$ approaches either 0 or 1. The intuition for applying such a weighting function is that if $\hat{\alpha}$ is closer to 0 or 1, I_i is closer to known F and B and actually has a higher probability to be a new foreground or background color.

We do this analysis for every pair of (F^a, B^b), and the top three pairs are chosen which generate the highest $\gamma_{a,b,i}$, and their average γ_i and $\hat{\alpha}_i$ are computed as the final results for I_i at the color sampling step.

Finally, individually estimated γ_i is still somewhat noisy, since there is no spatial smoothness constraint in γ_i estimation. However, for two neighboring pixels I_i and I_j, if their colors are similar, then their mixture probabilities γ_i and γ_j should also be close. To generate a smoother mixture map which respects the local image statistics, we apply the matting affinity proposed in [3] as a spatial smoothness constraint for the mixture map, and use the matting Laplacian coefficients defined in that approach as smoothing weights between neighboring pixels. Mathematically, the smoothing operation is applied as

$$\gamma_i^{t+1} = (1 - \lambda_s)\gamma_i^t + \lambda_s \sum_{j \in N(i)} (\mu(i, j) \cdot \gamma_j^t) / \sum_{j \in N(i)} \mu(i, j), \tag{12}$$

where $N(i)$ is a 3×3 window centered at i. $\mu(i, j)$ are coefficients in the matting Laplacian matrix. t stands for smoothing iteration, which is fixed to be 20 in our system. λ_s is the step width parameter which is set to be 0.5.

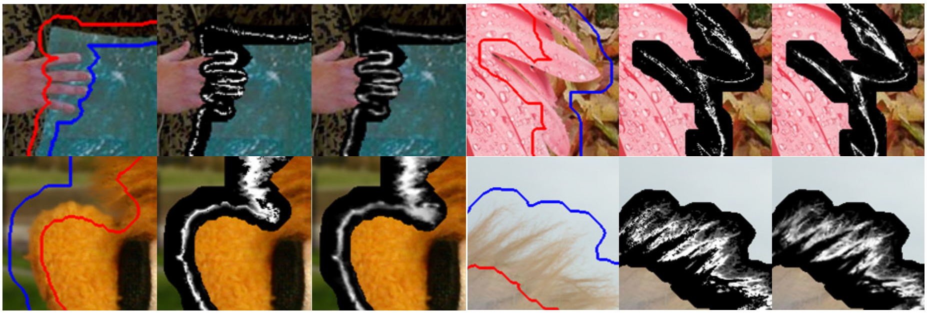


Fig. 5. Example of estimated mixture maps. Each example from left to right: original image with trimap boundaries, γ_i before adaptive smoothing, γ_i after adaptive smoothing.

Figure 5 shows some examples of estimated mixture probability maps. Note how the mixture maps capture the real foreground edge for near-solid boundaries (first row) as well as large fuzzy regions (bottom row).

4.4 Iterative Refinement

One may have noticed that the mixture map estimation largely depends on the available F and B training data. After the optimization problem in Equation 6 is solved as a large linear system, some pixels in the unknown region may have been classified as F and B, giving us new F and B samples which could be used to refine the mixture map estimation. In this way the whole process can be iterated until convergence. The convergence is guaranteed since the upper limit of the number of possible new F and B samples is all the pixels in U, and in practice we found the matte usually becomes stable after 2 to 3 iterations.

5 Link with Other Approaches

Many previous matting and segmentation approaches can be treated as special cases of the proposed algorithm. If we simply set $\gamma_i = 1$ everywhere in U, then L^b becomes an empty matrix and our algorithm degrades to a regular matting algorithm, which shares similar components with the state-of-the-art matting algorithms. For example, L^m incorporates the matting Laplacian [3], and the matte prior $\hat{\alpha}$ is computed in a similar way as in [2]. On the contrary, if we set $\gamma_i = 0$ for all pixels, then L^m becomes an empty matrix and the algorithm degrades to a transductive segmentation algorithm which is similar to the one proposed in [15]. By automatically varying γ_i at different regions, our algorithm combines the advantages of transductive labeling and matting together, thus is able to generate accurate alpha mattes in a more robust way.

Some tri-level segmentation algorithms have been recently proposed which are able to generate relatively accurate trimaps based on user-specified scribbles [17,14]. These approaches usually build color models not only for F and

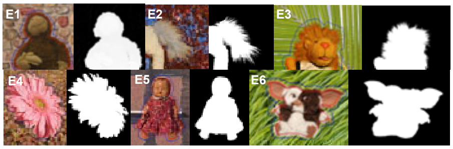


Fig. 6. Six test images containing textured backgrounds, with trimap boundaries overlayed and ground-truth mattes

	Bayesian	Clo.form	Robust	Our
T1	793.2	346.7	392.0	94.6
T2	2395.2	451.2	420.7	280.9
T3	263.0	123.0	152.7	79.6
T4	1786.5	401.9	331.4	117.3
T5	3233.0	339.6	320.1	216.6
T6	971.9	106.5	91.8	55.7

Fig. 7. MSE of mattes generated by different algorithms on the data set in Figure 6

B, but also for U by linearly blending F and B models, thus are similar to the linear mixture analysis proposed in our approach on the concept level. However, our approach differs from these approaches from two major aspects. first, in trimap generation systems the trimap generation and alpha matting are treated as separate steps, thus any errors in trimap generation will be magnified in the matting step. In our system the matting and transductive labeling are integrated together and they help each other. Second, trimap generation methods mostly use inductive inference, relying on the user to provide enough color samples to construct the proper statistical models (for instance Gaussian Mixtures). In our system the new F and B labeling is done under a more robust transductive inference framework.

Nevertheless, one can imagine integrating these technique together to build a more efficient system. Given an input image, a trimap can be interactively generated using trimap segmentation algorithms. Since the resulting trimap will not be perfect where U region may still contain some F and B colors, our algorithm can be applied to improve the matting quality, especially for complex images.

6 Evaluations and Comparisons

To quantitatively evaluate the algorithm, a test data set is constructed which, unlike data sets used in previous approaches, contains highly-textured backgrounds, as shown in Figure 6. For image $E5$ and $E6$ we shoot the foreground dolls against multiple known backgrounds, and use triangular matting methods [18] to extract the ground-truth mattes. The foregrounds and ground-truth mattes in E1 to E4 are borrowed from the data sets in [2] and [3], but we compose them onto more complicated backgrounds to create test images. Note that the data set contains both hairy foreground objects and near-solid ones. For each example a relatively loose trimap is specified as the user input.

Four algorithms are applied on the test data set, including Bayesian matting [5], closed-form matting [3], Robust matting [2], and the proposed transductive matting algorithm. Specifically, Bayesian matting is chosen as a representative

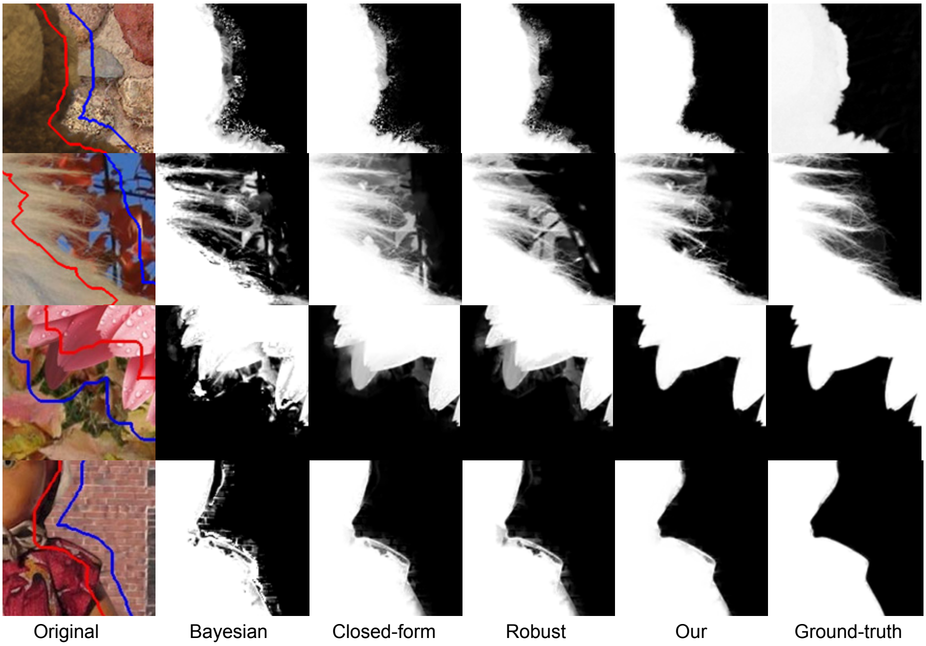


Fig. 8. Partial mattes extracted by different algorithms

Table 1. MSE of mattes generated by our algorithm on the data set in [1], and their ranks among total of 8 test systems. Format: $Min^{rank} : Max^{rank}$.

T1	T2	T3	T4	T5	T6
$58.9^2 : 93.5^2$	$51.6^2 : 142.3^2$	$41.8^2 : 70.5^2$	$74.7^1 : 248.5^3$	$154.2^2 : 355.3^1$	$36.5^3 : 47.6^1$

sampling-based approach, closed-form matting as the most accurate affinity-based method, and robust matting as a well-balanced optimization-based algorithm which combines sampling and affinities.

Figure 7 shows the Mean Squared Errors (MSE) of extracted mattes against the ground-truth. Alpha values are stretched to $0-255$ for MSE calculation. Figure 8 and 2 shows partial mattes generated by different algorithms. There results clear suggest the proposed algorithm outperforms previous approaches on these complex images.

To evaluate the performance of the proposed algorithm on simper images with smooth F and B regions, we apply it on the test data set proposed in [1], which contains 6 test images, each with a ground-truth matte and a series of trimaps. Table 1 shows the MSE values of our extracted mattes, and their ranks comparing with the other 7 matting algorithms. The results suggest that the proposed algorithm performs comparably well with other matting methods when the input image does not contain complex textures.

7 Conclusion

Previous inductive-inference-based matting algorithms tend to produce erroneous mattes when dealing with textured objects. In this paper we propose a transductive matting algorithm which explicitly models the trimap-based matting task under a transductive inference framework, thus not only is able to produce higher quality results on textured or non-homogeneous images, but also can produce accurate mattes for regular images with smooth foreground and background regions.

References

1. Wang, J., Cohen, M.: Image and video matting: A survey. Foundations and Trends in Computer Graphics and Vision 3(2), 97–175 (2007)
2. Wang, J., Cohen, M.: Optimized color sampling for robust matting. In: Proc. of IEEE CVPR (2007)
3. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Analysis and Machine Intelligence 30, 228–242 (2008)
4. Ruzon, M.A., Tomasi, C.: Alpha estimation in natural images. In: Proceedings of IEEE CVPR, pp. 18–25 (2000)
5. Chuang, Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proceedings of IEEE CVPR, pp. 264–271 (2001)
6. Rhemann, C., Rother, C., Gelautz, M.: Improving color modeling for alpha matting. In: Proc. of BMVC (2008)
7. Sun, J., Jia, J., Tang, C.-K., Shum, H.-Y.: Poisson matting. In: Proceedings of ACM SIGGRAPH, pp. 315–321 (2004)
8. Grady, L., Schiwietz, T., Aharon, S., Westermann, R.: Random walks for interactive alpha-matting. In: Proceedings of VIIP 2005, pp. 423–429 (2005)
9. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: Proc. of IEEE ICCV (2007)
10. Levin, A., Rav-Acha, A., Lischinski, D.: Spectral matting. In: Proc. of IEEE CVPR (2007)
11. Singaraju, D., Vidal, R.: Interactive image matting for multiple layers. In: Proc. of IEEE CVPR (2008)
12. Wang, J., Cohen, M.: An iterative optimization approach for unified image segmentation and matting. In: Proceedings of ICCV 2005, pp. 936–943 (2005)
13. Guan, Y., Chen, W., Liang, X., Ding, Z., Peng, Q.: Easy matting. In: Proc. of Eurographics (2006)
14. Rhemann, C., Rother, C., Rav-Acha, A., Sharp, T.: High resolution matting via interactive trimap segmentation. In: Proc. of IEEE CVPR (2008)
15. Duchenne, O., Audibert, J.-Y.: Segmentation by transduction. In: Proc. of IEEE CVPR (2008)
16. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: Proc. of IEEE CVPR (2001)
17. Juan, O., Keriven, R.: Trimap segmentation for fast and user-friendly alpha matting. In: Proc. of IEEE Workshop on VLSM (2005)
18. Smith, A.R., Blinn, J.F.: Blue screen matting. In: Proceedings of ACM SIGGRAPH, pp. 259–268 (1996)

A New Buckling Model for Cloth Simulation

Tung Le Thanh and André Gagalowicz

Rocquencourt INRIA, France

Abstract. Textiles are normally incompressible: when we try to compress them, they immediately buckle. Unfortunately, many cloth simulation solvers disregard this fact. In this paper, we present an efficient method to model buckling using distance contraint. This constraint is formulated as a linear complementarity problem (LCP) and can be easily integrated within a collision handling process.

1 Introduction

Buckling for thin materials is one of the most challenging problems in computer animation. Accurate simulation of cloth buckling is important for a wide variety of natural phenomena and has numerous applications, including video games and fashion.

Many researches point out that buckling occurs when an object cannot resist to a compressing force and creates a disequilibrium status. As for cloth materials, when a compression force is added to them, at first cloth can stand force and keep the same shape while after a critical moment it reaches an unstable state producing a huge shape change.

Cloth simulation would not be noticeably realist without including buckling. However, simulation of buckling involves several important difficulties. One such problem is keeping cloth incompressible during the simulation without losing physical dynamical effect.

Most of the methods for buckling are based upon geometrical constraints. For example, Choi [6] presented a post buckling model based upon the construction of a post-buckling shape in every spring of a cloth model. Decaudin [8] studied the procedure to model a pressed cylinder cloth and generated buckling effect using a diamond geometrical hypothesis. Both of existing methods generate a buckling effect from a predefined shape instead of from the dynamics rules of a cloth simulation system, which makes the simulation not very realistic. Therefore, it is critical to develop a physically-based buckling model combined with the cloth simulation procedure to generate fast and realistic buckling effects.

In this paper, we propose a new method using distance contraints formulated as a linear complementarity problem which fits to cloth simulation. The simulation uses a mass-spring representation to model cloth physically. The buckling disturbance is applied to the system when some springs are compressed. In our method, a linear complementarity problem (LCP) is incorporated into our collision treatment process. This allows us to use only one LCP solver for the

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 251–261, 2011.

distance contraints and collisions treatment. Our experimentation with our dynamics solver and textile Kawabata evaluation system (KES) parameters shows realistic results.

The remainder of this paper is organized as follows. We review related work in Section 2. Our new method for buckling is presented in Section 3. It is incorporated in implicit collision handling in Section 4. Results and discussion are presented in Section 5.

2 Related Work

In most previous works, the problem of distance constraint in textile tension modeling is solved using stretch resistance (Choi and Ko [7] have an excellence survey on cloth simulation).

The general approach is to treat cloth as an elastic material [20] [3] [2] [5]. To reduce stretching, elastic models adopt sometimes stiff springs. Unfortunately, stiff springs system degrades the numerical stability of the solver [10].

We class the constraint-based approach in two categories depending on the implementation of the solvers used: individual constraint or global constraint.

Individual Constraint. Provot [17] presented a first method for spring length control; the solver iteratively displaces the vertices related to stretched springs. However, he found a poor convergence since each displacement may stretch neighbor springs.

When tight tolerances of cloth are not required, the Provot's method was used widely because it's simple to implement. Bridson [4] limits the changing of spring length per timestep to 10% of the current length. Müller [14] used position based approach to enforce inextensibility on each spring separately.

Global Constraint. In contrast to iterative constraint enforcement, House et al. [12] used Lagrange multipliers to treat stretching. Their approach alleviates the difficulties associated with poor numerical conditioning and artificial damping. House et al. later encountered difficulties in handling collision response within their late works [13].

Hong et al. [11] used a linearized implicit formulation in order to improve stability of constrained dynamics. Tsiknis [21] proposed triangle-based strain limiting together with a global stitching step for stable constraint enforcement. This allowed for larger time-steps and reduced the need for springs to maintain the cloth on the constraint manifold. Both of these approaches enforce inextensibility only for strain exceeding 10%.

Recently, Goldenthal and al. [9] used a projection method based upon Constrained Lagrangian Mechanics to produce inextensible cloth. However their method cannot deal with real behavior cloth, that act with KES (Kawabata Evaluation System) parameters.

3 Buckling Model for Cloth Simulation

Section 2 was related to the problem of textile stretching. In order to model buckling, we need, on the contrary to deal with textile compression problems.

In this section, we introduce the necessary background material on which we build our method. This consists of a buckling disturbance and contraint force computation from compressed springs. We will present briefly a review of constrained dynamics.

3.1 Buckling Strategy

In general, cloth simulation using mass-spring systems encountered difficulties of in-plane compressing. This problem is due to the fact that gradient of constraints is also in-plane and a compressed point never moves away from the plane.

Our global strategy consists in suppressing all compressed springs as textile does not put up with compression. In order to obtain a convergent algorithm, we fully decompress the most compressed one (by displacing one of the two masses of the corresponding springs orthogonally to the surface) and we continue by choosing the next most compressed spring (it will be compulsorily less compressed than the former one) until the stack of compressed spring is empty. The decompression of the chosen compressed spring is realized in the following way: if AB is the most compressed spring and the compression of AD is greater than that of BE and if the compression of AC is greater than the compression of AF, we pull the mass A out of the plane to its destination position A1 (geometrically ideal) using the ABCD modifying schema shown on the left of figure 2. In general, for this configuration, the position A_1 is the intersection point of three spheres, the ray of which is equal to l_0, the rest length of the springs (supposed

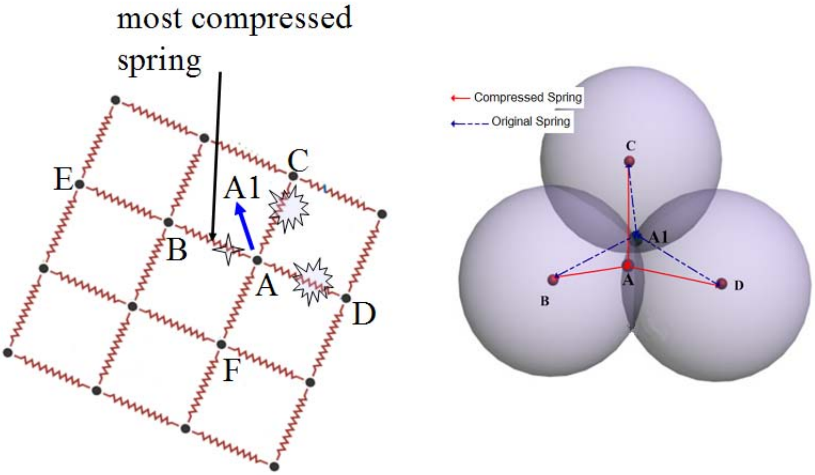
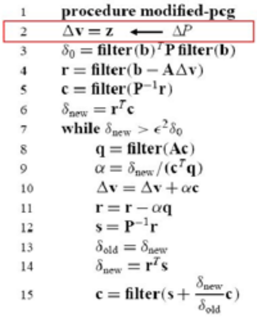


Fig. 1. Compressed springs and its decompression



```
procedure modified-pcg
$\Delta\mathbf{v} = \mathbf{z} \longleftarrow \Delta P$
$\delta_0 = \mathbf{filter}(\mathbf{b})^T \mathbf{P}\, \mathbf{filter}(\mathbf{b})$
$\mathbf{r} = \mathbf{filter}(\mathbf{b} - \mathbf{A}\Delta\mathbf{v})$
$\mathbf{c} = \mathbf{filter}(\mathbf{P}^{-1}\mathbf{r})$
$\delta_{new} = \mathbf{r}^T\mathbf{c}$
while $\delta_{new} > \epsilon^2\delta_0$
    $\mathbf{q} = \mathbf{filter}(\mathbf{A}\mathbf{c})$
    $\alpha = \delta_{new}/(\mathbf{c}^T\mathbf{q})$
    $\Delta\mathbf{v} = \Delta\mathbf{v} + \alpha\mathbf{c}$
    $\mathbf{r} = \mathbf{r} - \alpha\mathbf{q}$
    $\mathbf{s} = \mathbf{P}^{-1}\mathbf{r}$
    $\delta_{old} = \delta_{new}$
    $\delta_{new} = \mathbf{r}^T\mathbf{s}$
    $\mathbf{c} = \mathbf{filter}(\mathbf{s} + \frac{\delta_{new}}{\delta_{old}}\mathbf{c})$
```

Fig. 2. Integrating of position constraint Δp in modified gradient conjugated algorithm

to have the same lengthes in our case). We choose, between the two possible intersections A_1, that one which does not change the sign of the curvature and choose it randomly if the surface is flat.

The position correction is done directly in our dynamical solver. We have used the modified gradient conjugated algorithm presented in the works of Baraff and al. [2]. $\overrightarrow{AA_1}$ is simply the position correction Δp which is considered as a position constraint and forces the simulator to solve the dynamical equation with this constraint. The modified algorithm is shown as follows:

However, this method has some limitations, all in-plane compressed points cannot be treated at the same time but a parallel version of this technique is in preparation.

Testing has shown that, the in-plane compression problem can be treated correctly but computing time is presently enormous. However, this test gives us the possibility to study in details the interactions between the control of buckling and that of collisions which is an important problem not yet fully solved.

Figure 3 and 5 show the experimentation results, more details will be presented in section 5 (Videos will be shown during the oral presentation).

4 Buckling Model Incorporated in the Collision Handling

We used the implicit contact handling for deformable objects presented by Otaduy [15] and combined our buckling within the collision solver. This method can handle complex and self-collision situations.

4.1 Numerical Integration and Dynamics

We consider a physical system governed by the ordinary differential equation:

$$\begin{aligned} \mathbf{M}\dot{v} &= f(x, v) + f_{ext} \\ \dot{x} &= v \end{aligned}$$

where x denotes a state vector, v denote the velocity vector, $\mathbf{M}$ denotes the mass matrix, $f(x, v)$ denote the internal forces and f_{ext} denote the external forces (such as those due to gravity, etc.).

We have used time discretization methods that have been used or described in depth in the research by Baraff and Witkin [2], Otaduy and al. [15], Pabst et al. [16], etc.

Given a state $S(x_0, v_0)$ at the beginning of a time step, the velocity v is updated by this linear equation:

$$\mathbf{A}v = \mathrm{b}$$

We have used a backward Euler scheme with linear approximation of forces and assumed a constant mass matrix per time step. The linear velocity update rule is rewritten as:

$$\mathbf{A} = \mathbf{M} - h\frac{\partial f}{\partial v} - h^2\frac{\partial f}{\partial x}$$
$$\mathrm{b} = hf_0 + v_0(\mathbf{M} - h\frac{\partial f}{\partial v})$$

4.2 Implicit Collision Handling

The basic concept for the implicit contact handling is a non penetration constraint that can be described briefly as follows.

The set of object q configurations free of contact can be limited by a constraint manifold in a high-dimensional configuration space G. Collision detection locally samples this constraint manifold. If we group all contact points in one vector p, the free space defined by the constraint manifold G can be approximated by a set of algebraic inequalities $g(p) > 0$.

For example, if we have a contact point p_a and a normal vector n of the object surface at p_a, a non-penetration constraint at p_a can be satisfying : $g(p) = n^T(p - p_a) \geq 0$

In order to enforce non-penetration at the end of the time step, we can formulate the constraints implicitly. We propose a semi-implicit formulation of contact constraints linearized as:

$$g(p) = g_0 + \frac{\partial g}{\partial p}(p - p_0) \geq 0 \tag{1}$$

with the rows of the Jacobian $\frac{\partial g}{\partial p}$ formed by the contact normal n at the time of impact and $g_0 = g(p_0)$.

Note that the contact point p can be found anywhere in the object surface. In our cloth simulation system, p is represented by the three end-points of a triangle containing p:

$$p = p_a\alpha + p_b\beta + p_c(1 - \alpha - \beta)$$

we obtained:

$$\dot{p} = \dot{p_a}\alpha + \dot{p_b}\beta + \dot{p_c}(1 - \alpha - \beta) = \frac{\partial \dot{p}}{\partial v}v$$

if we rewrite $p = p_0 + h\dot{p}$ we obtain:

$$g(p) = g_0 + \frac{\partial g}{\partial p}\frac{\partial \dot{p}}{\partial v} vh \geq 0 \tag{2}$$

we obtained :

$$\mathbf{J}v \geq -\frac{1}{h} g_0 \tag{3}$$

where $\mathbf{J} = \frac{\partial g}{\partial p}\frac{\partial \dot{p}}{\partial v}$ and $\frac{\partial \dot{p}}{\partial v}$ denote the barycentric coordinates of the contact point.

We used the collision detection method presented in the research of Provot [18], readers can find more details in his papers for edge-edge, point-triangle collision detection technics.

The solution to our constrained dynamics problem alone does not guarantee a penetration-free state at the end of a time step. There are two possible reasons: the linearization of the contact constraints, and the fact that the collision response induced by some constraints may in turn violate other constraints that were not yet accounted for. To overcome this problem, a collision test will be done at the end of each iterative loop to ensure a contact free state.

4.3 In-Plane Incompressible Constraint

We propose a new method to solve the problem of in-plane compressed springs incorporating the collision handling process. In this method, we treat all in-plane compressed springs simultaneously by formulating our problem as a linear complementarity problem (LCP).

Using the same definitions as in section 4.2, the constraints of in-plane compressed springs can be represented by a set of algebraic inequalities $g(p) \geq 0$.

The in-plane compression problem can be represented by two subproblems: spring compression and pulling points out of plane. We firstly define the spring compression problem. Given an in-plane compressed spring r_{ab} connected by a pair of points p_a and p_b and a direction vector u from p_a to p_b, an individual in-compressible constraint can be rewritten as:

$$g_{ab} = u^T(p_b - p_a) - l_0 \geq 0$$

where l_0 is the length of the spring r_{ab} at rest state. In semi-implicit form, the constraint g can be rewritten as:

$$g(p) = g_0 + \frac{\partial g}{\partial p}(p - p_0) \geq 0$$

with each row of Jacobian $\frac{\partial g}{\partial p}$ formed by direction vector u at the time of compressing $g_0 = g(p_0)$. If we rewrite the equation above with $p = p_0 + vh$, we obtained:

$$g_0 + \frac{\partial g}{\partial p} vh \geq 0 \tag{4}$$

note that $g_{0ab} = u^T(p_{0a} - p_{0b}) - l_0$ the deformation of the spring at the beginning of the iterative process.

The in-plane point pull out can be defined as a modification of point position along the direction of surface normal vector. Our idea is to add a very small force to pull the point out of the plane. In reality, cloth is never absolutely flat and try to buckle immediately when it is compressed. We consider a force of the contact surface (e.x: a table) which raises cloth when it is compressed in-plane.

The pull-out constraint is simply defined as:

$$g(p) = g_0 + n^T(p - p_0) \geq 0$$

where g_0 could be set to *zero*. If we rewrite the g(p) as semi-implicite equation, we obtained:

$$g(p) = \frac{\partial g}{\partial p} vh \geq 0 \tag{5}$$

where $\frac{\partial g}{\partial p}$ formed by normal vector of the surface at the point p.

In summary, each in-plane compression constraint is composed by two constraints : incompressible constraint (equation 4) and pull-out constraint (equation 5). The set of constraint equations can be rewritten as a matrix J, where each row of J presented a constraint. We obtained:

$$\mathbf{J}v \geq -\frac{1}{h} g_0 \tag{6}$$

We have seen that our in-plane compression constraint (eq. 6) has the same properties as collisions constraint (eq. 3) and can be combined together. That allows us to solve collision and buckling problem at the same time.

4.4 Linear Complementarity Formulation

We use the method of Lagrange multipliers to model the adjustment forces as $\mathbf{J}^T\lambda$ with $\lambda \geq 0$ ($\lambda > 0$ when a point is compressed). A complementarity condition $0 \leq \lambda \perp g(p) \geq 0$ mean that forces ($\lambda > 0$) cannot be adjusted when a spring is not compressed ($g(p) > 0$).

Here we have a mixed linear complementarity problem (MLCP - combining of equalities and inequalities equations). We denote by v^* the unconstrained velocities (solved from A v = b), the MLCP that define the constrained velocities $v = v^* + \Delta v$ can be presented as follows:

$$\mathbf{A}\Delta v = \mathbf{J}^T\lambda \tag{7}$$

$$0 \leq \lambda \perp \mathbf{J}\Delta v \geq -\frac{1}{h} g_0 - \mathbf{J}v^* \tag{8}$$

4.5 Mixed Linear Complementarity Problem Solver

In order to solve the MLCP, we rewrite our problem as:

$$0 \leq \lambda \perp \mathbf{B}\lambda \geq c$$

where $\mathbf{B} = \mathbf{J}\mathrm{A}^{-1}\mathbf{J}^T$ and $c = -\frac{1}{h} g_0 - \mathbf{J}v^*$.

Using Gauss-Seidel method, we can easily determine λ then solve $A\Delta v = J^T\lambda$ for Δv. In fact, to compute B, we need to compute A^{-1}. This matrix inversion is very expensive when A is order of 3nx3n where n is the number of masses (about 40000 masses to simulate $1m^2$ of cloth). To avoid the matrix inversion, we adapt the *iterative constraint anticipant* method presented in the work of Baraff [1] to solve our LCP.

Given a velocity correction $\Delta v(i-1)$ at an iterative step $i-1$, the Lagrange multipliers $\lambda(i)$ can be computed by:

$$0 \leq \lambda(i) \perp (\mathbf{J}\mathbf{D}_{\mathbf{A}}^{-1}\mathbf{J}^T)\lambda(i) \geq -\frac{1}{h}g_0 - \mathbf{J}v * -\mathbf{J}\mathbf{D}_{\mathbf{A}}^{-1}(\mathbf{L}_{\mathbf{A}} + \mathbf{U}_{\mathbf{A}})\Delta v(i-1) \qquad (9)$$

where $\mathbf{D}_{\mathbf{A}}$ is diagonal, $\mathbf{L}_{\mathbf{A}}$ is strictly lower triangular, $\mathbf{U}_{\mathbf{A}}$ is strictly upper, and $\mathbf{A} = \mathbf{D}_{\mathbf{A}} - \mathbf{L}_{\mathbf{A}} - \mathbf{U}_{\mathbf{A}}$.

The velocity correction $\Delta v(i)$ is refined using block-Jacobi relaxation:

$$\mathbf{D}_{\mathbf{A}}\Delta v(i) = (\mathbf{L}_{\mathbf{A}} + \mathbf{U}_{\mathbf{A}})\Delta v(i-1) + J^T\lambda(i) \qquad (10)$$

We start the *iterative constraint anticipant* with $\lambda(0) = 0$ and $\Delta v(0) = 0$ then we repeat the computation of λ and Δv using equation 9 and 10 until $\mathbf{A}\Delta v - \mathbf{J}^{\mathbf{T}}\lambda \leq \epsilon$ (in our method, we choice ϵ = 10E-12)

5 Results and Discussion

In our work, we performed two typical experiments in the buckling research area.

The first experiment is compressing textile around a cylinder the which is difficult to solve for buckling. Another experiment is the standard case of buckling, when a planar piece of cloth is pushed uniformly along two opposite sides. The planar cloth with in-plane compression is the standard model for verifying buckling effects, since without buckling, a cloth simulation system will come to a numerical unstable state when compression remains in plane and increases up to program divergence. The procedure of cylinder buckling is very similar to the effect produced when you raise up your sleeves around your arm. Therefore if the cylinder buckling experiment can achieve good results, the application to virtual try-on system would also get vivid effects.

The spring length used in our experimentat is $0.005m$ and the number of the masses is 20×20 for the planar cloth and 20×40 for the cylinder model. The time step is 10E-4 in order to get natural and stable results.

Cylinder Experiment: In this experiment the cylinder is produced by a $10cm \times 20cm$ rectangle cloth. At the initial state, there is no external force on the cylinder, then a force pulling down textile uniformly on its top border is added to the cloth through displacement constraint implementation. The energy equivalence condition is the criterion for the force and the threshold is 10E-8J. After adding the force, the cloth simulator computes the following status and add buckling displacement to the system constraint. With the energy criterion, after the system comes to a stable state, the in-plane force will continue to act on the cloth. Figure 3 shows the buckling result.

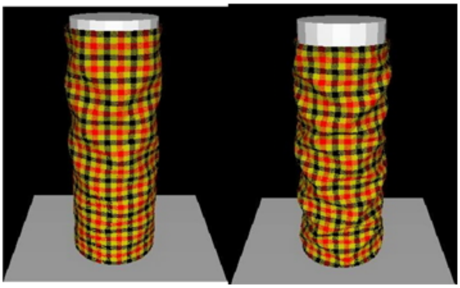


Fig. 3. Cylinder cloth buckling

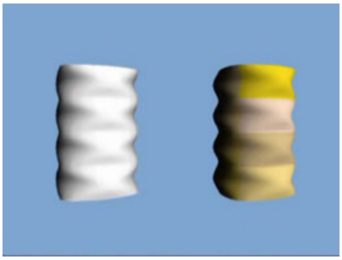


Fig. 4. Comparison: result with the diamond assumption method from [8]

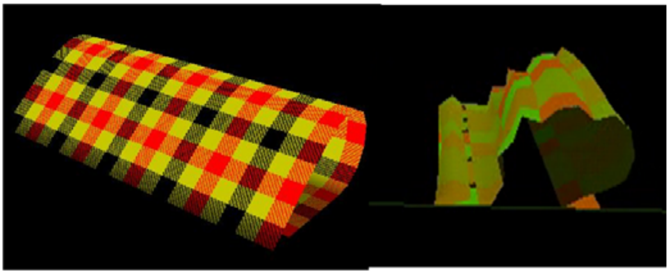


Fig. 5. Comparison: our result on the left and result coming from a shape control using Choi's approach [6]

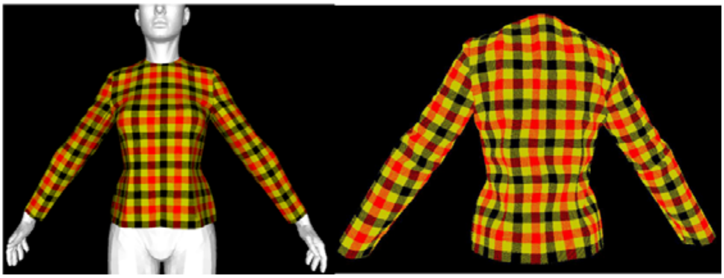


Fig. 6. Buckling results on the body model

In Plane Pulling Cloth Experiment: this experiment uses a $10cm \times 10cm$ cloth. Cloth is laid on a horizontal plane. The initial state of the cloth is rest, then one border of cloth is pushed uniformly along its border, while the opposite one is blocked. It is implemented with system constraint as in our paper. Then we keep on adding the same force to the cloth. Figure 5 is the comparison with the post spring shape assumption method from Choi [6]. Figure 6 is a result of virtual try-on in order to illustrate the validity to the 3D garment simulation.

6 Conclusion and Future Work

In this paper, we have introduced a new buckling model integrated efficiently in collision handling. Using incompressible springs as well as non linear cloth simulations, we found that this method can be used efficiently to model various thin materials. This allows also an efficient analysis of the evolution of contact surfaces over time.

Actually, we are developing a complete virtual try-on system, from design over tailoring until try-on and customization of virtual garments with few interactions required from the user.

In this goal, we would like to further investigate the scalability of our collision handling method in order to speed up the cloth simulation process. Many collisions computing can be treated in a same time using GPU-based streaming method [19]. This would allow us to obtain a final result in a shorter time.

References

1. Baraff, D.: Linear-time dynamics using lagrange multipliers. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, pp. 137–146. ACM, New York (1996)
2. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, pp. 43–54. ACM, New York (1998)
3. Breen, D.E., House, D.H., Wozny, M.J.: Predicting the drape of woven cloth using interacting particles. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, pp. 365–372. ACM, New York (1994)
4. Bridson, R., Marino, S., Fedkiw, R.: Simulation of clothing with folds and wrinkles. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation, SCA 2003, pp. 28–36. Eurographics Association, Aire-la-Ville (2003)
5. Choi, K.-J., Ko, H.-S.: Stable but responsive cloth. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2002, pp. 604–611. ACM, New York (2002)
6. Choi, K.-J., Ko, H.-S.: Extending the immediate buckling model to triangular meshes for simulating complex clothes. In: Eurographics 2003 Short Presentations, pp. 187–191 (2003)
7. Choi, K.-J., Ko, H.-S.: Research problems in clothing simulation. Comput. Aided Des. 37, 585–592 (2005)
8. Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A., Cani, M.-P.: Virtual garments: A fully geometric approach for clothing design. Computer Graphics Forum 25(3), 625–634 (2006)
9. Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., Grinspun, E.: Efficient simulation of inextensible cloth. In: ACM SIGGRAPH 2007 Papers, SIGGRAPH 2007. ACM, New York (2007)
10. Hauth, M., Etzmuss, O., Strasser, W.: Analysis of numerical methods for the simulation of deformable models. The Visual Computer 19, 581–600 (2003), doi:10.1007/s00371-003-0206-2

11. Hong, M., Choi, M.h., Jung, S., Welch, S.: Effective constrained dynamic simulation using implicit constraint enforcement. In: International Conference on Robotics and Automation, pp. 4520–4525 (2005)
12. House, D., Devaul, R.W., Breen, D.E.: Towards simulating cloth dynamics using interacting particles. International Journal of Clothing Science and Technology 8, 75–94 (1996)
13. House, D.H., Breen, D.E. (eds.): Cloth modeling and animation. A. K. Peters, Ltd., Natick (2000)
14. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. J. Vis. Comun. Image Represent 18, 109–118 (2007)
15. Otaduy, M.A., Tamstorf, R., Steinemann, D., Gross, M.: Implicit contact handling for deformable objects. Computer Graphics Forum (Proc. of Eurographics) 28(2) (April 2009)
16. Pabst, S., Thomaszewski, B., Strasser, W.: Anisotropic friction for deformable surfaces and solids. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2009, pp. 149–154. ACM, New York (2009)
17. Provot, X.: Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In: Davis, W.A., Prusinkiewicz, P. (eds.) Graphics Interface 1995, pp. 147–154. Canadian Human-Computer Communications Society (1995)
18. Provot, X.: Collision and self-collision handling in cloth model dedicated to design garments. Computer Animation and Simulation, 177–189 (1997)
19. Tang, M., Manocha, D., Lin, J., Tong, R.: Collision-streams: Fast GPU-based collision detection for deformable models. In: I3D 2011: Proceedings of the 2011 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 63–70 (2011)
20. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 205–214. ACM, New York (1987)
21. Dinos Tsiknis, K., Dinos Tsiknis, C.K.: Better cloth through unbiased strain limiting and physics-aware subdivision. Technical report, The University of British

A Public System for Image Based 3D Model Generation

David Tingdahl and Luc Van Gool

K.U. Leuven, ESAT-PSI

Abstract. This paper presents a service that creates complete and realistic 3D models out of a set of photographs taken with a consumer camera. In contrast to other systems which produce sparse point clouds or individual depth maps, our system automatically generates textured and dense models that require little or no post-processing. Our reconstruction pipeline features automatic camera parameter retrieval from the web and intelligent view selection. This ARC3D system is available as a public, free-to-use web service (`http://www.arc3d.be`). Results are made available both as a full-resolution model and as a low-resolution for web browser viewing using WebGL.

1 Introduction

Computerised 3D models have become part of peoples everyday lives. Indeed, the average home is filling up with 3D capable products, such as 3D-TVs, smart phones, computers, video games, etc. We also see an increased use of 3D on the web, with new standards such as WebGL [1] and XML3D [2]. That said, the *creation* of 3D content has not followed suit. Most 3D content in movies and games are modelled using expensive scanners or manual labour. We attempt to bridge this wide gap between users and producers of 3D content by introducing a public, easy-to-use 3D reconstruction pipeline that outputs textured and dense 3D models, from normal photos as the only user-generated input.

1.1 Previous Work

There already exist a number of public methods for 3D reconstruction. ARC3D [3] is the "mother" of 3D reconstruction services and is also the predecessor of the system presented in this paper. The original ARC3D has been on-line since 2005 and produces dense depth maps that can be (manually) merged into a complete model. Two similar services are Microsoft's well-known PhotoSynth [4] and the more recent 3dTubeMe [5]. Both solutions generate a sparse point cloud which gives an overview of scene geometry but does not provide enough detail for photorealistic visualisation. There do exist previous extensions for dense reconstruction, but they are limited to advanced users. The Bundler software [6] is an open source package for reconstructing a sparse 3D point cloud. It can be combined with the PMVS software [7] to create dense point clouds. This method is intended for computer vision professionals rather than for the general public, and is thus

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 262–273, 2011.

non-trivial to install and operate. In contrast, our ARC3D extension generates a complete and textured 3D mesh, is fully automatic and is easy to use. The user simply uploads images to our web service which takes care of the rest.

All systems mentioned above are based on approximately the same structure from motion fundamentals [8,9]. Local features are extracted and matched between the images. The matches are used to compute relative poses between image pairs and to triangulate a sparse reconstruction. The sparse reconstruction is then optionally upgraded to dense by a multi-view stereo algorithm. The method assumes that the internal parameters of the cameras are known or fixed, which is typically not the case if arbitrary cameras are used for the photos. However, there exist two practically usable methods for computing them. The classical methods start with a projective reconstruction which is upgraded to euclidean by finding the absolute quadric or its dual [10] (but also see [8] for a more intuitive, euclidean interpretation). The euclidean upgrade is analogous to recovering the camera intrinsic parameters and can thus be used to calibrate the cameras. This method provides good results if the camera motion is general, but may break down in cases of turntable motion and planar scenes. More recently, it has been shown that EXIF meta data can be used to approximate a calibration matrix [6]. Since EXIF data may be inaccurate, one has to rely on a subsequent bundle adjustment. Relying on EXIF may increase robustness to degenerate camera motion but may decrease accuracy.

Recent research [11,12] have shown promising results in large scale 3D reconstruction using images from photo-sharing communities. Such methods require an abundance of photos to ensure sufficient overlap and quality. It is thus only applicable to well-photographed, famous landmarks; one can simply not expect a high-quality 3D model out of a limited set of casually taken photographs. A state-of-the-art 3D reconstruction algorithm such as ARC3D is able to produce models with accuracies comparable to a lidar scanner [13]. As our goal is quality rather than quantity, we focus on reconstructions on a smaller scale, at the order of 100 images, taken with the purpose of creating 3D. We encourage but do not enforce ordered images and constant camera parameters.

Our method consists of a sparse reconstruction followed by a dense upgrade (Section 2). We improve the EXIF based pre-calibration by searching the web for missing camera parameters (Section 3). This yields a higher success rate. At the end of the dense reconstruction, we add a new step that first selects suitable views for remeshing (Section 4) and then computes a textured mesh from their depth maps (Section 5). To the best of our knowledge, there are no public 3D reconstruction pipelines available that provide this functionality.

2 Reconstruction Pipeline

Our 3D reconstruction pipeline computes depth maps from input images:

- **Precalibration.** We extract information from the EXIF image meta data to create an initial calibration matrix K_{pre} for each camera. We automatically search the Web for missing information. See Section 3 for more details.

- **Epipolar Geometry Computation.** SURF features [14] are extracted and matched between all image pairs. For pairs where both cameras have an estimated K_{pre}, we compute an essential matrix using the 5-point algorithm [15] in a RANSAC scheme. For pairs without K_{pre}, we compute a Fundamental matrix using the 7-point algorithm [9].
- **Optional step: Self calibration.** If there are less than two images with K_{pre}, we employ the self calibration method described in [3]. This method breaks down if K_{pre} varies throughout the image sequence or if the motion and/or structure are not general enough.
- **Sparse Reconstruction.** This first selects an initial view pair. To ensure a large enough baseline, we examine whether they cannot be matched too well with the infinite homography [8]. The scene is reconstructed by adding views sequentially. For each new view, we resection its camera using the 3-point algorithm [16] and triangulate all matches with already reconstructed views. A bundle adjustment is then performed to minimise the reprojection errors. We optimise for the six pose parameters, focal length and two radial distortion coefficients.
- **Dense Reconstruction.** The sparse reconstruction is used to initialise a dense reconstruction with 3D coordinates for each pixel, and not just for SURF feature points. Using the computed cameras, we order overlapping images into chains, rectify them and employ a dense stereo algorithm based on dynamic programming [17]. The stereo pairs are then linked together in their chain, and the optimal depth for each pixel is tracked using a Kalman filter. The result is a dense depth map for each view, where each pixel tells us its distance from the camera. In addition, the method also produces quality maps, recording in how many views each pixel was successfully matched. The depth maps are used as input to the mesh generation step which is detailed in Section 5.

3 Camera Parameter Retrieval

Most recent Structure from Motion pipelines use EXIF meta data for camera pre-calibration [5,4,6]. The EXIF data typically contains the focal length f_e of the camera lens, given in millimetres. This can be used to obtain an estimate of the focal length in pixels: $\hat{f} = \frac{f_e w_i}{w_s}$.

Here, w_i and w_s are the widths of the image (in pixels) and camera sensor (in millimetres), resp. Assuming zero skew, the principal point at the origin and unity aspect ratio (square pixels) gives us a pre-calibration matrix of the following form:

$$K_{pre} = \begin{bmatrix} \hat{f} & 0 & \frac{w_i}{2} \\ 0 & \hat{f} & \frac{h_i}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

3.1 Obtaining the Sensor Width

While almost every camera stores f_e in the EXIF tag, many low-end consumer cameras do not store w_s. However, the information can usually be found in the technical specifications of the camera. To this end, we maintain a database of sensor widths for different camera models, indexed by the EXIF model name. Whenever an unknown camera model is encountered, our software automatically searches the web for w_s. If the camera model is not very obscure, there is likely to be a review or product description web page that contains the required information. In practice, we perform the following steps:

- **Google search.** The *model* and *make* tags of the EXIF data are combined to generate a Google search query. To better focus the search, we append the words *camera*, *specification*, *specs* and *sensor* to the query. We also add the string *dpreview* to favour web pages from the Digital Photography Website[1], which contains many camera specifications.
- **Parse for sensor sizes.** The first five resulting HTML documents from Google are parsed for candidate sensor sizes. Sensor sizes can be given in two different formats: *width* x *height* in *mm* or as an imperial fraction, such as $1/1.8''$. We use a regular expression matcher to obtain all potential sensor sizes in the document.
- **Reject invalid dimensions.** The imperial fractions do not measure the diagonal of the sensor, but rather the diagonal of an imagined Vidicon tube [18] centred on the sensor. Since these fractions are well defined, we use a look-up table to convert them to *mm* of width and height. A fraction that is not found in the look-up table is discarded. We also discard any measurements outside the interval $[2.0, 40.0]$ *mm* and we limit ourselves to dimensions with aspect ratio close to 4:3. Camcorders and industrial cameras may have a different aspect ratio than 4:3, but such cameras typically do not produce EXIF data and would not be usable nevertheless. If there is still more than one sensor size left, we select the one closest to the word *sensor* in the document.

3.2 Retrieval Results

Uploaded Content. We first evaluated the retrieval system using data uploaded by users during the course of two months. Out of 23 950 uploaded images, we first removed 9 788 images which were lacking f_e. Note that ARC3D is still able to process such images using the alternative calibration method. Out of the remaining images, 5 505 already had w_s in the EXIF data and were thus removed from the experiment. Finally, 170 of the remaining images lacked a model tag and were also discarded. This left us with 8 487 images that were used as input to the algorithm. These images were found to have been taken with 114 different camera models.

We evaluated the performance using two methods: per camera and per image. In the per-camera evaluation we simply search for each of the 114 camera models. In the per-image evaluation we weight the per-camera result with the number of images taken with each camera. This is to reflect the fact that some camera

[1] http://www.dpreview.com

models are more probable than others. We use the term *retrieved* for camera models that returned any value and *correct* when the returned value was correct (verified manually). The results were as follows:

	Nr input	**Retrieved**	**Correct**	**Precision**	**Recall**
Per camera	114	76	73	96%	64%
Per image	8487	7622	7450	98%	88%

The precision is of high importance, as a false positive might severely harm the reconstruction algorithm. We achieve a high value both for the per-camera and for the per-image evaluation. The recall value is seen to be remarkably higher in the per-image evaluation result. This reflects the fact that most people use well-known cameras with a review on the Web. Failure cases include cellphone cameras (for which the camera specs are typically not listed on-line), old or unfamiliar camera models, and Google responses leading to web pages where several cameras are compared.

In summary, we can expect around 88% chance for any randomly chosen image that has f_e but a missing w_s to retrieve a correct sensor dimension. This is indeed a great improvement for reconstruction systems relying on EXIF data for pre-calibration.

Bundler Package. We also evaluated the tool using the camera list included in version 0.4 of the Bundler software package[2]. The list contains 268 different camera models with sensor widths. For each camera, we retrieved a value from the web, w_w, and compared it to the corresponding value from Bundler, w_b. We computed the relative error between the two as $\epsilon = (|w_w - w_b|)/w_b$, and considered the measurement to be correct if $\epsilon < 5\%$. This margin was accepted since the conversion from imperial fractions to millimetres is non-standard and may produce slightly different results. Out of the 268 models, we managed to retrieve a sensor width for 249 cameras. 208 of them were within the error margin. This gave us the following result: **Precision:** 84%, **Recall:**78%. These rates are inferior to the results that we obtained. However, most cameras on the Bundler list date back to 2005 and earlier. The quality of the Bundler list can also be questioned, as we have found several of its values to be incorrect.

4 View Selection for Mesh Generation

Reconstructed depth maps typically have large overlaps since each scene point has to be visible in several images for 3D reconstruction. On the other hand, too much overlap causes unnecessary computational load for a mesh generation algorithm. Moreover, low-quality images (e.g. with motion blur) may produce low-quality depth maps that harm the mesh generation process. We therefore employ a view selection algorithm that from a set of views, V_{in}, selects the best suited views for mesh generation, V_{out}. Recent publications on view selection for

[2] `http://phototour.cs.washington.edu/bundler/`

3D reconstruction [19,20] are concerned with the selection of views *before* dense reconstruction and not *after* as in our case. Here, we are not interested to the same degree in the overlap between images.

We identify three criteria for our algorithm: **(1)** The views should cover the whole scene with as little overlap as possible, **(2)** we want to prioritise depth maps of high quality and **(3)** the number of views for re-meshing should be limited. While **(3)** is simply an upper limit of m such that $|V_{out}| \leq m$, the other criteria require some more attention.

(1) Coverage. We add a view to V_{out} only if it does not overlap too much with views already added. Scene elements common between views are used to determine the degree of overlap. Rather than using the dense depth maps, we avoid heavy computations by using the sparsely reconstructed 3D points to determine scene overlap. As in [20], we argue that the coverage of the sparse reconstruction is approximately the same as for the dense one. We also use their definition of a covered 3D point; a point is covered if it is seen in one or more selected views. To quantify the amount of overlap between views, we could count the number of uncovered points in each new view. However, using this number directly would bias towards views with a larger number of 3D points. This is undesirable, as the sparse 3D points are generally not evenly distributed and areas of high point densities would thus be favoured. Instead, we compute the *coverage ratio* as the ratio between the uncovered and the total number of points seen in a view. The coverage ratio ranges between 0 when no new scene content is present in the view and 1 when all points are uncovered, irrespective of the absolute number of 3D points.

(2) Quality measure. Our multiview stereo algorithm generates a quality map for each depth map. How many times each pixel was seen in another view serves as quality measure. We use the average value as quality measure q_i of the entire depth map.

4.1 Algorithm

Denoting the set of sparsely reconstructed 3D points as M, and $M_V \subset M$ as the 3D points seen in view V, we iterate the following steps:

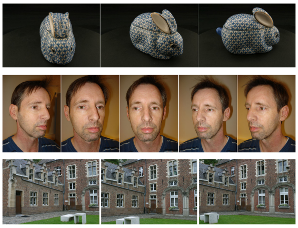


Fig. 1. Example input images for bunny, face and corner sequences

Table 1. Results for view selection algorithm

		Cams	**Faces**	**Time**
Bunny	V_{in}	88	616k	45m
	V_{out}	22	607k	15m
Face	V_{in}	18	475k	9.7m
	V_{out}	6	457k	2.7m
Corner	V_{in}	8	399k	4.2m
	V_{out}	2	380k	1.8m

1. From all views we have not already checked or added, select the one with the highest quality. V_c is the set of checked views.

$$\hat{V} = \max_q (V_{in} \backslash [V_c \cup V_{out}])$$

2. Compute the coverage ratio:

$$c = \frac{|M_{\hat{V}} \backslash M_{V_{out}}|}{|M_{\hat{V}}|}$$

3. Add the view if the uncovered part was large enough:

$$V_{out} = \begin{cases} V_{out} \cup \hat{V}, & c \geq \tau \\ V_{out}, & c < \tau \end{cases}$$

4. Keep track of the views we have checked:

$$V_c = V_c \cup \hat{V}$$

5. If $V_c = V_{in}$ all views are checked. In that case we reduce τ, set $V_c = \emptyset$ and start over, unless one of the stop criteria is satisfied.

Stop Criteria. The iteration stops if any of the following criteria is fulfilled:

1. Max allowed views reached: $|V_{out}| = m$
2. All views added: $V_{out} = V_{in}$
3. All 3D points covered: $M_{V_{out}} = M$

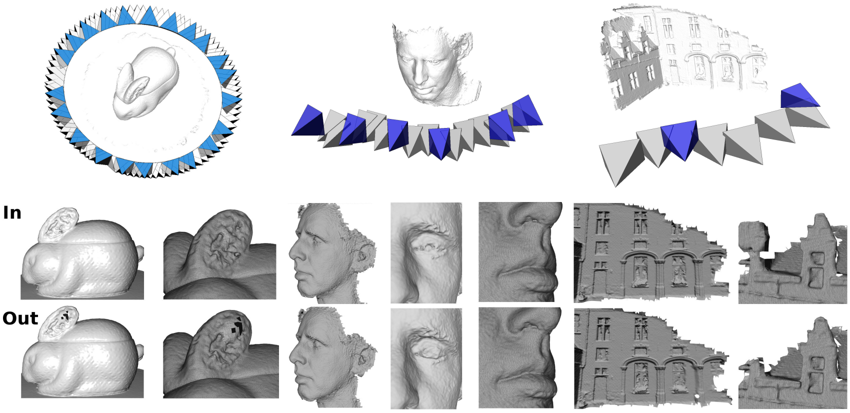


Fig. 2. Results of view selection algorithm. Top: Selected cameras are highlighted. Bottom: Mesh details for V_{in} and V_{out}.

4.2 Results

We evaluated the view selection algorithm on three image sequences, Fig. 1. For each set, we computed two 3D models, one out of all views (V_{in}) and one out of the selected views (V_{out}). We compared the performance in terms of computational speed and model completeness, measured as the number of faces in the resulting mesh. It should be mentioned that the view selection algorithm itself is very fast and its computation time is negligible here.

For all experiments, $m = 30$ and $\tau = 0.7$ with a decrement of 0.1. The numerical results are presented in Table 1. The resulting camera poses as well as comparative views of the resulting meshes, with and without applying view selection, are shown in Fig. 2. The visual difference is negligible for most parts of the meshes. Discrepancies tend to occur in poorly reconstructed areas, such as in the ears of the bunny and in the eye of the face. The eye is actually slightly better reconstructed from V_{out}, which is likely due to the emphasis on the quality measure. In the corner sequence, areas on the edge of the model which are not covered well by the sparse reconstruction are lost. The marginal loss of geometry is however well motivated by the large improvement in computational speed.

5 Mesh Generation

This section describes the creation of a textured mesh out of a scene's depth maps. Given a set of suitable, selected views (Section 4), we clean the depth maps and create a mesh using a state-of-the-art remeshing algorithm. Finally, we use the input images to texturise the model and create a WebGL-ready version by downsampling the mesh.

5.1 Depth Maps to Model

The depth maps belonging to V_{out} are combined into a complete 3D model by applying the following steps, illustrated in Figure 3:

- **Filtering of depth maps.** Since it is more efficient to work with 2D images than with 3D meshes, we perform as much of the filtering as possible in image space. By using the corresponding quality map, we first remove all pixels in the depth map not seen in at least three views. We then find the largest connected component and remove all pixels not belonging to the hull of this component. Finally, an erosion is performed to get rid of possibly noisy border pixels.
- **Depth map to 3D.** Each depth map is back projected into a 3D range grid. A rough mesh is obtained by connecting neighbouring vertices in the grid. We measure the angle between each triangle normal and the optical axis of the camera and remove all triangles where the angle is greater than 80°. We also remove sliver triangles, where the length of the smallest side is less than 2 times the length of the largest. As this operation often creates small floating pieces, we remove all connected components having a face count of less than 10% of the largest component.

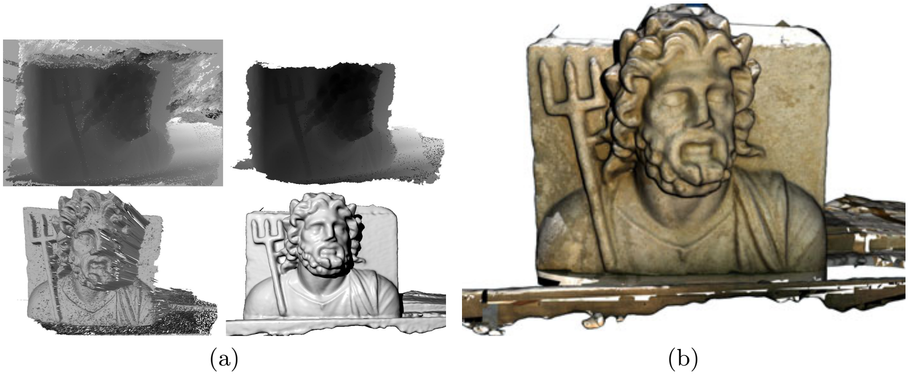


Fig. 3. (a) Creating a mesh out of depth maps. Top left: Original noisy depth map. Top right: The depth map is cleaned and back projected onto a range grid (Bottom left). All range grids are combined and remeshed (bottom right). (b) The mesh is textured to produce the final result. $V_{in} = 18$, $V_{out} = 4$.

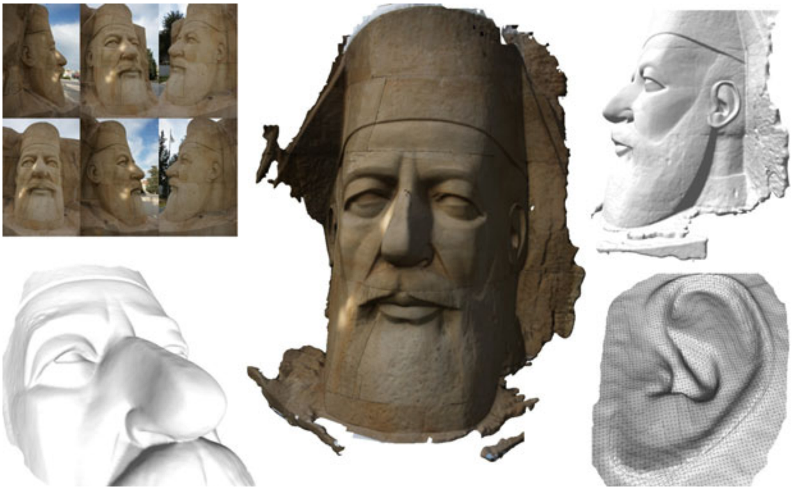


Fig. 4. Resulting model reconstructed from a DSLR camera. $V_{in} = 21$, $V_{out} = 6$.

- **Poisson reconstruction.** The vertices from all depth maps are merged and used as input to the Poisson reconstruction algorithm [21]. This method casts the remeshing into a spatial Poisson problem and produces a watertight, triangulated surface. The Poisson reconstruction algorithm assumes a complete and closed surface, and covers areas of low point density with large triangles. To get rid of these "invented" surfaces we remove all triangles with a side larger than five times the mean value of the mesh.

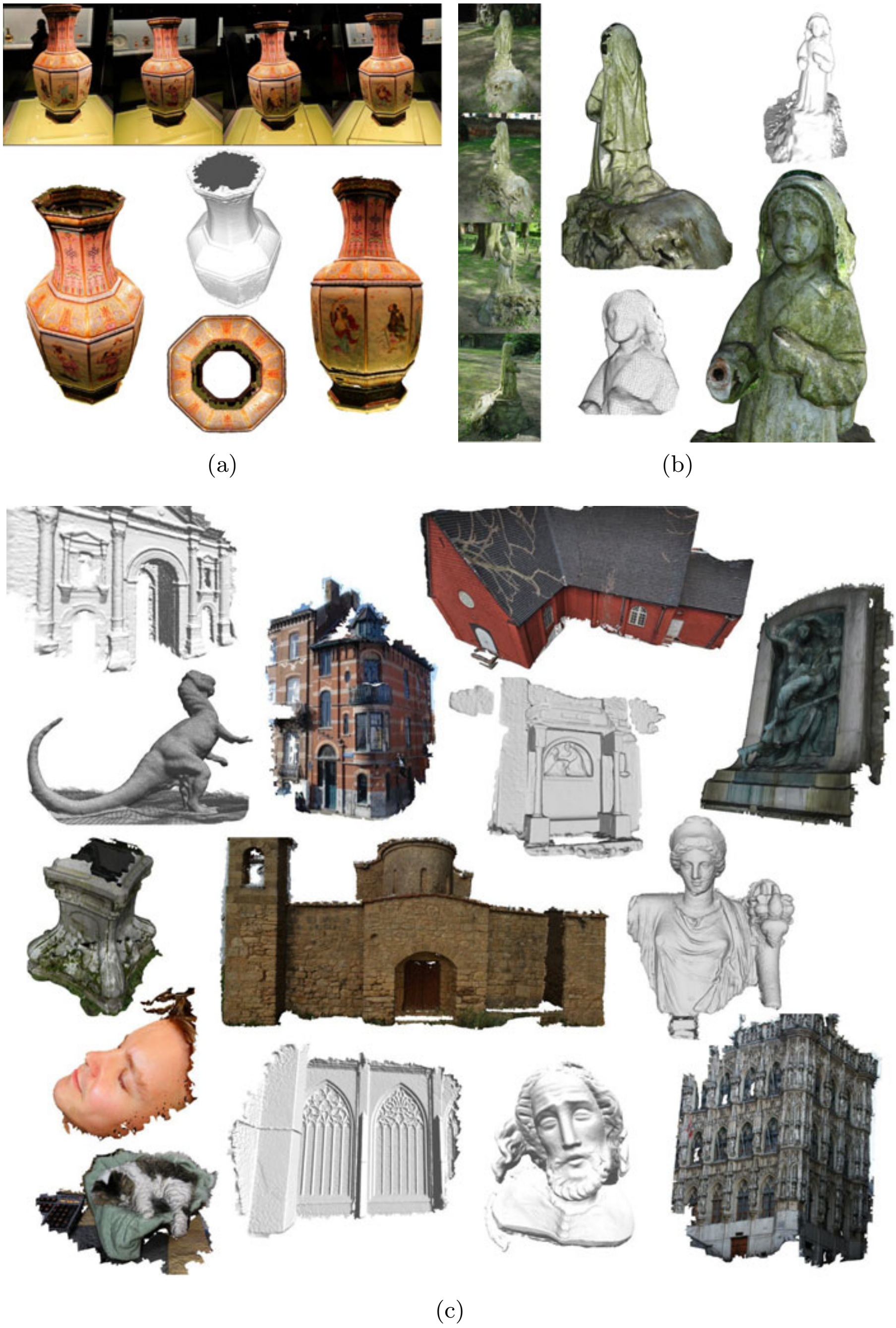


Fig. 5. Reconstruction results. (a) Vase, $V_{in} = 63$, $V_{out} = 18$. (b) Statue, $V_{in} = 26$, $V_{out} = 6$. (c) Further reconstruction results.

- **Applying texture.** We compute texture coordinates for each face by determining in which cameras the face is seen. To handle occlusions, we examine if the rays between the camera centre and face vertices pass through any other face of the model. Using an octree data structure and employing line and frustum culling makes this computationally feasible. If the face is seen in more than one view, we project the triangle into each image and select the one with the largest projection area. This favours fronto-parallel, high resolution and close-up images.
- **Downscaling of the model.** The remeshed model is large and may contain hundreds of thousands of faces. While such a high detail is desirable in some contexts, the amount of data makes the mesh unsuitable for on-line viewing. To this end, we apply a subdivision algorithm [22] to create a low-resolution version of the mesh. Since the texture remains at full resolution, the viewing experience is not seriously degraded. We set the maximum number of faces to 10 000 which results in an uncompressed size of around 1MB.

5.2 Results

All models shown in this paper are the direct output of our reconstruction pipeline and have not been modified in any way. The input images have been taken with the purpose of creating 3D content in terms of their coverage, but can be offered in random order.

Figs 3 and 4 display models obtained with semi-professional DSLR cameras under good lighting conditions. Both geometry and texture are of good quality. Fig. 5 shows results obtained from consumer cameras. The vase in (a) was captured without a tripod in a dimly lit museum. The glass casing around the vase made it impossible to use the flash. Nevertheless, the resulting model is well reconstructed. The same can be said about the statue in (b), which was captured under daylight conditions. In (c) we show further models produced by the system.

6 Conclusion

We have presented an automatic system for the creation of textured meshes out of images taken with a digital camera. Both the initial and final part of a traditional 3D reconstruction pipeline have been enhanced, with functions for camera parameter retrieval and mesh generation, resp. Result are delivered both as an archive with the full resolution model, and as a link that opens the low resolution version in a WebGL model viewer. We also provide the original, raw depth maps and the recovered camera parameters.

Our system is currently on-line at `http://www.arc3d.be` and open to the public, completely free for non-commercial purposes.

Acknowledgements. We gratefully acknowledge support from the EC 7th Framework Program under grant agreement n. 231809 (IP project 3D-COFORM) and the KULeuven Research Council grant Center for Archeological Studies. We thank Dr. Sven Havemann of TU Graz, Dr. Sorin Hermon of the Cyprus Inst., and Paul Konijn from KULeuven, for providing us with images sets.

References

1. http://www.khronos.org/webgl (2011)
2. Sons, K., Klein, F., Rubinstein, D., Byelozyorov, S., Slusallek, P.: XML3D: interactive 3D graphics for the web. In: Web3D (2010)
3. Vergauwen, M., Van Gool, L.: Web-based 3D reconstruction service. MVA 17, 411–426 (2006)
4. http://www.photosynth.net (2011)
5. http://www.3dtubeme.com/ (2011)
6. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH (2006)
7. Furukawa, Y., Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis. In: CVPR (2007)
8. Moons, T., Van Gool, L., Vergauwen, M.: 3D reconstruction from multiple images: Part 1 - principles. Found. and Trends in Comp. Graph. and Vis. 4, 287–404 (2009)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
10. Pollefeys, M., Verbiest, F., Van Gool, L.: Surviving dominant planes in uncalibrated structure and motion recovery. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 837–851. Springer, Heidelberg (2002)
11. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV (2009)
12. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building rome on a cloudless day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 368–381. Springer, Heidelberg (2010)
13. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In: CVPR (2008)
14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features. CVIU 110, 346–359 (2008)
15. Nistér, D.: An efficient solution to the five-point relative pose problem. TPAMI 26, 756–777 (2004)
16. Haralick, B.M., Lee, C.N., Ottenberg, K., Nlle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. IJCV 13, 331–356 (1994)
17. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual Modeling with a Hand-Held Camera. IJCV 59, 207–232 (2004)
18. Tozer, E.P.J.: Broadcast engineer's reference book, 13th edn. Elsevier; Focal Press, Boston, Massachusetts (2004)
19. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-View Stereo for Community Photo Collections. In: ICCV (2007)
20. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multiview stereo. In: CVPR (2010)
21. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: SGP (2006)
22. Hoppe, H.: Progressive Meshes. In: SIGGRAPH (1996)

A Novel Approach to Image Assessment by Seeking Unification of Subjective and Objective Criteria Based on Supervised Learning

Pipei Huang, Shiyin Qin, and Donghuan Lu

School of Automation Science and Electrical Engineering,
Beihang University, Beijing, China
huangpipei@gmail.com, qsy@buaa.edu.cn, ludonghuan9@sina.com

Abstract. Image quality assessment is a challenge research topic in imaging engineering and applications, especially in the case where the reference image cannot be accessed, such as aerial images. In view of such an issue, a novel learning based evaluation approach was developed. In practice, only objective quality criteria usually cannot achieve desired result. Based on the analysis of multiple objective quality assessment criteria, a boosting algorithm with supervised learning, LassBoost (Learn to Assess with Boosting), was employed to seek the unification of the multiple objective criteria with subjective criteria. This new approach can effectively fuse multiple objective quality criteria guided by the subjective quality level such that the subjective /objective criteria can be unified using weighted regression method. The experimental results illustrate that the proposed method can achieve significantly better performance for image quality assessment, thus can provide a powerful decision support in imaging engineering and practical applications.

Keywords: supervised learning, image quality evaluator, unification of subjective and objective assessment criteria, Boosting.

1 Introduction

Image quality assessment is a challenging topic in image engineering and application. PSNR is the most typical measurement for image assessment when the reference image is provided. However, only a few methods can serve to assess without the reference image, such as aerial image evaluation. Moreover, most of those methods are only capable of working for the image corrupted by single factor and cannot yet exceptionally deal with the assessment under the complicated background without reference image. In order to address this problem, there are several difficulties should be overcome. First of all, there are no unified criteria for the image assessment without reference image. Secondly, it requires too much human efforts to assessing with subjective criteria. And the image corrupted by multi-factor cannot be assessed by single criteria, and so on.

Recently, the research of image quality evaluation mainly includes two aspects, subjective criteria and objective criteria. NIIRS (National Imagery Interpretability

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2011, LNCS 6930, pp. 274–285, 2011.

Ratings Scale) is a widely-used quantitative subjective criterion[1][2], which specifically considers the relationship between user requirements and the quality of remote images. Moreover, the objective evaluation usually considers image resolution as the metric for image quality assessment[3], in which image sharpness plays an important role and is often measured by gradient functions. Gradient depicts the variation of gray values between different pixel locations. We can admit that the sharpness of image is proportional to the value of gradient. In general, gradient can be quantitatively calculated by the method of edge detection. Thus, there are several assessment methods using image sharpness to build the gradient functions[4], such as energy gradient function, Tenengrad function, Brenner function, variance function, etc. The image noise can be viewed as a high frequency signal which is of larger gradient value, therefore, it is a tradeoff between anti-noise capability and image quality evaluation. Besides, based on the Shannon sampling theorem—the greater entropy is, the more information it comprises. As a result, the entropy[5] can be also considered as a metric when the mean of image gray values is a constant. However, the value of entropy doesn't reflect the detail information such as image edges, hence, it also can't be viewed as the only criteria of evaluating the image under the complicated background. SNR(Signal to Noise Ratio) is widely used criteria for image quality evaluation, which doesn't work without the reference image, since it is too difficult to separate noise from the corrupted image. J.M. Delvit and D.Leger et al.[6] presented a new approach which applies artificial neural network to estimate the value of SNR. First of all, training dataset is built based on the known noise model. Subsequently, a mapping function is learnt between image features and the corresponding SNR. Finally, the learned mapping model can be applied for precisely predicting the SNR of testing image. Inspired by this method, we developed a novel approach to explore the problem of image assessment based on boosting methodology.

Furthermore, the boosting algorithm is widely used in the supervised learning and semi-supervised learning [7][8], which predicts samples using a weighted vote over a set of weak classifiers. Freund and Schapire[9] proposed the AdaBoost algorithm, which is the most typical version of boosting algorithm. More importantly, Friedman et al.[10] imported an additive logistic regression model into AdaBoost and exactly explained the boosting algorithm from a statistical view. From the perspectives of additive regression model and exponential loss function, Friedman et al. proposed some new boosting algorithms including "GentleBoost" and "LogitBoost" which are widely used in text classification and image recognition. Torralba et al.[11] proposed the algorithm "JointBoost" for multi-class classification in computer vision. Based on this work, Pipei Huang et al.[12] proposed a novel algorithm for solving multi-task learning problem and which is capable of working for the high dimensional feature space. Recently, Balazs Kegl et al.[13][14] constructed the products of base classifiers as a new hypothesis and presented a novel boosting approach.

In order to deal with the image quality assessment without reference image, especially for which are influenced by multiple degradation factors, there are several issues should be overcome: Firstly, single objective criteria only serves to evaluate the images degraded by one kind of factors and reflects a particular characteristic of image quality, which lead to be sensitive to noise and worse assessment results the testing image is corrupted by multiple degradation factors. Moreover, desirable unified criteria of image quality assessment without reference usually cannot be determined, whereas the subjective knowledge from experts is crucial to the assessment. However, it requires too much human resources to implement in each corrupted image using subjective method. In order to synchronously resolve the two issues mentioned above, we present a new approach, LassBoost, to make unification between the subjective and objective criteria. At first, we select out multiple typically objective assessment criteria. Subsequently, make use of LassBoost to learn the weights of objective criteria under the expert supervision. Moreover, the approach would fuses all the objective criteria using weighted regression such that the subjective/objective criteria can be effectively unified. Besides, the approach also can update the feature space of objective criteria and relearn the weights for the other complicated assessment task.

The rest of this paper is organized as follows. Section 2 analyzes the existing limitations in single objective criteria. Section 3 conducts the theory of image quality assessment in the paper, describes the possible mapping between feature space and subjective quality levels which are built by experts, as well as the feasibility of fitting this mapping with supervised learning technology. In Section 4, LassBoost is proposed such that the resulting image quality evaluator can be designed. Section 5 illustrates the experimental results and the final section concludes the paper.

2 Characteristic and Limitation of Single Objective Assessment Criteria

There are two typical problems in image quality assessment. The first problem is assessment with reference image, which is required to use the criterion, such as PSNR, to measure the “distance” between degraded image and reference image. The second one is the quality assessment method without reference image. Objective criteria could be work in the case where the image is degraded by single factor. However, no single objective criteria can achieve good performance on the testing image which is degraded by multiple factors, such as Gaussian noise, salt and pepper noise, motion blur, defocus blur, etc. Therefore, we should import subjective knowledge from experts into the image assessment problem.

In general, the classical objective criteria of image quality assessment can be classified into three categories[4][5]. The first category is objective criteria based on the definition of gradient (such as energy gradient function, TenenGrad function, Brenner function, point sharpness function and so on), and which depend on the prerequisite that the image gradient is proportional to the sharpness of image textures and edges. The assessment criteria take use of the gradient values to evaluate image edge and specific details, apparently, cannot achieve good performance, when the densities of image gradient are destroyed by a large amount of noise drawn from different distribution models. Moreover, the second criteria are derived by the order degree of image information which is determined by the probability distribution of gray values, such as entropy. These criteria are of promising anti-noise capability, since the random noises from Gaussian model reduce the degree of order, and which would lead to entropy increase. Whereas the metric abilities of the second criteria are worse than those derived from image gradient. The third assessment criteria are derived from the image variance, such as MSE and improved PSNR, and which are dependent on the assumption that variance is proportional to the sharpness values of image texture and edge. So the criterion is actually a tradeoff between the sensitiveness of noise and the metric capability of image edge details. In addition, weighting algorithm based on multiple objective criteria seems to absorb the advantages of several criteria, such as the abilities including anti-noise, sharpness measurement. In fact, the discrepancy of the images corrupted by different factors is very large, so the weight coefficients of the criteria cannot be easily determined and updated only if the expert information in not taken into account. As a result, the traditional weighting methods would have plenty of limitations in the complicated assessment issue. In fact, the specific multi-objective assessment criteria are selected from the three categories above. And the existing characteristics in all the categories are covered by the selected criteria as many as possible. However, because of the limitation of pages, the detail is not listed.

In summary, each objective criterion has its own advantages, such as anti-noise, the metric capability of edge details, but there are still of much limitation whichever you uses. As a result, there is no one objective criteria which can exceptionally handle the images assessment problem without reference image, especially in that case where the images are corrupted by multiple factors or under the situation without reference image.

3 The Mapping between Feature Space of Multiple Objective Criteria and Experts' Subjective Assessment Criteria

Due to the limitations of using single objective criteria as method for image assessing, we can reasonably imagine that experts' evaluation results are the golden standard. Hence, there should be a certain mapping relation between multiple objective criteria and the subjective assessment criteria from experts, whereas this relation cannot be specifically obtained by mathematical formula. However, it can be solved by learning a mapping model with supervised learning technology, and consequently derive a new approach for image assessment, where both subjective knowledge and objective criteria will be taken into account.

As a result, we propose a novel approach to explore the image quality assessment problem with machine learning technology. The crucial steps of our approach are shown as Figure 1. First of all, we calculate the features with the objective assessment criteria on each training image. Subsequently, the image label evaluated by experts is constructed and considered as absolutely correct quality level. Thirdly, we use the LassBoost algorithm to learn a generalized model through which the specific mapping between the features and labels would be built. Finally, the weights learnt from the model will apply to predict the quality level of testing data.

Based on the limitation analysis concerning objective criteria, the experiments in the paper choose nine classical objective criteria to construct the feature space. The space is able to expand to high dimensions based on different task requirements.

Furthermore, the images in the dataset are evaluated to ten levels by several experts and each level is represented by one particular score. All the score (referred to as label) used for accessing the average should be evaluated from different experts and the average value would be viewed as the final subjective assessment result.

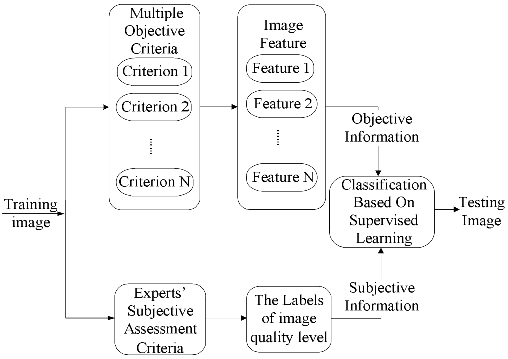


Fig. 1. Unification of the multiple objective criteria and subjective assessment criteria

4 Learning Algorithm for Constructing Evaluator System

In the multi-class classification problems, we have C learning categories, and assume that all the samples come from the same space $X \times Z$. For simplicity, we assume that $X \subset R^F$ and $Z \subset R$, where F indicates the number of feature dimension as well as the number of objective criteria of the paper. The available data from different categories is denoted as $\{(x_i, z_i)\}_{i=1}^N \subset X \times Z$. Each experiment is considered as a multi-class classification problem, $z_i \in \{1, 2, ..., C\}$.

In order to evaluate the experts' subjective assessment quantitatively and precisely, the images in dataset divide the score into 10 levels based on their qualities, thus the quality assessment issue is specified and reformulated to a multiple classes classification problem. Our algorithm in the paper is expanded from the framework of GentleBoost[12][15], which is a classical boosting classifier in binary classification. Before we conduct the new algorithm, we need to make a brief explanation on GentleBoost algorithm. The loss function of GentleBoost is defined as follows:

$$J = E(e^{-yH(x)}) \tag{1}$$

Where x and y denote an input training sample and an output label $y \in \{-1, +1\}$. $H(x)$ represents the additive model such as $H(x) = \sum_{t=1}^T h_t(x)$, where $h_t(.)$ and t denote a weak classifier and the iterations number respectively.

Through the second order Taylor approximation, the optimization problem in GentleBoost becomes:

$$\arg\min_{h_t} E[e^{-y(H(x)+h_t(x))}] \simeq E[e^{-yH(x)}(y - h_t(x))^2] \tag{2}$$

where $h_t(.)$ is chosen to minimize the new loss.

If $w = e^{-yH(x)} \forall i$ is defined as the weight of each sample, optimizing (2) is equal to minimize the weighted squared error as follows:

$$J_{wse} = \sum_{i=1}^{N} w_i (y_i - h_t(x_i))^2 \tag{3}$$

where N is the number of training samples. If the weak classifier is the decision stump, i.e.,

$$h_t(x_i) = a\delta(x_i^f > \theta) + b\delta(x_i^f \le \theta) \tag{4}$$

where x_i^f denotes the f th feature of the sample x_i, θ is threshold, $\delta(.)$ is the indicator function, a and b are parameters. Setting $\frac{\partial J_{wse}}{\partial a}$=0, $\frac{\partial J_{wse}}{\partial \mathrm{b}} = 0$, we have:

$$a = \frac{\sum_{i=1}^{N} w_i y_i \delta(x_i^f > \theta)}{\sum_{i=1}^{N} w_i \delta(x_i^f > \theta)} \tag{5}$$

$$b = \frac{\sum_{i=1}^{N} w_i y_i \delta(x_i^f \le \theta)}{\sum_{i=1}^{N} w_i \delta(x_i^f \le \theta)} \tag{6}$$

In each iteration of GentleBoost, the algorithm chooses the feature with which the weak classifier mostly minimizes the exponential loss function. This feature is referred to as the predictive feature.

In order to implement assessing in different scenarios, we extend the GentleBoost algorithm to multiple classes with the method of one-against-all. The new algorithm referred to as LassBoost is illustrated as follows:

Algorithm 1. LassBoost: Learn to assess based on boosting methodology

```
Input: $(x_1, z_1), ..., (x_i, z_i), i = 1...N, 1 < z_i \le C$
Output: Ensemble classifier $H : X \to Z$
Initialize: Set the weights $w_i = 1, 1 \le i \le N$
For c=1 to C Do
IF $z_i = c$ Then
$z_i^c = +1$
Else
$z_i^c = -1$
End IF
For t=1 to T Do
For f=1 to F Do
Calculate $(a, b)$ to construct the weak classifiers:
        $h_t^c(x\ ) = a\delta(x^f > \theta) + b\delta(x^f \le \theta)$
Use weak classifier to evaluate loss function:
        $J_{wse} = \sum_{i=1}^{N} w_i (z_i^c - h_t^c(x_i))^2$
End For
```

```
        Determine the predictive feature $f^*$ and its related threshold $\theta^*$ :
            $(f^*,\theta^*) = \arg\min_{f,\theta} J_{wse}(x^f,\theta)$
        Update the ensemble classifier:
            $H_t^c(x) := H_{t-1}^c(x) + h_t^c(x^{f^*},\theta^*)$
        Update weights:
            $w_i^c := w_i^c e^{-z_i^c h_t^c(x^{f^*},\theta^*)}$
    End For
End For
```

In the description of Algorithm 1, LassBoost adapt GentleBoost algorithm into a multi-class problem that includes C binary classification elements. Subsequently, LassBoost is required to scan each feature and corresponding threshold to learn the regression parameters (a,b) of classifier as well as build the loss function iteratively. The feature and the corresponding threshold which mostly minimize the loss function are chosen as the predictive feature f^* and the predictive threshold θ^* . The algorithm updates the ensemble classifier in terms of $H_t^c(x) := H_{t-1}^c(x) + h_t^c(x^{f^*},\theta^*)$ and updates the weights to samples according to $w_i^c := w_i^c e^{-z_i^c h_t^c(x^{f^*},\theta^*)}$. The algorithm repeats the procedure above to ensemble the weak learner into the final classifier.

At the prediction step, LassBoost makes the final decision over each sample with the ensemble classifier which is combined by C components. More specifically, the final predictive result of each sample is determined by the highest confidence that is output by the ensemble learner. For the weak classifier, $h_t^c(x^f\ ,\theta\ )$, the output of regression stump is the value of regression parameter decided by the threshed θ of the f th feature, a or b. Finally, the algorithm combines the output of each weak learners in terms of $H_T^c(x) = \sum_{t=1}^T h_t^c(x^{f^*},\theta^*)$. The algorithm iterates the procedure described above and learns C category classifiers with which each of the testing samples would be evaluated with C confidence values. The category corresponding to the highest confidence score is the finally predictive assessment result.

5 Experiments

5.1 Dataset

We build the dataset with the image of "Lena" to evaluate the LassBoost algorithm. There are three category images which are respectively corrupted by noise, blur or mixture. The noises used in the experiment include Gaussian, salt and pepper, and mixture noise, while the blur type would cover motion blur, defocus blur as well as mixture one. Table 1 illustrates fifteen corruptions used in our experiment.

Table 1. Categories of the image corruption and corresponding parameter

Image category	Corruptions	Parameter values
Image corrupted by noise	Gaussian noise	Gaussian noise with average 0 and variance 0.01-0.03 Salt and pepper noise with density 0.02-0.06
	Salt and pepper noise	
	Mixture noise (Gaussian + salt and pepper)	
Image corrupted by blur	Motion blur	Motion blur with scale 10-30 and angle 20 Defocus blur with scale 5-15
	Defocus blur	
	Mixture blur(motion + blur)	
Image corrupted by noise and blur	Gaussian noise + motion blur	Gaussian noise with variance 0.01 , Salt and pepper noise with density 0.02; Motion blur with scale10-30 and angle 20, Defocus blur with scale 5-15
	Gaussian noise + defocus blur	
	Gaussian noise + mixture blur	
	Salt and pepper noise + motion blur	
	Salt and pepper noise + defocus blur	
	Salt and pepper noise + mixture blur	
	Mixture noise + motion blur	
	Mixture noise + defocus blur	
	Mixture noise + mixture blur	

The experimental formulation is shown as Table 1 where each group has 200 images which are produced by the model with different parameters. Nine dimensional features calculated by objective criteria are applied and 200 images of each group are evaluated into 10 levels scored by experts, in terms of the scale of corruption parameters.

5.2 Experimental Results and Analysis

There are several classification methods used in the experiments which include LassBoost, SVM and logistic regression. The LassBoost algorithm is a multi-class nonlinear classifier based on the theory of one-against-all. SVM used in the experiment is a typical nonlinear kernel algorithm offered by libsvm, where the radial basis kernel method is applied. Moreover, linear logistic regression model serves to make a comparison with other nonlinear methods. The five-fold cross validation is applied to each baseline method in the experiments.

Figures 2-6 depict the accuracies and standard deviation results concerning LassBoost, SVM and logistic regression on fifteen different problems. The horizontal axis denotes the images degraded by different corruptions, while the vertical axis denotes the classification accuracies, and the black bar means standard deviations of each classification result. Five-fold cross validation is employed in the experiment where three folds are used for the training, one left fold is used for baseline methods for parameter tuning (such as the regularization parameter and the times of iteration) and the last fold is used in the testing. All the accuracies demonstrated in the figures are the average results of five-fold experiment.

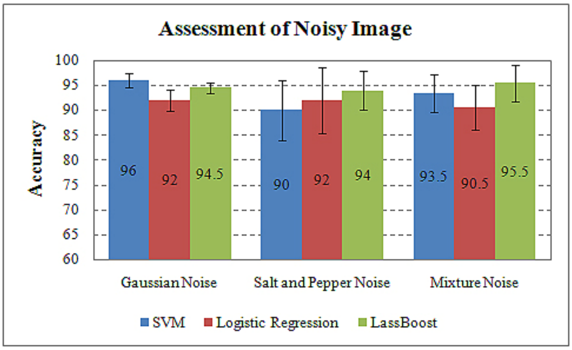


Fig. 2. The evaluation results of images corrupted by noise

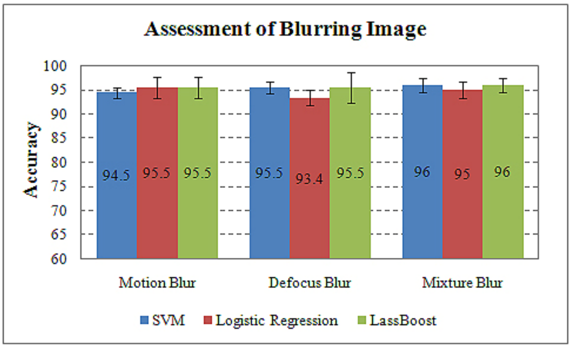


Fig. 3. The evaluation results of images corrupted by blurring

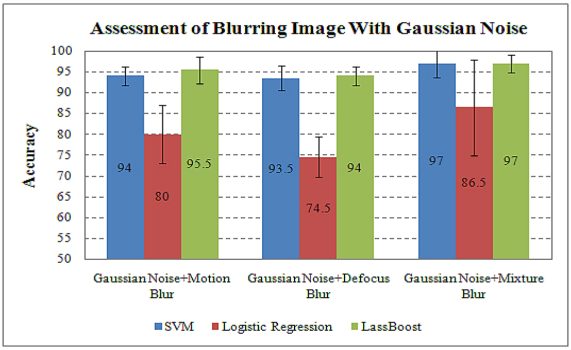


Fig. 4. The evaluation results of images corrupted by Gaussian noise and blurring

In the experiments of Figures 2-3, the data corrupted by single factor obey a linear distribution, hence, both the linear and the nonlinear model can get the high accuracies and the strong stabilities. Since image parameters used for building the dataset vary continuously, the resulting features produced from neighboring images should have high proximity. However, the scoring errors of subjective assessment from experts are inevitable, thus the assessment accuracy is very difficult to achieve up exactly to one hundred percent. Nevertheless, the error influences the results slightly, such that it can be accepted for image quality assessment in practice.

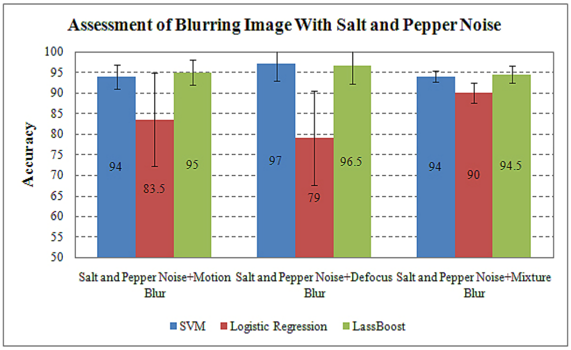


Fig. 5. The evaluation results of images corrupted by salt-pepper noise and blurring

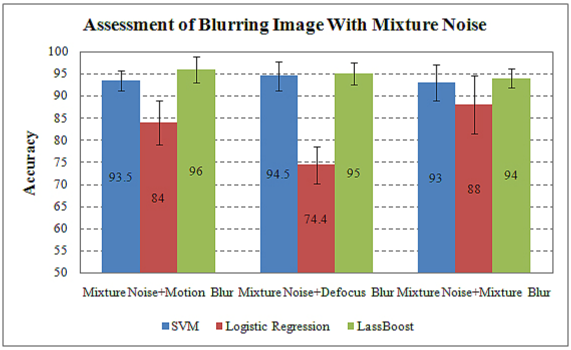


Fig. 6. The evaluation results of images corrupted by mixture noise and blurring

Figures 4-6 show that the three classifiers perform absolutely different generalized abilities in the case of mixture corruptions. The nonlinear classifiers such as LassBoost and SVM get higher accuracies and stronger stabilities than the liner classifier. We can notice that the experimental data distributions will be no longer fitting linear models. Therefore, only nonlinear classifier model can fit the separating hyperplanes corresponding to the nonlinear image data. Furthermore, taking into account the time complexity of kernel SVM algorithm of libsvm, we design the quality evaluator using LassBoost to solve the assessment problem.

In Table 2, the results are selected randomly from the experimental results in five-fold cross validation. The first row is the name of noise and blur and the rows from second to tenth mean the confidence results that the three testing samples are classified into different quality levels, yet the confidence value is normalized to [-1 1]. The results of last two rows make a comparison of final predictive results and real quality levels.

Table 2. The confidence comparison of images which are classified into the wrong category in the mixture corruption experiment

Experiment Name Confidence of Category	Mixture Noise+Motion Blur	Mixture Noise+Defocus Blur	Mixture Noise+Mixture Blur
Category 1	-1.0	-1.0	-0.999
Category 2	0.153	-0.043	0.036
Category 3	0.382	-0.036	0.046
Category 4	0.290	-0.030	0.055
Category 5	0.207	0.028	0.063
Category 6	0.401	0.038	1.0
Category 7	0.404	0.147	0.255
Category 8	1.0	0.354	0.047
Category 9	0.521	1.0	0.035
Category 10	0.295	0.665	-1.0
Predict Level	Level 8	Level 9	Level 6
Real Level	Level 9	Level 10	Level 7

The final predictive result of our algorithm is determined by the level with the highest confidence score. For instance, the sample in second column is classified into level 8 and the corresponding confidence is 1.0, however the ideal prediction would be level 9, thus it leads to a mistake. If the confidence is sorted by descending order, we can notice that the real level ranks the second place, and which is actually yet subject to the principle that the value of confidence is inversely proportional to the distance between predictive result and real category. It is because that the deviations caused by experts' subjective assessment are inevitable, the evaluator learned from these data probably makes a wrong prediction on few particular individuals.

6 Conclusions

In the paper, we present a novel approach for image quality assessment which can unify the subjective and objective criteria with supervised learning technology, and design a learning evaluator system. The evaluator can effectively synchronize multiple objective criteria and objective criteria, which employs the weighted regression method. Moreover, the proposed method is capable of implementing in the large scale problem without reference image. Besides, the feature dimension of the dataset and the corresponding assessment score from expert system can be updated as different application requirements. Finally, the experimental results show that the LassBoost algorithm achieves better and more stable assessment results in different problems.

Acknowledgment. The authors would like to acknowledge Yongfei Zhang and Wenze Zhao for recommending the classic approach of image assessment beneficial discussions. The authors also would like to thank the other members in the Laboratory of Pattern Recognition and Artificial Intelligence, Beihang University. This work was funded by the National Natural Science Foundation of China under Grant 60875072, International Cooperation Project under Grant 2007DFA11530, and Beijing Natural Science Foundation (Intelligent optimizing controls of networked robot group towards advanced processing implementation of food and pharmacy).

References

1. Weihong, S., Shiping, C.: A Remote Sensing Image Quality Standard Orienting to User's Mission Requirements - NIIRS. Spacecraft Recovery & Remote Sensing 24(3), 30–35 (2003)
2. Qiang, Z.: A Research On the Influence of Satellite Population Parameter to Remote Sensing Image Quality. A Dissertation Submitted to Harbin Institute of Technology for the Master's Degree (2002)
3. Riehl, K.: A Historical Review of Reconnaissance Image Evaluation Techniques. SPIE 28(29), 322–333 (1996)
4. Li, Q., Feng, H., et al.: Research On the Assessment Function of Digital Image's Sharpnes. Acta Photonica Sinica 31(6), 736–738 (2002)
5. Xiu, J.: Aerial Image Quality Evaluation Based On Image Power Spectra. A Dissertation Submitted to the Academy of Sciences for the Degree of Ph.D. of Science (2005)
6. Delvit, J.-M., Leger, D., et al.: Signal to noise ratio assessment from non-specific view. SPIE 45(52), 40–44 (2002)
7. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
8. Schapire, R.E.: A brief introduction to boosting. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (1999)
9. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning (1996)
10. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28, 1135–1168 (2000)
11. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 854–869 (2007)
12. Huang, P., Wang, G., Qin, S.: A Novel Learning Approach to Multiple Tasks Based on Boosting Methodology. Pattern Recognition Letters 31(12), 1693–1700 (2010)
13. Kegl, B., Busa-Fekete, R.: Boosting products of base classifiers. In: Proceedings of the 26th International Conference on Machine Learning (2009)
14. Busa-Fekete, R., Kegl, B.: Fast boosting using adversarial bandits. In: Proceedings of the 27th International Conference on Machine Learning (2010)

Author Index