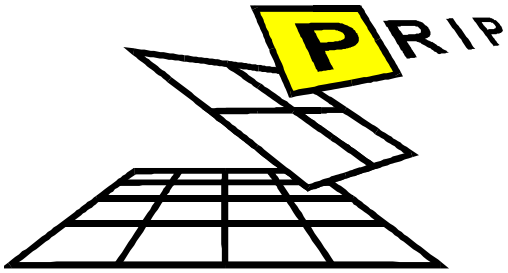
Horst Wildenauer
Walter Kropatsch (Eds.)

Computer Vision – CVWW'02

Proceedings of the Computer Vision Winter Workshop
Bad Aussee, Austria
4–7 February 2002

Pattern
Recognition &
Image
Processing
Group



Institute of
Computer Aided Automation

TECHNISCHE UNIVERSITÄT WIEN

Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/1832
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: sec@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-72 March 26, 2002

Computer Vision Winter Workshop 2002

Horst Wildenauer, Walter Kropatsch (Eds.)

Preface

This technical report collects the papers presented at the *Computer Vision Winter Workshop 2002* (CVWW'02) held 4-7 February 2002, in Bad Aussee, Austria. The workshop was organized by the Pattern Recognition and Image Processing Group, Vienna University of Technology. Each paper submitted to this workshop was reviewed by two members of the Programm Committee, and I would like to take the opportunity to express my sincere appreciation for their services and timely efforts. I would also like to thank Beatrix Forsthuber and Christa Kropatsch for the excellent preparation of the workshop. Finally, my thanks go to all the authors for their contributions.

March 2002

Horst Wildenauer

Computer Vision Winter Workshop 2002

Bad Aussee, Austria
4–7 February 2002

Program Chair
Walter G. Kropatsch
Vienna University of Technology
Institute for Computer Aided Automation
Pattern Recognition and Image Processing Group
Favoritenstrasse 9/183-2
A-1040 Vienna, Austria
Tel: +43 1 58801 18350
Fax: +43 1 58801 18392
E-mail: krw@prip.tuwien.ac.at

Co-Chair
Horst Wildenauer
Vienna University of Technology
Institute for Computer Aided Automation
Pattern Recognition and Image Processing Group
Favoritenstrasse 9/183-2
A-1040 Vienna, Austria
Tel: +43 1 58801 18372
Fax: +43 1 58801 18392
E-mail: wilde@prip.tuwien.ac.at

Contents

Session: 3D

Session: Matching

Session: Segmentation - Filters

Session: Topological Representations

Session: Color

Session: Wide Angle Vision

Session: Applications

Reconstruction from Perspective Images with Occlusions

Daniel Martinec and Tomáš Pajdla*
Center for Machine Perception, Czech Technical University in Prague
Karlovo nám. 13, 121 35 Praha, Czech Republic
phone ++420 2 24357458, fax ++420 2 24357385
e-mail:{martid1,pajdla}@cmp.felk.cvut.cz

Abstract

This paper proposes a method for recovery of projective shape and motion from multiple images by factorization of a matrix containing the images of all scene points. Compared to previous methods, this method can handle perspective views and occlusions jointly. The projective depths of image points are estimated by the method of Sturm & Triggs [5] using epipolar geometry. Occlusions are solved by the extension of the method by Jacobs [4] for filling of missing data. This extension can exploit the geometry of perspective camera so that both points with known and unknown projective depths are used. Many ways of combining the two methods exist, and therefore several of them have been examined and the one with the best results is presented. The new method gives accurate results in practical situations, as demonstrated here with a series of experiments on laboratory and outdoor image sets. It becomes clear that the method is particularly suited for wide base-line multiple view stereo.

Keywords: projective reconstruction, structure from motion, wide base-line stereo

1. Introduction

In the past geometric and algebraic relations among uncalibrated views up to four in number have been described [5]. Various algorithms for scene reconstruction with both orthographic and perspective camera have been proposed [3, 6, 4, 5, 7, 2]. The reconstruction problem from orthographic camera is satisfactorily solved but this could not be claimed for the case of a perspective camera. The biggest problem that remained to be solved was dealing consistently with scene occlusions.

The situation is similar for two, three, and four uncalibrated images. 3D structure of a scene can be recovered up to an unknown projective transformation, where the camera geometry can be represented by the fundamental matrix, the trifocal and the quadrifocal tensor respectively [5]. For more than 4 images, image coordinates of the projections of 3D points can be combined

*This research was supported by the grants GACR 102/00/1679, MSMT KONTAKT 2001/09 and ME412, and MSM 210000012.

into a so called *measurement matrix*. Tomasi and Kanade [6] developed a factorization method of the measurement matrix for scene reconstruction with an orthographic camera and Sturm and Triggs [5] extended this method from affine to perspective projections.

Occlusions present a significant problem for reconstruction. The above mentioned Tomasi and Kanade's method solves this problem under the orthographic projection but the result depends on the choice of some initial submatrix of the measurement matrix. The method is iterative and errors may increase gradually with the number of iterations. Jacobs' method [4] improves the above approach so that no initial submatrix is needed. He combines constraints on the reconstruction derived from small submatrices of the full measurement matrix. It treats all data uniformly and is independent of image ordering.

Under perspective projection, the occlusion problem has not yet been generally solved. Method [7] by Urban et al. is dependent on the choice of a central image, which is combined with other images in a so called "cake" configuration. Only points whose projections are contained in the central image can be reconstructed. Method [2] by Fitzgibbon & Zisserman computes reconstruction from a sequence of images using trifocal tensors and fundamental matrices. Subsequent images are taken one after another and used to extend actual reconstruction.

Jacobs [4] solves reconstruction with occlusions for orthographic camera, Sturm & Triggs [5] solve reconstruction without occlusions for perspective camera. We present a novel method that builds on these two methods so that scene reconstruction from many perspective images with occlusions is obtained. Our method is independent of image ordering and treats all data uniformly.

The paper is organized as follows. The reconstruction problem is formulated in Section 2. In Section 3.1 and 3.2, algorithms [5] and [4] are reviewed, respectively. In 3.3, the new filling algorithm is presented. In 3.4, the new reconstruction method is proposed. Experiments with artificial and real data are presented in Sections 4 and 5. Section 6 summarizes the paper.

2. Problem formulation

Suppose a set of n 3D points and that some of them are visible in m perspective images. The goal is to recover 3D structure (point locations) and motion (camera locations) from the image measurements. This recovery will be called *scene reconstruction*. Let $\mathbf{x}_j$ be the unknown homogeneous coordinate vectors of the 3D points, $\mathtt{P}^i$ the unknown 3×4 projection matrices, and $\mathbf{x}_j^i$ the measured homogeneous coordinate vectors of the image points, where $i = 1, \ldots, m$ labels images and $j = 1, \ldots, n$ labels points. The basic image projection equation says that $\mathbf{x}_j^i$ are the projections of $\mathbf{x}_j$ up to unknown scale factors λ_j^i, which will be called (*projective*) *depths*: $\lambda_j^i \mathbf{x}_j^i = \mathtt{P}^i \mathbf{X}_j$. The complete set of image projections can be gathered into a single $3m \times n$ matrix equation:

$$\underbrace{\begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \ldots & \lambda_n^1\mathbf{x}_n^1 \\ \times & \lambda_2^2\mathbf{x}_2^2 & \ldots & \times \\ \vdots & & & \vdots \\ \lambda_1^m\mathbf{x}_1^m & \times & \ldots & \lambda_n^m\mathbf{x}_n^m \end{bmatrix}}_{\mathtt{J}} = \underbrace{\begin{bmatrix} \mathtt{P}^1 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}}_{\mathtt{P}} \underbrace{[\mathbf{X}_1 \ldots \mathbf{X}_n]}_{\mathtt{X}}$$

where marks $\times$ stand for unknown elements which could not be measured due to occlusions. The $3m \times n$ matrix $[\mathbf{x}_j^i]_{i=1..m, j=1..n}$ will be called the *measurement matrix*, shortly MM, whereas

J will be called the *partially rescaled measurement matrix*, shortly PRMM.

3. New reconstruction algorithm

A complete rescaled measurement matrix has rank four and therefore a projective reconstruction can be obtained by its factorization. However, from measurements in perspective images with occlusions, we can only compose a measurement matrix, let say M, which is neither complete nor rescaled. When it is at all possible to compute projective depths of some known points in M, e.g. via multi-view constraints, some missing elements of M can often be filled using the knowledge that every five columns of complete rescaled M are linearly dependent.

It would be ideal to first compute the projective depths of all known points in M and then to fill all the missing elements of M by finding a complete matrix of rank four that would be equal (or as close as possible) to the rescaled M in all elements where M is known. Such a two-step algorithm is almost the ideal linearized reconstruction algorithm, which uses all data and has a good statistical behavior. We have found that many image sets, in particular those resulting from wide base-line stereo, can be reconstructed in such two steps.

Of course, there are image sets, e.g. sets with the structure of missing data on the borderline of reconstructibility or long sequences with very factionalized tracks, which cannot be solved in the above two steps. Instead, the two steps have to be repeated while the measurement matrix M is not complete. If the correspondences between the images are such that the measurement matrix is large and diagonally dominant, then it is possible to use another reconstruction technique, e.g. to fuse the partial consecutive reconstructions [2]. However, if there is no clear sequence of images or no clear central image like in [7], the proposed algorithm has a clear advantage. It can handle arbitrary scenes in pseudo-optimal manner without a priori preferring any particular image. It provides a unique solution and thus is suited for the initialization of bundle adjustment optimizations. In what follows, we shall describe the two steps and how to combine them.

3.1. Estimating the projective depths

Many works dealt with estimating the projective depths. In this work, we used Sturm & Triggs' method [5] exploiting epipolar geometry but other methods, e.g. [3], can be applied also. Method [5] was proposed in two alternatives. The alternative with a central image is more appropriate for wide base-line stereo while the alternative with a sequence is more appropriate for video-sequences. The former will be denoted as $\omega_{cent,c}$ where c denotes the number of a central image while the latter will be denoted as ω_{seq}. Thus, we have altogether the totality $\Omega = \{\omega_{seq}, \omega_{cent,1} \dots \omega_{cent,m}\}$ of alternatives for computing the projective depths. Also, the method from [5] has to be furthermore slightly modified on account of missing data. The complete algorithm is summarized in Algorithm 1. The p-th track there denotes a subsequence of known points in $\mathbf{x}_p^1 \dots \mathbf{x}_p^m$.

3.2. Filling of missing elements in J

Before describing our new method for filling of missing data for the perspective camera, the Jacobs' algorithm for the orthographic case, which we build on, has to be explained. In [4], D. Jacobs treated the problem of missing elements in a matrix as fitting an unknown matrix of a

1. Set depths $\lambda_p^j = 1$ for known points $\mathbf{x}_p^j$ where $j = \begin{cases} 1: & \text{for } \omega_{seq} \\ c: & \text{for } \omega_{cent,c} \end{cases}$

2. For $\begin{cases} j = 1 \,..\, m-1, & i = j+1 : & \text{for } \omega_{seq} \\ j = c, & i \neq j \quad : & \text{for } \omega_{cent,c} \end{cases}$ do the following. If images i and j have enough points in common to compute a fundamental matrix uniquely[a] then compute fundamental matrix $\mathtt{F}^{ij}$, epipole $\mathbf{e}^{ij}$ and depths λ_p^i according to

$$\lambda_p^i = \frac{(\mathbf{e}^{ij} \wedge \mathbf{x}_p^i) \cdot (\mathtt{F}^{ij}\mathbf{x}_p^j)}{\|\mathbf{e}^{ij} \wedge \mathbf{x}_p^i\|^2} \lambda_p^j$$

   if the right side of the equation is defined, where $\wedge$ stands for the cross-product.

   For ω_{seq}: if the p-th track ($p = 1 \,..\, n$) is discontinuous, start with $j = b(p)$ where $b(p)$ denotes the initial image of the longest continuous subtrack of the p-th track.

[a] See Section 3.4.

Algorithm 1: Estimating the depths: alternatives ω_{seq} and $\omega_{cent,c}$

certain rank to an incomplete matrix resulting from measurements in images. When perspective images are assumed, an unknown matrix of rank 4 denoted by $\tilde{\mathtt{J}}$, $\tilde{\mathtt{J}} \in \mathbb{R}^{3m \times n}$, is fitted to PRMM $\mathtt{J}$. Technically, a basis of the linear vector space that is spanned by the columns of $\tilde{\mathtt{J}}$ is searched for. Thus, when there are 4 complete linearly independent columns in $\mathtt{J}$, then they form the desired basis. When no such 4-tuple of columns exists, the basis has to be constructed from incomplete columns. Fortunately, some 4-tuples of incomplete columns provide constraints on the basis and a sufficient number of such constraints determine it.

Let us explain what we mean by saying that an incomplete column c of $\mathtt{J}$ spans (generates) a subspace. Every complete column of $\mathtt{J}$ generates a one-dimensional subspace of $\mathbb{R}^{3m}$. Thus, an incomplete c generates a subspace V, as the smallest linear space containing all one-dimensional subspaces generated by c after replacing unknown elements by some arbitrary real numbers. Linear subspaces form a complete lattice and therefore such smallest linear space V exists. It is a subspace of $\mathbb{R}^{3m}$ and equals the linear hull of all one-dimensional subspaces. The generators of V can be obtained by constructing the column containing the known elements of c and zeros instead of the unknown ones and augmenting it with the standard basis spanning the dimensions of the unknown elements. See the example in Section 3.3.

Let the space generated by the columns of $\tilde{\mathtt{J}}$ be denoted by L. Let the span of the t-th 4-tuple of linearly independent columns of $\mathtt{J}$ be denoted by L_t. L is included in each L_t and thus also in their intersection i.e. $L \subseteq \bigcap_{t \in T} L_t$, where T is some set of indices. When the intersection is 4D, L is known exactly. If it is of a higher dimension, only an upper bound on L is known and more constraints from 4-tuples must be added. Any column in $\tilde{\mathtt{J}}$ is a linear combination of vectors of a basis of $\tilde{\mathtt{J}}$. Thus, having a basis $\mathcal{B}$ of $\tilde{\mathtt{J}}$, any incomplete column c in $\mathtt{J}$ can be completed by finding the vector $\tilde{c}$ generated by $\mathcal{B}$, which is closest to c in the subspace where c was known in $\mathtt{J}$.

The 4-tuples of columns, whose span includes L as its subspace, generate the constraint.

However, not every 4-tuple has the span that includes L as its subspace. Consider, e.g., a 4-tuple consisting of four equal columns, thus spanning only a 1D space. Even if three coordinates in one of its columns are made unknown, and thus a 4D space is spanned, L does not have to be included in the span. That is why it must be distinguished whether a given 4-tuple includes L or not, and only if it does, the constraint can be formed.

For a 4-tuple to span a subspace that contains L, each column must add to the span at least one linearly independent vector common with L. A row with some missing coordinates includes no information because the entire corresponding dimension is spanned and the constraint on L is always satisfied in this dimension. A 4-tuple of columns forms a *constraint on* L if, after discarding the rows with at least one element missing, it generates a 4D space. Such a 4-tuple will be called the *generating 4-tuple*. Because of noise in the data, the intersection $\bigcap_{t\in T} L_t$ quickly becomes empty. This is why L is searched for as the closest 4D space to spaces L_t in the sense of the minimal sum of square differences of known elements [4, 1].

3.3. Filling of missing elements for perspective cameras

Jacobs' method [4] cannot use image points with unknown depths. But, PRMM constructed from measurements in perspective images often has many such points where the corresponding depths cannot be computed. Therefore, we extended the method to exploit also points with unknown depths. It brings two advantages: (i) because the actual iteration of the two-step algorithm exploits more information, the number of iterations may decrease and consequently more accurate results may be obtained; (ii) it is possible to reconstruct more scene configurations. See Section 8 in [1] for more details about this. It is important that the proposed extension is still a linear method as was the original Jacobs' method [4].

Let us first explain the extension for two images. Suppose that λ_j^i and $\mathbf{x}_j^i$ are known for $i = 1, 2$, and $j = 1\,..\,4$ except λ_4^2. Then, consider the first four columns of $\mathtt{J}$ to be the t-th 4-tuple of columns, $\mathtt{A}_t$. A new matrix $\mathtt{B}_t$ can be defined using known elements of $\mathtt{A}_t$ as

$$\mathtt{A}_t = \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & ?\ \mathbf{x}_4^2 \end{bmatrix} \longrightarrow \mathtt{B}_t = \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 & 0 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & 0 & \mathbf{x}_4^2 \end{bmatrix}$$

It can be proved (see Corollary 1 in Appendix A in [1]) that if $\mathtt{B}_t$ is of full rank (i.e. five) then $L \subseteq Span(\mathtt{B}_t)$, which is exactly the constraint on L.

In a general situation there are also some missing elements in $\mathtt{J}$. Then, the matrix $\mathtt{B}_t$ is constructed from the t-th 4-tuple $\mathtt{A}_t$ of columns of $\mathtt{J}$ as follows:

1. Set $\mathtt{B}_t$ to $\mathtt{A}_t$.
2. Set all rows in $\mathtt{B}_t$, which contain some unknown element, to zero.
3. Replace all known points in $\mathtt{B}_t$, which have unknown depths, by zero.
4. For each unknown depth λ_p^i in $\mathtt{A}_t$, add to $\mathtt{B}_t$ a column with $\mathbf{x}_p^i$ and zeros everywhere else.
5. For each triple of rows in $\mathtt{A}_t$ containing some unknown point, add to $\mathtt{B}_t$ the standard basis spanning the dimensions of the unknown point.

The following example demonstrates the construction of $\mathtt{B}_t$ from a matrix $\mathtt{A}_t$ containing, for the sake of simplicity, only two columns instead of four:

$$\underbrace{\begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 \\ ?\ \mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 \\ \times & \lambda_2^3\mathbf{x}_2^3 \\ \lambda_1^4\mathbf{x}_1^4 & \lambda_2^4\mathbf{x}_2^4 \end{bmatrix}}_{\mathtt{A}_t} \xrightarrow{2} \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 \\ ?\ \mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 \\ \mathbf{0} & \mathbf{0} \\ \lambda_1^4\mathbf{x}_1^4 & \lambda_2^4\mathbf{x}_2^4 \end{bmatrix} \xrightarrow{3} \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 \\ \mathbf{0} & \lambda_2^2\mathbf{x}_2^2 \\ \mathbf{0} & \mathbf{0} \\ \lambda_1^4\mathbf{x}_1^4 & \lambda_2^4\mathbf{x}_2^4 \end{bmatrix} \xrightarrow{4}$$

$$\xrightarrow{4} \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \mathbf{0} \\ \mathbf{0} & \lambda_2^2\mathbf{x}_2^2 & \mathbf{x}_1^2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \lambda_1^4\mathbf{x}_1^4 & \lambda_2^4\mathbf{x}_2^4 & \mathbf{0} \end{bmatrix} \xrightarrow{5} \underbrace{\begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda_2^2\mathbf{x}_2^2 & \mathbf{x}_1^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \begin{pmatrix}1\\0\\0\end{pmatrix} & \begin{pmatrix}0\\1\\0\end{pmatrix} & \begin{pmatrix}0\\0\\1\end{pmatrix} \\ \lambda_1^4\mathbf{x}_1^4 & \lambda_2^4\mathbf{x}_2^4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathtt{B}_t}$$

3.4. Combining the filling method with estimating the depths

Due to occlusions, the computation of projective depths can be carried out in various ways depending on which depths are computed first and if and how those already computed are used to compute the others. One way of depth computation will be called a *strategy*.

Depending on the chosen strategy for estimating the depths, different subsets of depths are computed and different submatrices of PRMM are filled. For accurate data, all strategies should be equivalent. It is not so if the data is noisy. In such case, the task is to choose the strategy which results in the smallest error. It would be unrealistically costly to compute all possibilities and to choose the best one. Fortunately, we do not have to compute all of them in order to find some good one. From the structure of missing data, it is possible to predict a good strategy for estimating the depths that results in a good reconstruction. Some criterion to decide which strategy is good is needed. For scenes reconstructible in more steps, such criterion also determines which subset of depths are better to be computed first.

The following two observations have been made. First, the more iterations are performed the less accurate results are obtained because the error from the former iteration spreads in subsequent iterations, which was also mentioned in [4]. Secondly, unknown elements should not be computed from fewer data when they can be computed from more data, and thus more accurately. Both these observations support the following

Principle 1 *The more image points that are filled in one step, the smaller the expected error.*

This principle leads to a pseudo-optimal number of iterations that need to be performed. However, it is not crucial problem that such obtained strategy is only pseudo-optimal because it is possible to realize Principle 1 so that, for most scenes, only one iteration is performed.

Proposition 1 *The more depths known before the filling, the smaller the expected error.*

Proof of Proposition 1 inheres in our extension of Jacob's method (see Appendix B in [1]). Usage of Principle 1 and Proposition 1 in order of their designation proved to be a good criterion. We choose the set of strategies which fill the most points, and from this set, we choose those which

1. Estimate depths using an arbitrary strategy $\omega^* \in \Omega^*$ where

$$\Omega_{\mathcal{F}} = \left\{ \omega \in \Omega \;\middle|\; \mathcal{F}(\omega) = \max_{\tau \in \Omega} \mathcal{F}(\tau) \right\}$$
$$\Omega^* = \left\{ \omega \in \Omega_{\mathcal{F}} \;\middle|\; \mathcal{S}(\omega) = \max_{\tau \in \Omega_{\mathcal{F}}} \mathcal{S}(\tau) \right\}$$

2. Fill the missing data.

Repeat steps 1. and 2. until J is complete or no data can be filled in. Then factorize a maximal complete submatrix of J.

Algorithm 2: Scene reconstruction using a set of strategies for estimating the depths Ω

scale the most points. From the resulting set, an arbitrary strategy can be used. Let ω denote some strategy for estimating the depths and Ω denote some set of strategies. Let $\mathcal{F}(\omega)$ and $\mathcal{S}(\omega)$ denote the predicted number of newly filled unknown image points and estimated depths resp. during one iteration when ω is used. The complete new method is summarized in Algorithm 2.

The usefulness of the concept of predictor functions $\mathcal{F}, \mathcal{S} : \Omega \longrightarrow 0\,..\,mn$ consists in their ability to evaluate without neither estimating the depths nor data filling. The knowledge of which image points are known or unknown is the only information for the evaluation of $\mathcal{F}$ and $\mathcal{S}$. It is very simple (and fast) but it cannot detect degenerate configurations of points because, in fact, the multi-view tensors are not computed. If it then, when the tensor is computed, turns out that the configuration is degenerate, the second best strategy is used, etc.

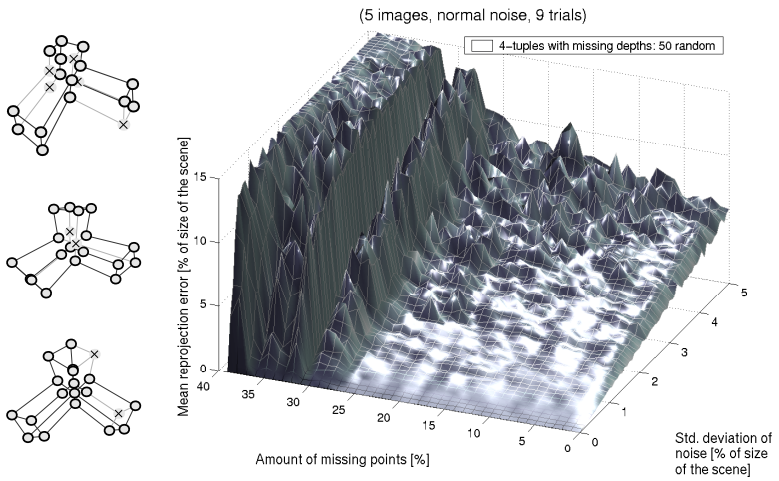
To define $\mathcal{F}$ and $\mathcal{S}$, a few symbols have to be introduced. Let x_p^i be true if and only if the image point $\mathbf{x}_p^i$ is known. Let i and j be as in the step 2. of Algorithm 1. Let $\mathcal{I}^{ij}$ be true if and only if the data of image i can be used by the filling method consistently with other images [5]. It is only possible if $i = j$ or if images i and j have enough (at least seven) points in common, which are necessary to compute a fundamental matrix uniquely.[1]

$$\mathcal{I}^{ij} \equiv \left|\{p \mid x_p^i \;\wedge\; x_p^j\}\right| \geq 7 \quad \vee \quad i = j \tag{1}$$

Let us define the predictor functions for alternative $\omega_{cent,c}$ (see [1] for ω_{seq}). Let $\mathcal{P}_p^c$ be true if and only if the p-th 3D point can be filled in by the filling method when depths were estimated using strategy $\omega_{cent,c}$. To recover a 3D point uniquely from known basis of PRMM, at least two its images are needed. Moreover, it can be proved (see Theorem 4 in Appendix A in [1]) that at least two known depths in each image are needed for the constraints on L. It means that $\mathcal{P}_p^c$ is true if and only if the p-th 3D point is seen in at least 2 images and the corresponding fundamental matrices, which are needed for estimating at least some two depths in the images, can be computed i.e.

$$\begin{aligned} \mathcal{P}_p^c &\equiv \left|\{i \mid \mathcal{I}^{ic} \;\wedge\; x_p^i\}\right| \geq 2 \\ \mathcal{F}(\omega_{cent,c}) &= \left|\{< i,p > \mid \mathcal{I}^{ic} \;\wedge\; \mathcal{P}_p^c \;\wedge\; \neg x_p^i\}\right| \\ \mathcal{S}(\omega_{cent,c}) &= \left|\{< i,p > \mid \mathcal{I}^{ic} \;\wedge\; \mathcal{P}_p^c \;\wedge\; x_p^i \;\wedge\; x_p^c\}\right| \end{aligned} \tag{2}$$

[1] The uniqueness is demanded for the depth consistency with other images.



Experiment 1: Dependency of reprojection error on noise and amount of missing data

4. Experiments with artificial scenes

For experiments with artificial scenes, a simulated scene with cubes was used. The scene simulates a real scene, hence it represents a generic situation. 20 points in space were projected by perspective cameras into several images from different locations and directions. Some image points were made unknown to simulate scene occlusions, see left hand side of Experiment 1.

Points were taken out from the scene randomly but in a uniform fashion so that, first, the numbers of missing points in each image differed maximally by one, and secondly, the numbers of images of each point differed maximally by one. Points were only removed as long as the whole scene could still be reconstructed. The necessary condition for complete reconstruction is that each image contains at least 7 points and each point has at least 2 images (see (1) and (2)). The more data available, the higher the percentage of missing data permissible. For this specific experiment, 20 points in 5 images, i.e. 65 % of missing data, is the upper bound allowable to get a complete reconstruction. But because of randomly spread holes in data, the actual level of the maximum amount of missing data for the complete reconstruction is lower.

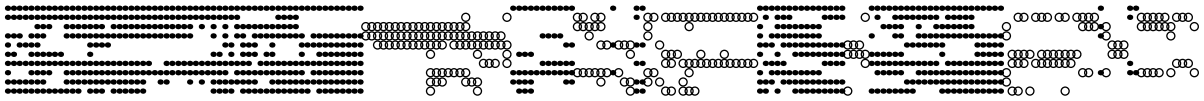
Experiment 1 shows the dependency of the reprojection error of the reconstruction using Algorithm 2 on noise and amount of missing data. Along the left horizontal axis, the amount of the missing data grows while along the right horizontal axis, standard deviation of Gaussian noise of zero mean value added to image points increases. The standard deviation of the added noise as well as the reprojection error is displayed in percentage of the scene size.

If no noise is present, the reconstruction is precise. The reprojection error grows linearly with noise with slope approximately equal one and is almost constant in the direction of missing points up to the level of missing data above which the reconstruction fails. To conclude, the new algorithm is accurate and robust with respect to noise as well as missing data.

5. Experiments with real scenes

For each experiment, one image, an error table, and the structure of PRMM are provided. The correspondences across the images have been detected either manually or by the Harris interest

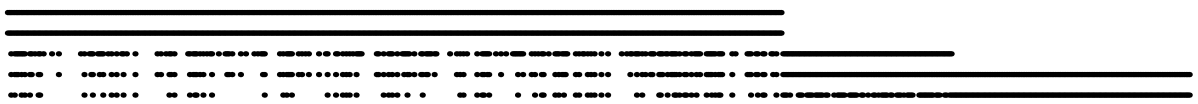
	Scene *House* Point detection Estimating the depths	10 images [2952×2003] manual cent, 1
	Amount of missing data	**47.83 %**
LM LM + BA	Mean error per image point [pxl]	**3.91** **1.44**



”•” scaled (75.7 %), ”○” unscaled (24.3 %), ” ” missing

Experiment 2: House

	Scene *Temple (Leuven)* Point detection Estimating the depths	5 images [867×591] Harris' operator seq
	Amount of missing data	**46.32 %**
LM LM + BA	Mean error per image point [pxl]	**0.49** **0.23**



”•” scaled (100.0 %), ”○” unscaled (0.0 %), ” ” missing

Experiment 3: Temple (Leuven)

operator. The table includes also the chosen strategy for estimating the depths and reprojection errors for our linear method (LM) Algorithm 2 and bundle adjustment initialized by the output of the linear method (LM + BA). All scenes have been reconstructed in one iteration of Algorithm 2.

The “House” scene (see Experiment 2) was captured on 10 images at very high resolution. Approximately 100 points were manually detected in each image. Many occlusions occurred (47.83 % data was missing) but still the reprojection error per image point, given in pixels, is very low considering the image sizes. It can be seen that our algorithm could have exploited all known data including 24.3 % unscaled points.

The data in Experiment 3 contained outliers. These were removed one after another in the following manner. The scene was first reconstructed with all the data including outliers. Then, the column of PRMM, which contained the point with the highest reprojection error, was discarded. Afterwards, the scene was again reconstructed, another column discarded etc. These two steps were repeated till the highest reprojection error was significant. For the “Temple” scene in Experiment 3, the threshold was set to 4 pixels which lead to discarding 23 columns of 719 in total. Usage of this simple technique proved that the new method does not fail if a small amount of outliers is present in the data. To conclude, the new algorithm is accurate on real scenes.

6. Summary and Conclusions

A new linear method for scene reconstruction has been proposed and tested on artificial and real scenes. The method extends and suitably combines previous methods so that the reconstruction in an entirely general situation, i.e. many images with perspective camera and occlusions, is possible. A new way of exploiting points with unknown depth was developed. Correctness of this way was proved as well as its abilities and limitations were studied in [1]. Its theoretical asset is the ability to reconstruct linearly some very small scene configurations, which can be reconstructed by other methods only nonlinearly (see Theorem 3 in [1]), cannot be reconstructed at all (see Theorem 2 in [1]), or cannot exploit all known data (see Theorem 1 in [1]). Moreover, it gives good results in practical situations as presented here.

The proposed method was intended to deal with several problems in 3D reconstruction. These were the perspective projection, many images, and occlusion. However, one problem was not taken into account explicitly and that is the problem of outliers in correspondences. Although the method was not intended to deal with outliers, it was proved that it can deal with them if they are few compared to the number of inliers (see commentary to Experiment 3). To deal well with a bigger amount of outliers, a RANSAC based algorithm could be used. This extension is left for further research.

Acknowledgement: Marc Pollefeys from K.U.Leuven provided the data used in Experiment 3.

References

[1] D. Martinec and T. Pajdla. Structure from many perspective images with occlusions. Research Report CTU-CMP-2001-20, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, July 2001.

[2] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, pages 311–326. Springer-Verlag, June 1998.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[4] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *CVPR*, pages 206–212, 1997.

[5] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV96(II)*, pages 709–720, 1996.

[6] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In *IJCV(9), No. 2*, pages 137–154, November 1992.

[7] M. Urban, T. Pajdla, and V. Hlaváč. Projective reconstruction from N views having one view in common. In *Vision Algorithms: Theory & Practice*. Springer LNCS 1883, pages 116-131, September 1999.

Correspondences From Epipolar Plane Images, Experimental Evaluation *

Martin Matoušek, Václav Hlaváč
Czech Technical University, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
121 35 Prague 2, Karlovo náměstí13, Czech Republic
phone +420 2 2435 7637, fax +420 2 2435 7385
{xmatousm,hlavac}@cmp.felk.cvut.cz

Abstract

The method seeking correspondences in Epipolar Plane Images (EPI) is presented. The principal idea is to employ dense sequence to get more information which could guide the correspondence algorithm. Theoretically a simple technique finding homogeneous straight lines in EPI suffices to establish correspondences.

This paper verifies the method in real experiments. The paper builds on results of [6]. Real data uncovered several optical and mechanical phenomena which introduce additional problems. They are revealed in the paper one by one and undone.

Keywords: epipolar plane image, correspondence, Lambertian surface

1 Introduction

We report about our attempts to improve correspondence between images, i.e. seeking coordinates of pixels which match the single point in the 3D scene. The underlined idea is to use a dense epipolarly aligned sequence of images to get more information than the wide baseline stereo has at hand. However, we do not follow the optical flow paradigm which uses local differential operators and cannot avoid error accumulation.

We have suggested the method seeking for correspondences and tested it on synthetic images [6]. This paper shows experimental results on real data, describes several pitfalls we trapped in, and comments on methods allowing us to extricate from traps. We believe that our relatively thorough study uncovered phenomena which are not so obvious.

*This research was supported by the Czech Ministry of Education under Research Programme MSM 212300013 and by the Grant Agency of the Czech Republic under Project GACR 102/00/1679.

2 Related Work

The first group of papers relates to computational binocular stereo. The state-of-the-art methods in stereo are of interest for us because we like to compare our results with them. The dynamic programming-based methods are commonly used [4]. We use the better correspondence matcher suggested by R. Šára [10].

The second group deals with contributions that use more than two images to establish correspondences. The paper [9] verifies correspondences by comparing intensity values in more than two images. The other possibility is to combine wide base and a larger data set. The moving wide base stereo rig provides two dense sequences [5] in which features are tracked. Short sections of tracked trajectories are obtained and ease finding correspondences.

The third group of papers uses the completely stacked sequence (spatio-temporal block) and aims at deriving scene structure from it. For instance, signal processing techniques (filtering) can be used to detect occlusions [3, 7].

3 Epipolar Plane Image

Let us assume the sequence of images $f(x, y, t)$ where x, y are coordinates of the pixel and t is an index in the sequence. Consider a particular case—a sequence of images captured by cameras with collinear locations of projection centers. Epipolar lines belonging to a selected epipolar plane (one epipolar line per image) can be stacked along the parameter t and forms an *epipolar plane image* (EPI).

Let us simplify the situation further and consider even more special case of an epipolarly rectified sequence which was sampled equidistantly in t. The situation is illustrated in Figure 1a.

Such simplified EPI has several properties that are useful.

- The space in which correspondence is sought is 1D only. It is assumed that epipolar lines do not depend on each other. This property allows to decompose the correspondence problem to simpler subproblems (which can be processed in parallel).
- The point in 3D scene maps to the *straight line* in EPI.

4 Simplifying Assumptions

Our aim is to analyze the simplest case. Therefore the following assumptions were considered.

- Surfaces in the scene are opaque and have Lambertian reflectance.
- The scene is occlusion-free, i.e., ordering and uniqueness constraint are not violated.
- The scene is covered by a 'sufficiently good' texture. The unambiguous solution of the correspondence problem can be obtained only if there are no extended regions with zero or undefined gradient in a light-field [2]. Let us note that EPI is a special case of a light--field.

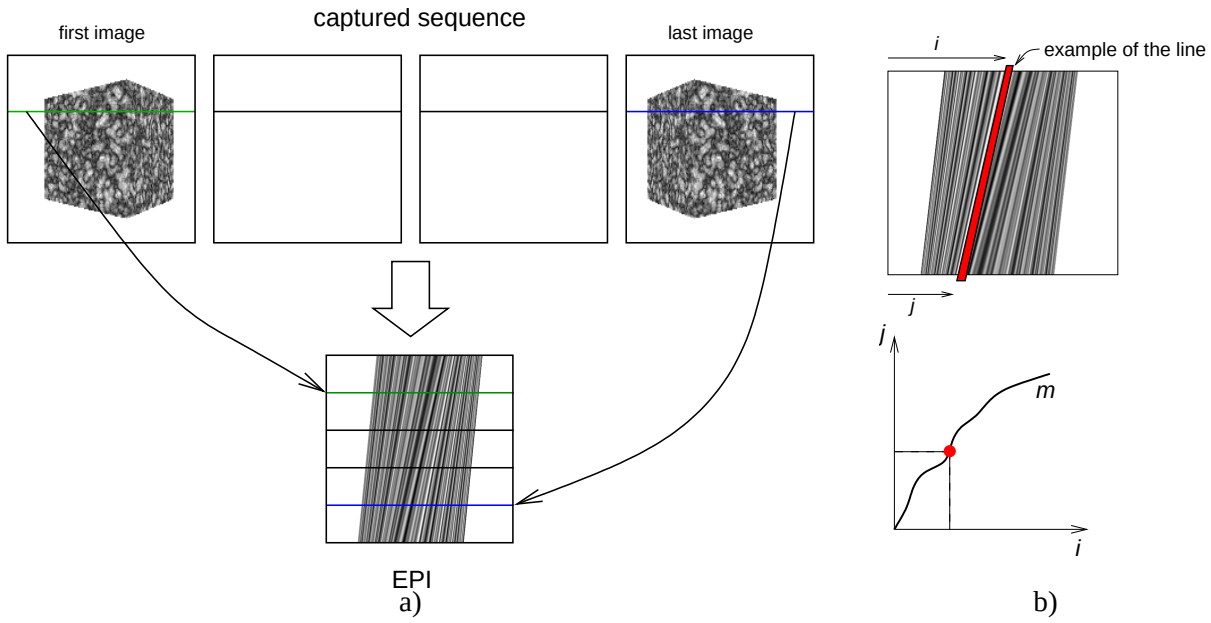


Figure 1: (a) EPI formation scheme. (b) (top) Selected EPI with one line shown and (bottom) search space in which correspondences are sought.

5 Seeking Correspondences

An elementary entity is the correspondence between a point in the first and in the last image. This correspondence of column coordinates in the first and the last image (denoted as a pair (i, j)) is realization of a single point in 3D scene. The *line in EPI* is another representation of the correspondence, see Figure 1b. Theoretically, the line corresponding to a single point in 3D scene should be of the same value. Practically, such line has only similar values due to various artifacts and noise. Analysis of related phenomena is actually the contribution of this paper, still to come.

The cost $c(i, j)$ quantifies our believe that the correspondence candidate agrees with the observed 3D point. The cost is calculated according to values (intensities) along the line found in EPI. *Variance* of the intensities was chosen similarly to [9]. The domain of the cost $c(i, j)$ is $\mathbb{R}^2$ because positions with subpixel accuracy can be obtained by interpolation neighbouring coordinates. Here we take advantage of the fact that there are many images at hand in the sequence.

Having defined cost function $c(i, j)$ it is possible to examine all possible correspondence pairs. The whole set of correspondences considered as *mapping* m between i and j is sought Figure 1b. The mapping is estimated by minimization of the global criterion $\mathcal{J}(m)$ which sums partial increments of $c(i, j)$ along the m.

The mapping m is a strictly increasing function because of ordering and uniqueness constraints. The estimate of the optimal mapping m^* can be computed by minimizing the criterion by dynamic programming which is computationally efficient. The dynamic programming is a discrete optimization method. In our case the raster is sampled in subpixel accuracy.

6 Experimental Evaluation on Real Data (Lessons Learned)

We have described the method so far. We can proceed now to actual experiments, problems that occurred, and our thoughts how to avoid them.

6.1 Experimental setup

The studied scene consisted of a single 15×15 cm bathroom tile. We captured reverse side of the tile. Our expectation was that unglazed surface of ceramics has almost Lambertian reflectance properties. The reverse side of the tile was sprayed by a random texture.

Images were captured by a stationary camera while the homogeneously illuminated scene (reverse side of the tile) was moved along the straight line. This configuration was chosen to minimize camera vibrations. The tile was moved on the translation table OPTEN by 15 cm. The images were captured in 1.5 mm steps, i.e, a hundred of images was taken.

A PULNIX TM-1001 digital CCD camera (1k×1k resolution, 9mm chip size) was used equipped with TAMRON 89698 25 mm lens mount (F-stop 12 was set). The camera was placed at the distance 90 cm from the scene.

The bathroom tile was chosen deliberately because the form of its reverse side suits our purpose. Even no ground truth about the 3D shape is available we can expect that the observed surface is bound by two parallel planes with known distance. The reverse side of the tile is covered by regular stripe-like hollows with depth approximately 0.2 mm. The approximate value of disparity between border images in the sequence was 427. This leads to expected difference of a disparity around 0.1 pixel.

Calibration markers (chessboard) were placed on the captured tile (Figure 6b) to ease epipolar rectification. The sequence $f(x, y, t)$ was separated to individual EPIs. Only one selected EPI was used in experiments.

6.2 Checking lines in EPI

This experiment attempts to verify the assumption that the intensities along the line in EPI corresponding to a single point in 3D scene are homogeneous and bear enough information about correspondences. First a single line in EPI was analysed. The intensity profile along the line mapping a correct correspondence should be constant in the ideal case. The observed profile is different, see top curve in Figure 2a. This would not matter if the 'correct' profile was significantly more homogeneous than 'erroneous' profile. This was tested by looking at profiles that are not correct, see two bottom curves in Figure 2a. We observed significant deformation of the profile.

Next test tries to identify what caused the problem. Fifty intensity profiles corresponding to 'correct' lines and covering the whole EPI were displayed in one figure. Each profile was normalized so that its maximal value is 1 and aligned according to the position of the pixel, see Figure 2b.

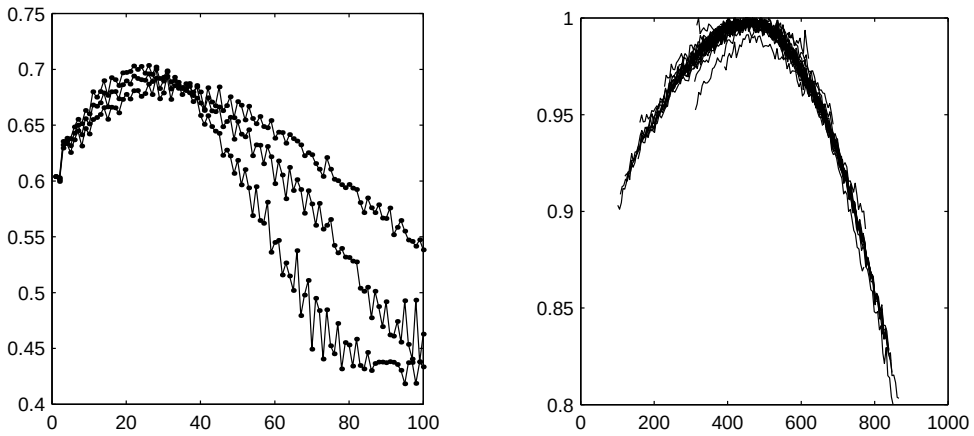


Figure 2: Intensity values related to the correct line in EPI. (a) Three intensity profiles. Horizontal axis gives t and vertical axis gives intensities. The top one corresponds to correct line (i, j). The other two profiles correspond to the modified lines $(i, j+1)$ and $(i, j-1)$. (b) Intensities along a several manually placed correct lines, normalized to maxima, and aligned according to position in image. Horizontal axis gives pixels and vertical axis gives intensities.

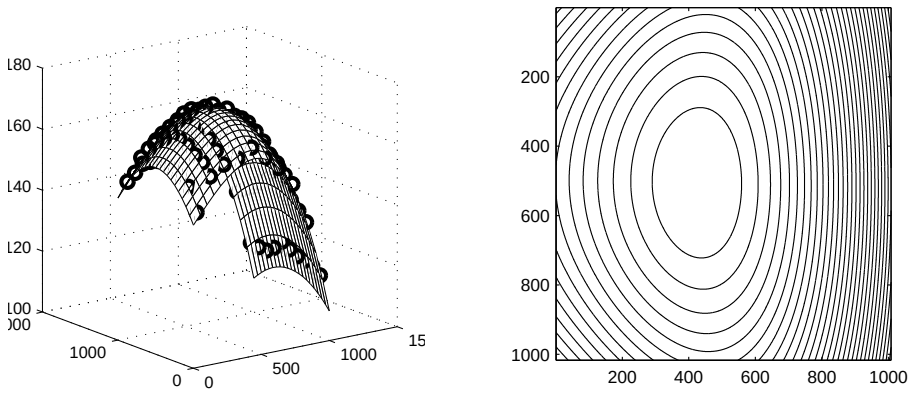
We noticed that the dominant shape of the profile depends only on the position in the image and does not depend on the position in the scene. We also observed that the intensity profiles are not symmetric. We concluded that the most significant phenomenon behind is photometric deficiency of the camera, namely field-darkening. Our decision was to perform proper photometric and radial camera calibration.

6.3 Photometric and Radial Calibration of the Camera

Field-darkening causes decrease of intensities for off-axis pixels. There are three main reasons for the phenomenon: (a) $cos^4\alpha$ effect, where α gives the angle between the ray and the optical axis, (b) vignetting effects which observed for open iris because not all iris is illuminated (likely absent in our case, F-stop = 12), and (c) pupil aberration (non-uniform light distribution across the aperture). The [1] concluded from experiments that the fall-off surface can be non-symmetric.

We obtained the fall-off surface by measuring the image irradiance depending on the position in the image for constant scene radiance. In [1] the homogeneous light source placed parallel to the camera was used. We could not use such a light source. We proposed alternative method. The small surface patch with almost Lambertian reflectance and almost homogenously illuminated was used as a light source. The camera was panned and tilted. Even if the light source is not exactly homogeneous the amount of light entering camera does not change because the camera center is at the same position. The measured surface is in Figure 3.

Having in mind that the method seeking correspondences reaches subpixel accuracy we also corrected the radial lens distortion. We used the RADIALD toolbox [8].



a) b)

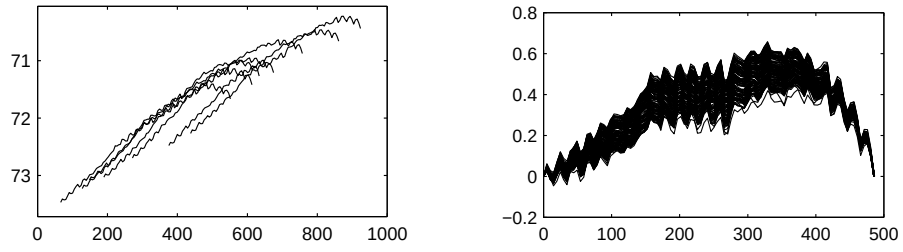
Figure 3: Measured fall-off phenomenon shown as the (a) 3D surface, and the (b) contour plot. The values (small circles) were measured in 10×10 regular grid and approximated by the second-order polynomial. Coordinates are given in pixels.

6.4 Do cameras lie on the straight line?

Theoretical model assumes many cameras lying on the straight line. Only one still camera is is used and the object moves in our case. This experiment validates the precision of the 'straight line' assumption.

Our experimental object, i.e. reverse side of the bathroom tile, contains calibration pattern (chessboard) which allows precise localization. We observed trajectories of selected location during movement, see Figure 4a. The y-differences of all targets from linear trajectory are depicted in Figure 4b. We conclude from uniformity of y-differences that our translation table is not precise enough and the 'straight line' assumption is significantly violated.

We attempted to suppress this effect. The mean value of y-differences for particular image in the sequence was used for shift the image. Figure 5 shows trajectories of the same selected calibration targets after shift correction. However, effects violating 'straight line' assumption cannot be undone in their entirety because the center of the camera changes position and the translation table moves in other directions too.



a) b)

Figure 4: Plot of location of calibration targets during translation table movement. Coordinates are in pixels. (a) Trajectory of a few selected targets. (b) y-differences from linear trajectory for all targets.

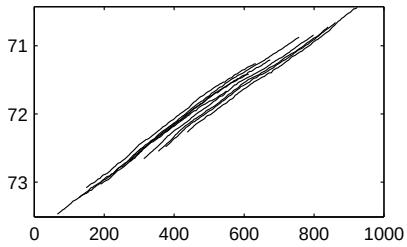
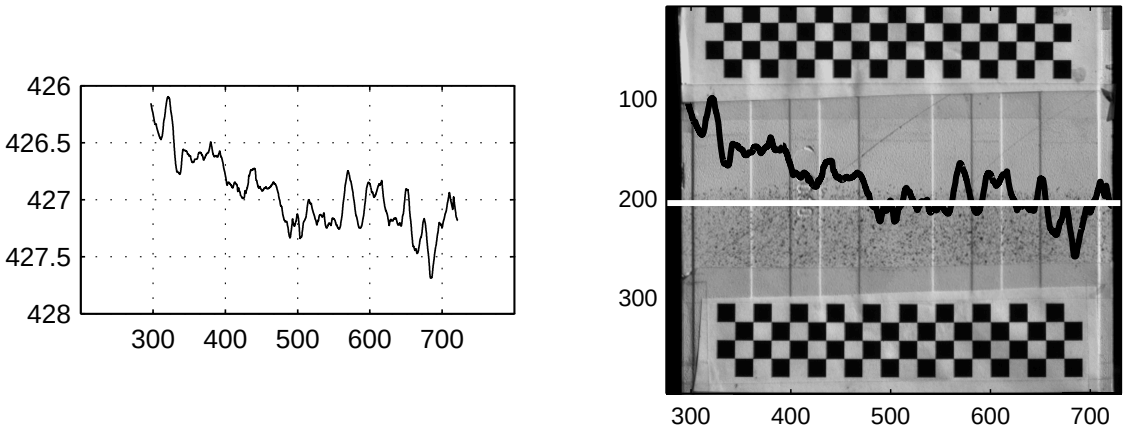


Figure 5: Plot of locations of a few selected calibration targets after shift correction. Coordinates are in pixels.

7 Final Results and Conclusions

All phenomena described above were understood, modelled, and suppressed. EPI made from corrected images was an input to the correspondence algorithm. The disparity plot is shown in Figure 6. The underlying depth variations can be detected but it is not so easy to distinguish them from noise.



a) b)

Figure 6: Computed disparity values. Both axes are in pixels. (a) Disparity only. (b) Disparity (scaled and shifted) overlayed on the original captured image. White line shows selected y-coordinate.

For comparison we calculated disparity using Šára's stereo matcher [10]. First and last images from the sequence were used. The method calculates correspondences with precision up to pixel correctly. Next the Šára's sub-pixel disparity correction (not published yet) was applied. In the resulting disparity map the 3D structure of the reverse side of the bathroom tile was not revealed at all.

Our method reveals depth variation where Šára's method does not but oscillates. It means that the solution is sometimes worse than Šára's stereo matcher at whole pixels.

To conclude, we admit that the phenomena involved are complicated. Even the idea aiming at finding distinct homogeneous lines in EPI is simple and appealing, it is difficult to meet our assumption in practical cases.

We like thing the problem over again and to do some more experiments in near future.

We acknowledge fruitful discussions with T. Werner, T. Pajdla, R. Šára, O. Drbohlav that oriented our work. V. Smutný helped us in setting experiments.

References

[1] Manoj Aggarwal, Hong Hua, and Narendra Ahuja. On cosine-fourth and vignetting effects in real lenses. In Jim Little and David Lowe, editors, *ICCV'01: Proc. 8th IEEE Intl. Conf. on Computer Vision*, volume 1, pages 472–479, Vancouver, British Columbia, Canada, July 2001. IEEE Computer Society Press, Los Alamitos, CA, USA.

[2] Simon Baker, Terence Sim, and Takeo Kanade. A characterization of inherent stereo ambiguities. In Jim Little and David Lowe, editors, *ICCV'01: Proc. 8th IEEE Intl. Conf. on Computer Vision*, volume 1, pages 428–435, Vancouver, British Columbia, Canada, July 2001. IEEE Computer Society Press, Los Alamitos, CA, USA.

[3] George T. Chou. A model of figure-ground segregation from kinetic occlusion. In Eric Grimson, editor, *ICCV'95: Proc. 5th IEEE Intl. Conf. on Computer Vision*, pages 1050–1057, Cambridge, Massachusetts, June 1995. IEEE Computer Society Press, Los Alamitos, CA, USA.

[4] Ingemar J. Cox, Sunita L. Higorani, Satish B. Rao, and Bruce M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[5] Pui-Kuen Ho and Ronald Chung. Stereo-motion with stereo and motion in complement. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(2):215–220, February 2000.

[6] Martin Matoušek, Tomáš Werner, and Václav Hlaváč. Accurate correspondences from epipolar plane images. In Boštjan Likar, editor, *Proc. Computer Vision Winter Workshop*, pages 181–189, Bled, Slovenia, February 2001. Slovenian Pattern Recorgnition Society.

[7] Sourabh A. Niyogi. Detecting kinetic occlusion. In Eric Grimson, editor, *ICCV'95: Proc. 5th IEEE Intl. Conf. on Computer Vision*, pages 1044–1049, Cambridge, Massachusetts, June 1995. IEEE Computer Society Press, Los Alamitos, CA, USA.

[8] Tomáš Pajdla, Tomáš Werner, and Václav Hlaváč. Correcting radial lens distortion without knowledge of 3-D structure. Research Report CTU-CMP-1999-4, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, June 1997.

[9] Sébastien Roy and Ingemar J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In Sharat Chandran and Uday Desai, editors, *ICCV'98: Proc. 6th IEEE Intl. Conf. on Computer Vision*, pages 492–499, Bombai, India, January 1998. IEEE, Narosa Publishing House.

[10] Radim Šára. Sigma-delta stable matching for computational stereopsis. Research Report CTU–CMP–2001–25, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, September 2001.

Spectral Embedding of Graphs

Bin Luo[1,2] Richard C. Wilson[1] and Edwin R. Hancock [1]

[1]Department of Computer Science,

University of York,York YO1 5DD, UK.

[2]Anhui University, P.R. China

Tel. +44-1904-433374

E-mail: erh@cs.york.ac.uk

Abstract

In this paper we explore how to embed symbolic relational graphs with unweighted edges in a pattern-space. We adopt a graph-spectral approach. We use the leading eigenvectors of the graph adjacency matrix to define clusters of nodes. For each cluster, we compute vectors of spectral properties. We embed these vectors in a pattern-space using two contrasting approaches. The first of these involves performing principal components analysis on the covariance matrix for the spectral pattern vectors. The second approach involves performing multidimensional scaling on the L2 norm for pairs of pattern vectors. We demonstrate the both methods result in well-structured view spaces for graph-data extracted from 2D views of 3D objects.

1 Introduction

Relational graphs have proved alluring as structural representations for both 2D and 3D shape in computational vision. Barrow and Burstall [1], and, Fischler and Enschlager [9] were among the first to demonstrate the potential of relational graphs as abstractions for pictorial information. Since then graph-based representations have been exploited widely for the purposes of shape representation, segmentation, matching and recognition. However, one of the problems that hinders the manipulation of large sets of graphs is that of measuring their similarity. This problem arises in a number of situations where graphs must be matched or clustered together. The large-scale matching problem arises in tasks involving recognition from image databases The graph-clustering task arises when the unsupervised learning of the class-structure of sets of graphs is attempted. Concrete examples here include the organisation of large structural data-bases [18] or the discovery of the view-structure of objects [8].

There are a number of ways in which the similarity of graphs may be measured. One of the classical methods is to use the concept of graph edit distance. This is an extension of the classical string edit distance, or Levenshtein distance to graphs. The idea of using graph edit distance was first explored by Fu and his co-workers [16]. Here edit distances are computed

using separate costs for the relabelling, the insertion and the removal of both nodes and edges. Recently, Bunke [4] has shown that the graph edit distance and the size of the maximum common subgraph are related under certain restrictions on the edge and node edit costs. Torsello and Hancock [20] have exploited this observation to efficiently compute tree-edit distance. By using the Motzkin-Strauss theorem they show how to compute an approximation to the edit distance using relaxation labelling. Another approach to computing graph similarity is to adopt a probabilistic framework. Here there are two contributions worth mentioning. First, Christmas, Kittler and Petrou [5] have developed an evidence combining framework for graph-matching which uses probability distribution functions to model the pairwise attribute relations defined on graph-edges. Second, Wilson and Hancock [22] show how to measure graph-similarity using a probability distribution which models the number of relabelling and graph-edit operations when structural errors are present [22]. Finally, set theoretic methods may be used. Here Huet and Hancock have used a robust variant of the Hausdorff distance to measure the similarity between attributed relational graphs [11].

There are a number of observations that can be drawn from this brief review of the literature. First, the computation of graph edit distance is potentially an NP-hard problem since it relies either explicitly or implicitly on the availability of correspondences between nodes and edges. Second, it is considerably easier to characterise the similarity of attributed or weighted graphs than purely symbolic ones. The reason for this is that when attribute information is to hand then the search-space for correspondences may be significantly reduced using similarity heuristics. For purely symbolic graphs the only information available for reducing the search-space is that provided by the topology of the edge connectivity pattern (for instance, the degree of different nodes) and this may result in highly ambiguous correspondences.

In this paper we aim to investigate whether graphs can be represented in a stable way using vectors of spectral attributes. With this representation to hand graph similarity can be measured by computing a distance norm between vectors. Learning class-structure or imposing organisation on the graphs can be achieved by clustering the vectors. Moreover, the pattern-space spanned by the graph-vectors can be simplified using techniques such as principal or independent components analysis. Unfortunately, the process of embedding graphs in a vector-space is not a straightforward one. The reasons for this are twofold. First, correspondences are required so that nodes and edges can be mapped to the relevant component of the pattern-vector. Second, there needs to be a means of accommodating graphs which contain different numbers of nodes and edges.

In this paper, to overcome these two problems, we provide a graph-spectral approach to the embedding problem. Spectral graph theory is a branch is mathematics which aims to characterise the properties of unweighted graphs using the eigenvalues and eigenvectors of the adjacency matrix or the closely related Laplacian matrix [6]. There are a number of well-known results. For instance, the degree of bijectivity of a graph is measured by the eigenvalue gap, the distribution of cycle length can be computed using a moments expansion of the eigenvalues, and the steady-state random walk on a graph is given by the leading eigenvector of the adjacency matrix. Although conceptually alluring, the main problem with spectral properties is that they are notoriously sensitive to small changes in the structure of the adjacency matrix.

Despite the fact that the eigenvectors of the adjacency matrix are highly susceptible to

changes in graph-structure, the eigenvalue order is more stable. This property has been repeatedly and successfully exploited in the computer vision literature to develop pairwise clustering algorithms. For instance, Shi and Malik use the leading eigenvector to perform image segmentation using the iterative normalised cut technique [19]. Sarkar and Boyer have used property matrix spectra for line segment grouping [17]. Inoue and Urahama [14] have used the leading eigenvector to develop a batch-iterative pairwise clustering algorithm.

Our aim in this paper is to use the pairwise clustering property of the eigenvectros of the adjacency matrix for the purposes of constructing pattern-vectors for graphs. The idea is as follows. Each of the leading eigenvectors of the adjacency matrix is taken to represent a pairwise cluster of nodes. The significance of the cluster is determined by the magnitude of the associated eigenvalue. The degree of cluster membership of graph-nodes to clusters is gauged by their co-efficients in the cluster eigenvector. We perform our vectorial embedding of the graphs using graph-theoretic attributes for the clusters. Each component of the vector represents a different spectral cluster. The order of the components of the vector is the magnitude order of the eigenvalues of the adjacency matrix. For each cluster, we use the components of the cluster eigenvectors to compute weighted spectral attributes. In this way we solve the problem of finding correspondences between nodes and vector-components. We compute both unary and binary cluster attributes. The unary attributes include the subgraph volume, the cluster perimeter length and the subgraph Cheeger number. The binary attributes represent the pairwise arrangement of the clusters. Here we investigate the shared cluster perimeter length and the number of graph-edges between cluster centres.

Once the cluster feature-vectors are to hand, then we investigate two alternative routes to embedding them in a pattern-space. The first of these involves principal components analysis. Here we construct the covariance matrix for the spectral pattern vectors of the graphs. We project the pattern-vectors onto the leading eigenvectors of the covariance matrix to give a graph pattern-space. The second approach is based on multidimensional scaling. Here we compute a matrix of pairwise similarities between pairs of graphs using the L2 distance norm.

2 Spectral Pattern Vectors

In this paper we are concerned with the set of graphs $G_1, G_2, .., G_k, ..., G_N$. The kth graph is denoted by $G_k = (V_k, E_k)$. where V_k is the set of nodes and $E_k \subseteq V_k \times V_k$ is the edge-set. Our approach in this paper is a graph-spectral one. For each graph G_k we compute the adjacency matrix A_k. This is a $|V_k| \times |V_k|$ matrix whose element with row index i and column index j is

$$A_k(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

From the adjacency matrices $A_k, k = 1...N$ at hand, we can calculate the eigenvalues λ_k by solving the equation $|A_k - \lambda_k I| = 0$ and the associated eigenvectors ϕ_k^ω by solving the system of equations $A_k\phi_k^\omega = \lambda_k^\omega \phi_k^\omega$. We order the eigenvectors according to the decreasing magnitude of the eigenvalues, i.e. $|\lambda_k^1| > |\lambda_k^2| > \ldots |\lambda_k^{|V_k|}|$ The eigenvectors are stacked in order to construct the modal matrix $\Phi_k = (\phi_k^1|\phi_k^2|\ldots|\phi_k^{|V_k|})$.

We use only the first n eigenmodes of the modal matrix to define spectral clusters for each graph. The components of the eigenvectors are used to compute the probabilities that nodes

belong to clusters. The probability that the node indexed $i \in V_k$ in graph k belongs to the cluster with eigenvalue order ω is

$$s_{i,\omega}^k = \frac{|\Phi_k(i,\omega)|}{\sum_{\omega\prime=1}^{n} |\Phi_k(i,\omega\prime)|} \tag{2}$$

Our aim is to use spectral features for the modal clusters of the graphs under study to construct feature-vectors. To overcome the correspondence problem, we use the order of the eigenvalues to establish the order of the components of the feature-vectors. We study a number of features suggested by spectral graph theory.

Our first vector of spectral features is constructed from the ordered eigenvalues of the adjacency matrix. For the graph indexed k, the vector is

$$B_k = (\lambda_k^1, \lambda_k^2, ..., \lambda_k^n)^T \tag{3}$$

The first pairwise cluster attribute studied is the shared perimeter of each pair of clusters. For the pair subgraphs S and T the perimeter is the set of nodes belong to the set $P(S,T) = \{(u,v)|u \in S \wedge v \in T\}$. Hence, our cluster-based measure of shared perimeter for the clusters is

$$U_k(u,v) = \frac{\sum_{(i,j)\in E_k} s_{i,u}^k s_{j,v}^k A_k(i,j)}{\sum_{(i\prime,j\prime)\in E_k} s_{i\prime,u}^k s_{j\prime,v}^k} \tag{4}$$

Each graph is represented by a shared perimeter matrix U_k. We convert these matrices into long vectors. This is obtained by stacking the columns of the matrix U_k in eigenvalue order. The resulting vector is $B_k = (U_k(1,1), U_k(1,2), ..., U_k(1,n),$
$U_k(2,1)....., U_k(2,n,), ...U_k(n,n))^T$ Each entry in the long-vector corresponds to a different pair of spectral clusters.

The second pairwise attribute is the between cluster distance. This is defined as the path length, i.e. the minimum number of edges, between the most significant nodes in a pair of clusters. The most significant node in a cluster is the one having the largest co-efficient in the eigenvector associated with the cluster. For the cluster indexed u in the graph indexed k, the most significant node is

$$i_u^k = \arg\max_i s_{iu}^k \tag{5}$$

To compute the distance, we note that if we multiply the adjacency matrix A_k by itself l times, then the matrix $(A_k)^l$ represents the distribution of paths of length l in the graph G_k. In particular, the element $(A_k)^l(i,j)$ is the number of paths of length l edges between the nodes i and j. Hence the minimum distance between the most significant nodes of the clusters u and v is

$$d_{u,v} = \arg\min_l (A_k)^l(i_u^k, i_v^k) \tag{6}$$

If we only use the first n leading eigenvectors to describe the graphs, the between cluster distances for each graph can be written as a n by n matrix which can be converted to a $n \times n$ long-vector $B_k = (d_{1,1}, d_{1,2}, ...d_{1,n}, d_{2,1}.....d_{n,n})^T$.

3 Embedding the Spectral Vectors in a Eigenspace Space

In this section we describe two methods for embedding graphs in eigenspaces. The first of these involves performing principal components analysis on the covariance matrices for the spectral pattern-vectors. The second method involves performing multidimensional scaling on a set of pairwise distance between vectors.

3.1 Eigendecomposition of the image representation matrices

Our first method makes use principal components analysis and follows the parametric eigenspace idea of Murase and Nayar [15]. The relational data for each image is vectorised in the way outlined in Section 3. The N different image vectors are arranged in view order as the columns of the matrix $S = [B_1|B_2|\ldots|B_k|\ldots|B_N]$.

Next, we compute the covariance matrix for the elements in the different rows of the matrix S. This is found by taking the matrix product $C = SS^T$ We extract the principal components directions for the relational data by performing an eigendecomposition on the covariance matrix C. The eigenvalues λ_i are found by solving the eigenvalue equation $|C - \lambda I| = 0$ and the corresponding eigenvalues $\vec{e}_i$ are found by solving the eigenvector equation $C\vec{e}_i = \lambda_i\vec{e}_i$.

We use the first 3 leading eigenvectors to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the three orthogonal vectors by $E = (\vec{e}_1, \vec{e}_2, \vec{e}_3)$. The individual graphs represented by the long vectors $B_k, k = 1, 2, \ldots, N$ can be projected onto this eigenspace using the formula $\vec{x}_k = \vec{e}^T B_k$ Hence each graph G_k is represented by a 3-component vector $\vec{x}_k$ in the eigenspace.

3.2 Multidimensional Scaling

Multidimensional scaling(MDS)is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. The classical multidimensional scaling method was proposed by Torgenson[23].Shepard and Kruskal developed a different scaling technique called ordinal scaling[12]. Here we intend to use the method to embed the graphs extracted from different viewpoints in a low-dimensional space.

To commence we require pairwise distances between graphs. We do this by computing the L2 norms between the spectral pattern vectors for the graphs. For the graphs indexed $i1$ and $i2$, the distance is

$$d_{i1,i2} = \sum_{\alpha=1}^{K}\Big[B_{i1}(\alpha) - B_{i2}(\alpha)\Big]^2 \tag{7}$$

The pairwise similarities $d_{i1,i2}$ are used as the elements of an $N \times N$ dissimilarity matrix D, whose elements are defined as follows

$$D_{i1,i2} = \begin{cases} d_{i1,i2} & \text{if } i_1 \neq i2 \\ 0 & \text{if } i1 = i2 \end{cases} \tag{8}$$

In this paper, we use the classical multidimensional scaling method to embed our the view-graphs in a Euclidean space using the matrix of pairwise dissimilarities D. The first step of MDS is to calculate a matrix T whose element with row r and column c is given by $T_{rc} = -\frac{1}{2}[d^2_{rc} - \hat{d}^2_{r.} - \hat{d}^2_{.c} + \hat{d}^2_{..}]$ where $\hat{d}_{r.} = \frac{1}{N}\sum_{c=1}^{N} d_{rc}$ is the average dissimilarity value over the rth

row, $\hat{d}_{.c}$ is the similarly defined average value over the cth column and $\hat{d}_{..} = \frac{1}{N^2}\sum_{r=1}^{N}\sum_{c=1}^{N} d_{r,c}$ is the average similarity value over all rows and columns of the similarity matrix T.

We subject the matrix T to an eigenvector analysis to obtain a matrix of embedding co-ordinates X. If the rank of T is $k, k \leq N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0$. The corresponding ordered eigenvectors are denoted by $\vec{e}_i$ where λ_i is the ith eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is $X = [\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_k]$ where $\vec{f}_i = \sqrt{\lambda_i}\vec{e}_i$ are the scaled eigenvectors. For the graph indexed i, the embedded vector of co-ordinates is $\vec{x}_i = (X_{i,1}, X_{i,2}, X_{i,3})^T$.

4 View-based Object Recognition

To provide an experimental vehicle for our new eigen-space representation of graphs, we focus on the problem of view based object recognition. This topic has been studied in the computer vision literature for over three decades [21]. Stated simply, the idea is to compile a series of images of an object as the set of possible viewing directions is spanned. The images are then subjected to some form of dimensionality reduction [15] or information abstraction [10]. This is a process of learning that may involve either feature extraction, principal components analysis or the abstraction of the main structures using a relational description Once a condensed image representation is to hand, then the aim is to embed the different images in a low-dimensional representation which can be traversed with viewing direction. Recognition and pose recovery may be effected by finding the closest representative view. In other words, the aim is to embed high-dimensional view based image data in a low dimensional structure which is suitable for view indexing.

Broadly speaking there are two different approaches to this problem. The first of these is to construct an eigenspace [15]. This approach was first introduced by Murase and Nayar [15], and has since been refined in a number of different ways. The idea is to perform principal components analysis on the images collected as the viewing direction and illumination direction [2] are varied. This is achieved by first storing each image as a long-vector. Next the covariance matrix for the long-vectors is found. The eigenvectors of the covariance matrix define the directions of principal components in the space spanned by the long-vectors. Dimensionality reduction is achieved by projecting the original images onto the principal component directions and selecting the components corresponding to the leading eigenvectors. The method has mainly been applied to pixel based image representations.

The second approach to the problem is older and involves constructing a relational abstraction of the features present in the raw images [21, 13]. The aim here is to extract surfaces or boundary groupings from 2.5D range data or 2D image data. From this data the view occurrence of the different image structures is noted. Hence a group of images which all yield the same feature configuration are deemed to belong to a common view [7]. View indexing can be achieved by matching a relational arrangement of image structures to the set of corresponding representative view graphs. This approach to the problem has its origins in the work of Freeman on characteristic views. It has also stimulated the study of aspect graphs [21]. The topic draws heavily on work from psychology [3] and differential topology [13].

5 Experiments

In this section we report experiments on 2D image sequences of 3D objects under slowly varying changes in viewer angle. Here we study both relatively simple polyhedral objects. From each object in the view sequence, we extract corner features. From the extracted corner points we construct Delaunay graphs. We experiment with two different sequences. In Figures 1 we show the raw images and extracted graphs for the INRIA MOVI toy house sequence. Figures 2 shows a house sequence for a model of a Swiss chalet which we collected in-house. There are a number of "events" in the sequences. For instance in the MOVI sequence, the right-hand gable wall disappears after the 12th frame, and the left-hand gable wall appears after the 17th frame. Several of the background objects also disappear and reappear. In the Swiss chalet sequence, the front face of the house disappears after the 15th frame.

Figures 3 to 5 show the results obtained with vectors of different spectral attributes. In each case the top row shows the results obtained for the MOVI house, while the bottom row is for the chalet sequence. In the left-hand column of each figure, we show the eigenspace extracted by applying PCA to the covariance matrix for the spectral feature vectors. The middle column shows the matrix of pairwise vector distances used as input to MDS. Finally, the right-hand column shows the result of applying MDS to the matrix of distances. Figure 3 shows the results obtained with the vector of adjacency matrix eigenvalues (i.e. the graph-spectra), Figure 4 is the result obtained with the long-vector of shared perimeter length, and, finally,. Figure 5 is the result obtained with inter-cluster edge distance.

There are a number of features in the plots that deserve comment. First, we consider the structure of the view-spaces obtained using PCA and MDS. These are rather different. In the case of PCA, a cluster structure emerges. By contrast, in MDS the different views execute smooth trajectories. Hence, the output of PCA would appear to be best for locating clusters of similar views, while MDS provides information which might be more useful in constructing parametric eigenspaces. For PCA, the best trajectories are obtained with the vectors of adjacency matrix eigenvalues and shared perimeter length. For MDS, the best clusters are obtained using the inter-cluster edge distance. This feature also emerges from the plots of pairwise distance, where a clear block structure is seen in Figure 5. These blocks reflect the event structure noted above.

6 Conclusions

In this paper we have investigated how vectors of graph-spectral attributes can be used for the purposes of embedding graphs in eigenspaces. The best view trajectories result when we apply MDS to the vectors of leading eigenvalues or the shared perimeter attribute. The best clusters result when we use the inter-cluster edge distance.

Hence, we have shown how to cluster purely symbolic graphs using simple spectral attributes. The graphs studied in our analysis are of different size, and we do not need to locate correspondences. Our future plans involve studying in more detail the structure of the pattern-spaces resulting from our spectral features. Here we intend to investigate the use of ICA as an alternative to PCA as a means of embedding the graphs in a pattern-space. We also intend to study how support vector machines and the EM algorithm can be used to learn the structure of the pattern spaces. Finally, we intend to investigate whether the spectral attributes studied here

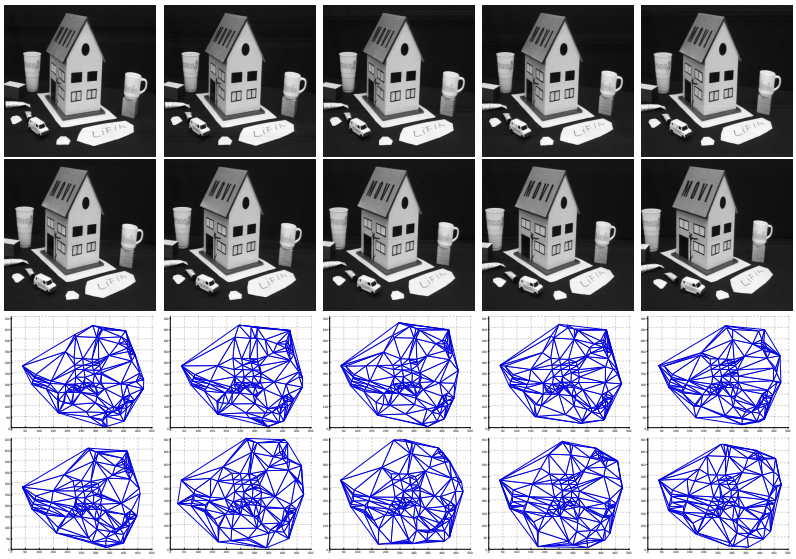


Figure 1: MOVI sequence and graphs

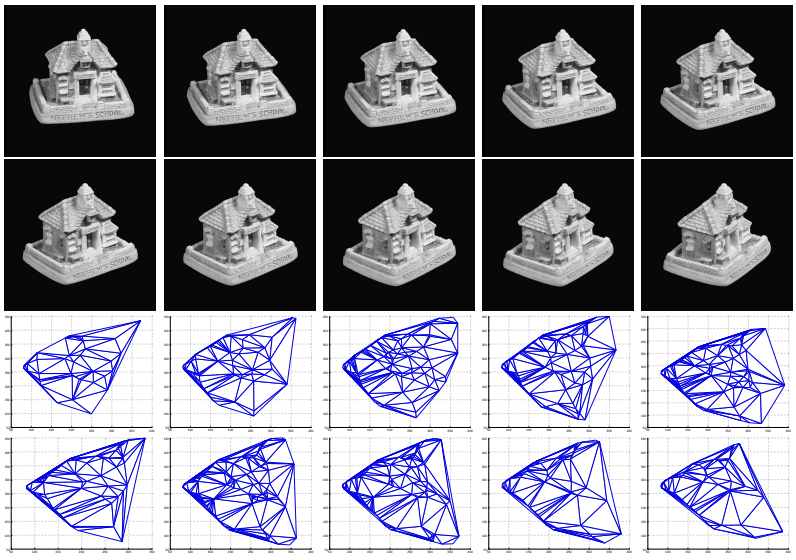


Figure 2: The chalet sequence

can be used for the purposes of organising large image data-bases.

References

[1] H.G. Barrow and R.M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *IPL*, 4:83–84, 1976.

[2] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible illumination conditions. *International Journal of Computer Vision*, 28(3):245–260, July 1998.

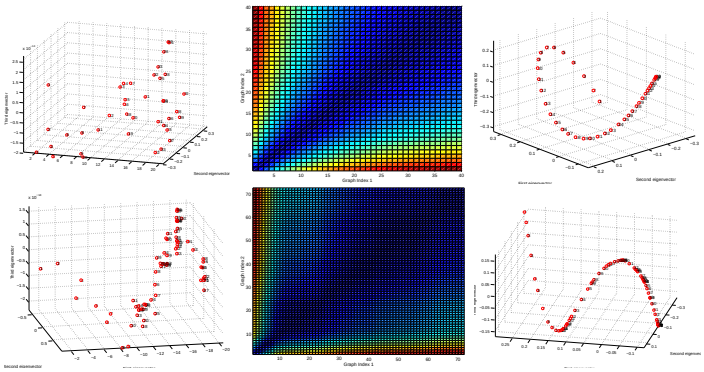


Figure 3: Graph spectra

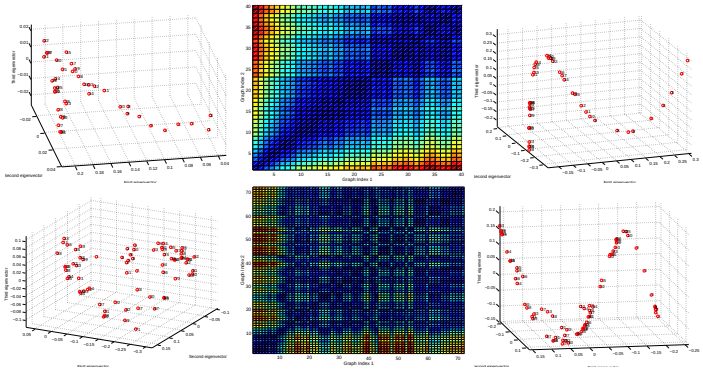


Figure 4: Shared perimeter

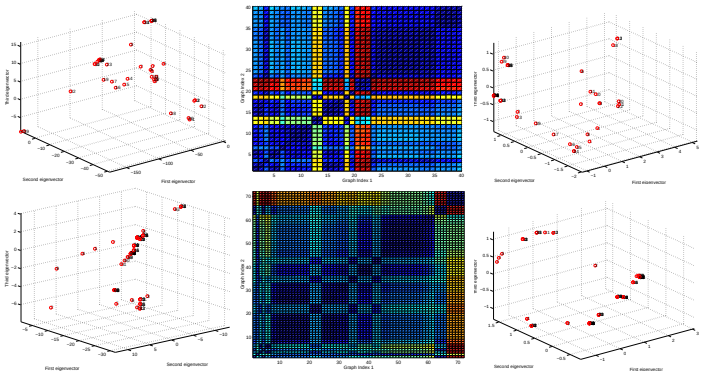


Figure 5: Cluster distances

[3] I. Biederman. Geon based object recognition. In *BMVC93*, page xx, 1993.

[4] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922, 1999.

[5] W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.

[6] F.R.K. Chung. *Spectral Graph Theory*. American Mathmatical Society Ed., CBMS series 92, 1997.

[7] M.S. Costa and L.G. Shapiro. 3d object recognition and pose with relational indexing. *Computer Vision and Image Understanding*, 79(3):364–407, September 2000.

[8] C.M. Cyr and B.B. Kimia. 3d object recognition using shape similarity-based aspect graph. In *ICCV01*, pages I: 254–261, 2001.

[9] M. Fischler and R. Elschlager. The representation and matching of pictorical structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[10] Z. Gigus and J. Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):113–122, 1990.

[11] B. Huet and E.R. Hancock. Fuzzy relational distance for large-scale object recognition. In *CVPR98*, pages 138–143, 1998.

[12] Kruskal J.B. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[13] D.J. Kriegman and J. Ponce. Computing exact aspect graphs of curved objects: Solids of revolution. *International Journal of Computer Vision*, 5(2):119–135, November 1990.

[14] Inoue K. and Urahama K. Sequential fuzzy cluster extraction by a graph spectral method. *PRL*, 20(7):699–705, 1999.

[15] H. Murase and S.K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, 1994.

[16] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions Systems, Man and Cybernetics*, 13(3):353–362, May 1983.

[17] S. Sarkar and K.L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *CVIU*, 71(1):110–136, July 1998.

[18] K. Sengupta and K.L. Boyer. Organizing large structural modelbases. *PAMI*, 17(4):321–332, April 1995.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, August 2000.

[20] Andrea Torsello and Edwin R. Hancock. Efficiently computing weighted tree edit distance using relaxation labeling. *Lecture Notes in Computer Science*, 2134:438–453, 2001.

[21] R. Wang and H. Freeman. Object recognition based on characteristic view classes. *Proc. ICPR*, I:8–12, 1990.

[22] R.C. Wilson and E.R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648, June 1997.

[23] Torgerson W.S. Multidimensional scaling. i. theory and method. *Psychometrika*, 17:401–419, 1952.

Reduction Factors of Pyramids on Undirected and Directed Graphs *

Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, W. G. Kropatsch

Pattern Recognition and Image Processing Group 183/2

Institute for Computer Aided Automation, Vienna University of Technology

Favoritenstr. 9, A-1040 Vienna, Austria

phone ++43-(0)1-58801-18351, fax ++43-(0)-1-58801-1839

e-mail:{yll, glz, saib, langs, krw}@prip.tuwien.ac.at

Abstract

We present two new methods to determine contraction kernels for the construction of graph pyramids. The first method is restricted to undirected graphs and yields a reduction factor of at least 2.0. This means that with our method the number of vertices in the subgraph induced by any set of contractible edges is reduced to half or less by a single parallel contraction. Our second method also works for directed graphs. In case of stochastic pyramids, the second method yields even higher reduction factors than the first one in all our tests.

1 Introduction

In a regular image pyramid (for an overview see [9]) the number of pixels at any level l, is r times higher than the number of pixels at the next reduced level $l+1$. The reduction factor r is greater than one and it is the same for all levels l. If s denotes the number of pixels in an image I, the number of new levels on top of I amounts to $log_r(s)$. Thus, the regular image pyramid may be an efficient structure to access image objects in a top-down process.

However, regular image pyramids are confined to globally defined sampling grids and lack shift invariance [1]. In [10] it was shown how these drawbacks can be avoided by irregular (stochastic) image pyramids. Each level represents a partition of the pixel set into cells, i.e. subsets of 4-connected pixels. The construction of an irregular image pyramid is iteratively local [10] [6]:

- The cells have no information about their global position.
- The cells are connected only to (direct) neighbors.

*This paper has been supported by the Austrian Science Fund under grants P14445-MAT and P14662-INF

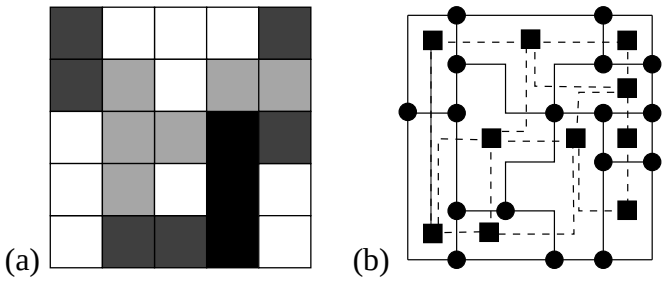


Figure 1: (a) Partition of pixel set into cells. (b) Representation of the cells and their neighborhood relations by a dual pair $(\overline{G}, G)$ of plane graphs. $\overline{G}$ has square vertices and dashed edges. G has circular vertices and solid edges.

- The cells cannot distinguish the spatial positions of the neighbors.

On the base level (level 0) of an irregular image pyramid the cells represent single pixels and the neighborhood of the cells is defined by the 4-connectivity of the pixels. A cell on level $l+1$ is a union of neighboring cells on level l. Two cells c_1 and c_2 are neighbors if there exist pixels p_1 in c_1 and p_2 in c_2 such that p_1 and p_2 are 4-neighbors (Figure 1a,b). We assume that any two successive levels are different, i.e. that at least two neighboring cells in the lower level have been united. In particular, there exists a highest level h . Furthermore, we restrict ourselves to irregular pyramids with an apex, i.e. level h contains only one cell.

In this paper we will represent the levels as dual pairs $(\overline{G_l}, G_l)$ of plane graphs $\overline{G_l}$ and G_l. The vertices of $\overline{G_l}$ represent the cells on level l and the edges of $\overline{G_l}$ represent the neighborhood relations of the cells on level l (Figure 1b). The edges of G_l represent the borders of the cells on level l, possibly including so called pseudo edges needed to represent neighborhood relations to cells enclosed by other cells. Finally, the vertices of G_l represent meeting points of at least three edges from G_l. The sequence $(\overline{G_l}, G_l)$, $0 \leq l \leq h$ is called graph pyramid. The plan of the paper is as follows. In Section 2 we will give the main idea of the stochastic pyramid algorithm and in Section 2.1 we will see that graph pyramids from maximal independent vertex sets may have a very poor reduction factor (arbitrarily close to 1.0). Moreover, experiments show that poor reduction factors are likely, especially when the images are large. We propose two modifications. The one in Section 3 guarantees a reduction factor of 2.0, but is applicable only if the edges may be contracted in both directions. The modification proposed in Section 4 also works in case of constraints on the directions. This modification yields the highest reduction factors in the case of stochastic graph pyramids, in all our tests.

2 Maximal Independent Vertex Set

In the following the iterated local construction of the (stochastic) irregular image pyramid in [10] is described in the language of graph pyramids. The main idea is to first calculate a so called *maximal independent vertex set* [3]. Let the vertex set and edge set of of $\overline{G_l}$ be denoted by $\overline{V_l}$ and $\overline{E_l}$, respectively. The incidence relation of $\overline{V_l}$, denoted by $\overline{\iota_l}(\cdot)$ maps each edge

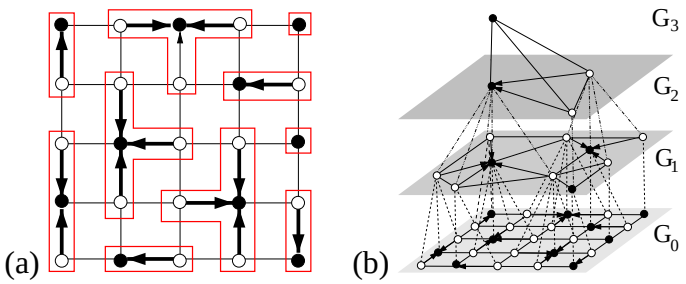


Figure 2: (a) The black vertices form a maximal independent vertex set. The frames indicate a corresponding collection of contraction kernels. (b) A graph pyramid from maximal independent vertex sets.

from $\overline{E_l}$ to its set of end vertices. The neighborhood $\Gamma_l(\overline{v})$ of a vertex $\overline{v} \in \overline{V_l}$ is defined by

$$\Gamma_l(\overline{v}) = \{\overline{v}\} \cup \{\overline{w} \in \overline{V_l} \mid \exists \overline{e} \in \overline{E_l} \text{ such that } \overline{v}, \overline{w} \in \overline{\iota_l}(\overline{e})\}.$$

A subset $\overline{W_l}$ of $\overline{V_l}$ is called maximal independent vertex set if:

1. $\overline{w_1} \notin \Gamma_l(\overline{w_2})$ for all $\overline{w_1}, \overline{w_2} \in \overline{W_l}$,
2. for all $\overline{v} \in \overline{V_l}$ there exists $\overline{w} \in \overline{W_l}$ such that $\overline{v} \in \Gamma_l(\overline{w})$.

An example of a maximal independent vertex set is shown in Figure 2a. Maximal independent vertex set (MIS) [10] [11] may be generated as follows.

MIS Algorithm:

1. Mark every element of $\overline{V_l}$ as *candidate*.
2. Iterate the following two steps as long as there are candidates.
   (a) Assign random numbers to the candidates of $\overline{V_l}$.
   (b) Determine the candidates whose random numbers are greater than the random numbers of all neighboring candidates and mark them as *member* (of the maximal independent set) and as *non-candidate*. Also mark every neighbor of every new member as *non-candidate*.
3. In each neighborhood of a vertex that is not a member there will now be a member. Let each non-member choose its neighboring member, say the one with the maximal random number (we assume that no two random numbers are equal).

The assignment of the non-members to their members determine a collection of *contraction kernels*: each non-member is contracted towards its member and all contractions can be done in a single parallel step. In Figure 2a the contractions are indicated by arrows. A graph pyramid from maximal independent vertex sets can be seen in Figure 2b. Note that we remove parallel edges and self-loops that emerge from the contractions, if they are not needed to encode inclusion of regions by other regions (in the example of Figure 2b we do not need loops nor parallel edges). This can be done by dual graph contraction [7].

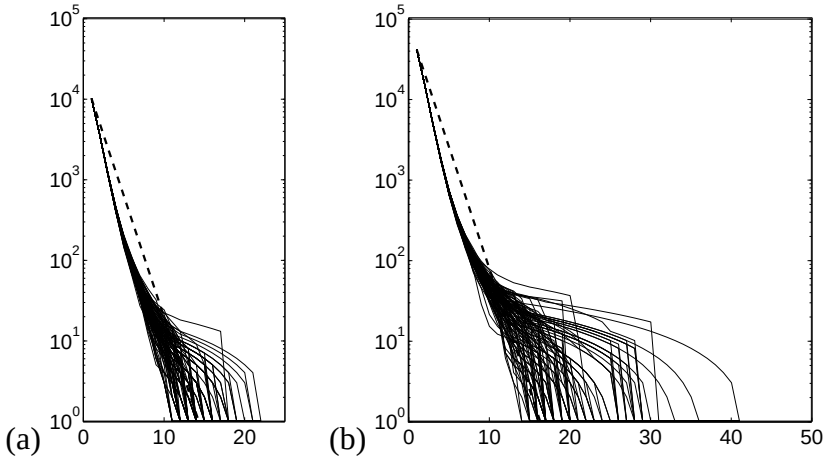


Figure 3: MIS Algorithm: Number of vertices (y-axis) for the graph of size (a) 100×100, and (b) 200×200. x-axis: levels of the graph pyramid. The slope of the lines depicts the reduction factor. Solid lines for test results and dashed line for reduction factor 2.0.

2.1 Experiments with Maximal Independent Vertex Sets

Uniformly distributed random (u.d) values are assigned to the vertices in the base level graphs. We generated 1000 graphs, on top of which we built stochastic graph pyramids. In our experiments, Section 2.1, Section 3.1 and Section 4.1, we used graphs of size 10000 and 40000 vertices, which correspond to image sizes of 100×100 and 200×200 pixels, respectively. Solid lines in Figure 3, 6 and 9 depict the first 100 of 1000 tests. Data in Table 1 were derived using graphs of size 200×200 vertices with 1000 experiments.

The numbers of levels needed to reduce the graph at the base level (level 0) to a graph consisting of a single vertex (top of the pyramid) are given in Figure 3 (a),(b). From Figure 3 we see that the height of the pyramid cannot be guaranteed to be logarithmic, except for some good cases. In the worst case the pyramid had 22 levels for 100×100 vertices and 41 levels for the graph with 200×200 vertices, respectively. Poor reduction factors are likely, as can be seen in Figure 3, especially when the images are large. This is due to the evolution of larger and larger variations between the vertex degrees in the contracted graphs (Table 1). The absolute maximum in-degree was 148. The $apriori$ probability of a vertex being the local maximum is dependent of its neighborhood. The larger the neighborhood the smaller is the $apriori$ probability that a vertex will survive. The number of iterations necessary to complete the maximum independent set per level (iterations for correction [10]) are the same as reported by [10].

To summarize, a constant reduction factor higher then 1.0 cannot be guaranteed and bad cases have a high probability, as can be seen in Figure 3.

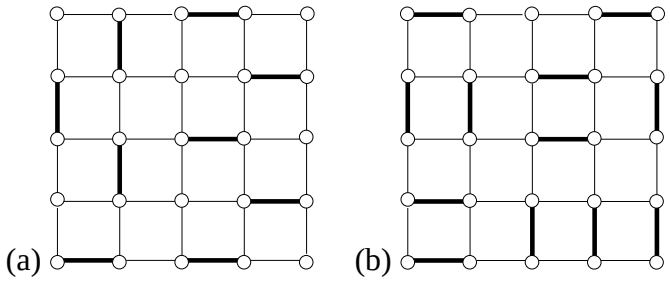


Figure 4: (a) A maximal matching. (b) A matching with more edges than in (a).

3 How to guarantee a Reduction Factor of 2.0

In the following we aim at a collection $\mathcal{C}$ of contraction kernels in a plane graph $\overline{G}$ such that

- each vertex of $\overline{G}$ is contained in exactly one kernel of $\mathcal{C}$, and
- each kernel $\mathcal{C}$ contains at least two vertices.

We assume that $\overline{G}$ is connected. Clearly, the contraction of all kernels in $\mathcal{C}$ will reduce the number of vertices to half or less. In contrast to [10] we start with independent **edge** sets or *matchings*, i.e. edge sets in which no pair of edges has a common end vertex. The selection of $\mathcal{C}$ is done in three steps.

MIES Algorithm:

1. A maximal matching M of edges from $\overline{G}$ is determined.
2. M is enlarged to a set M^+ that induces a spanning subgraph of $\overline{G}$.
3. M^+ is reduced to $\mathcal{C}$.

In the first step, a maximal matching may be determined by a iteratively local process as specified in the Section 2. Note that a maximal matching of $\overline{G}$ is equivalent to a maximal independent vertex set on the edge graph of $\overline{G}$ [4]. Since M is only required to be maximal, the edge set M cannot be enlarged by another edge from $\overline{G}$ without loosing independence. As can be seen in Figure 4(a), a maximal matching M is not necessarily maximum: there may be a matching M' that contains more edges than M.

The collection of contraction kernels defined by a maximal matching M may include kernels with a single vertex. Let v denote such an isolated vertex (isolated from M) and choose a non-self-loop e that has v as an end vertex. Since M is maximal, the end vertex $w \neq v$ of e belongs to an edge that is contained in the matching. Let M^+ denote the set of edges that are in M or that are chosen to connect isolated vertices to M (the second step of MIES). The subgraph of $\overline{G}$ that is induced by M^+ spans $\overline{G}$ and its connected components are trees of depth one or two (Figure 5(a)). A tree of depth two can be separated into two trees of depth one each by removing the unique edge, both end vertices if which belong to other edges of the tree (Figure 5(b)) (the third step of MIES). Still, each vertex of $\overline{G}$ belongs to a tree (of depth one). The arrows in

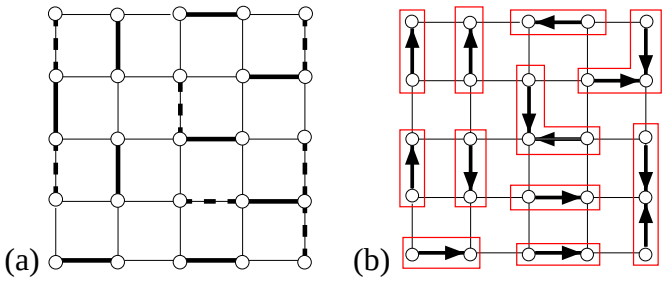


Figure 5: (a) The matching from Figure 4a enlarged by connecting formerly isolated vertices to the maximal matching. (b) After breaking up trees of depth two into trees of depth one. The arrows indicate possible directions of the contractions.

Figure 5b indicate possible directions of contractions. Note that in case of kernels with more than one edge the directions within the kernel cannot be chosen independently of one another. This is why the proposed method cannot be extended to applications in which there are a priori constraints on the directions of the contractions. However, the proposed method works for the stochastic case (no preconditions on edges to be contracted) and for connected component analysis, where the attributes of the end vertices are required to be identical.

3.1 Experiments with Maximal Independent Edge Sets

The numbers of levels needed to reduce the graph at the base level to a graph consisting of a single vertex are shown in Figure 6 (a),(b). The experiments show that the reduction factor, even in the worst case, is always bigger than the theoretical lower bound 2.0, indicated by the

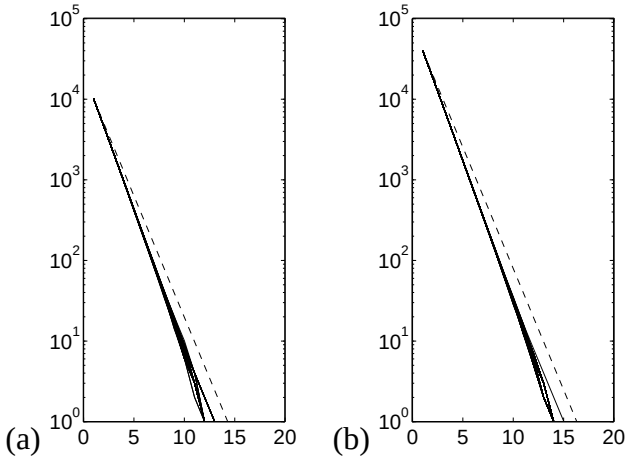


Figure 6: MIES Algorithm: Number of vertices (y-axis) for the graph of size (a) 100×100, and (b) 200×200. x-axis: number of levels. The slope of the lines depicts the reduction factor. Solid lines for test results and dashed line for reduction factor 2.0.

dashed line in Figure 6. This method is more stable than MIS. As can be seen in Figure 6, the variance of the slope is smaller then in case of MIS (Figure 3). The mean number of iteration for correction per level was higher for MIES (Table 1).

4 Constraints on the Directions of the Contractions

In many graph pyramid applications such as line image analysis [2, 8] and the description of image structure [5] a directed edge e with source u and target $v \neq u$ must be contracted (from u to v), only if the attributes of e, u, and v fulfill a certain condition. In particular, the condition depends on u being the source and v being the target. The edges that fulfill the condition are called *preselected* edges. From now on the plane graphs in the pyramid have directed edges. Typically, the edges in the base level of the pyramid form pairs of reverse edges, i.e. for each edge e with source u and target v there exists an edge e' with source v and target u. However, the set of preselected edges may contain e without containing e'. The goal is to build contraction kernels with a "high" reduction factor from the set of preselected edges. The reduction will always be determined according to the directed graph induced by the preselected edges. For example, if the number of vertices in the induced subgraph is reduced to half, the reduction factor will be 2.0. From the example in Figure 7a it is clear that, in general, no reduction factor larger than 1.0 can be guaranteed. We require that the contraction kernels are vertex disjoint rooted trees of depth one or zero (single vertices), each edge of which is directed towards the root. A set $\overline{C}$ of directed edges forms such a collection of contraction kernels if and only if $\overline{C}$ contains none of the edge pairs depicted in Figure 7b. Seen from a directed edge e with source u and target $v \neq u$ that one wants to contract (from u to v), no edge $e' \neq e$ with end vertex (source or target) equal to u or source equal to v may be contracted. An edge e together with those edges that one may not contract if e is contracted form a neighborhood $N(e)$ of e. Figure 8a depicts $N(e)$ in case of u and v both having 4 neighbors. To find a **maximal** (independent) set of directed edges (MIDES) forming vertex disjoint rooted trees of depth zero or one, we proceed analogously to the generation of maximal independent vertex sets, as explained in the Section 2. Let $\overline{E_l}$ denote the set of directed edges in the graph $\overline{G_l}$ of the graph pyramid. We proceed as follows.

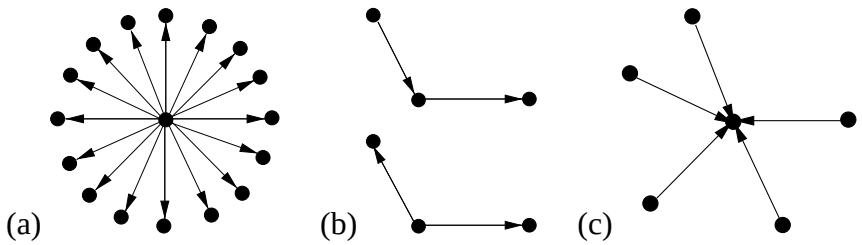


Figure 7: (a) The reduction factor of a star with n edges pointing away from the center is $(n+1)/n$. (b) Forbidden pairs of directed edges. (c) A legal configuration of directed edges

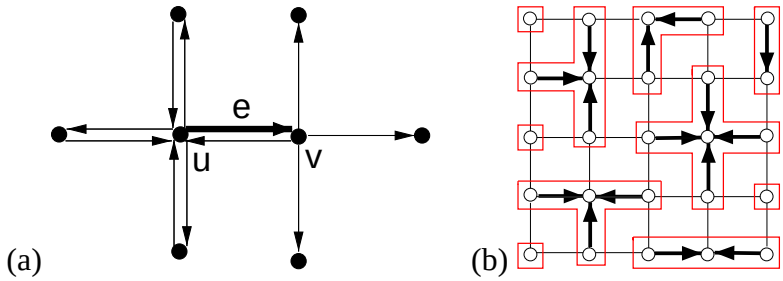


Figure 8: (a) The neighborhood $N(e)$. (b) Maximal independent edge set with respect to $N(e)$.

Process	$\mu(max)$	$\sigma(max)$	$\mu(\#iterations)$	$\sigma(\#iterations)$
MIS	70.69	23.88	2.95	0.81
MIES	11.74	0.71	4.06	1.17
MIDES	13.29	1.06	2.82	1.07

Table 1: Mean μ and standard deviation σ of maximum vertex degrees of the pyramids; Mean μ and standard deviation σ of number of iterations to complete maximum independent set per level of the pyramid.

MIDES Algorithm:

1. Mark every directed edge of $\overline{E_l}$ as *candidate*.
2. Iterate the following two steps as long as there are candidates.
   (a) Assign random numbers to the candidates.
   (b) Determine the candidates $\overline{e}$ whose random numbers are higher (larger) than the random numbers in $N(\overline{e}) \setminus \{\overline{e}\}$ and mark them as *member* (of a contraction kernel). Also mark every $\overline{e}' \in N(\overline{e})$ of every new member $\overline{e}$ as *non-candidate*.

4.1 Experiments with Maximal Independent Directed Edge Sets

Pictures in Figure 9 show the number of levels required to get on top of the pyramid. We see that the reduction factor is better than 2.0 (dashed line) even in the worst case. Also the in-degrees of the vertices is much smaller (13.29) than for MIS (70.69). For the case of the graph with size 200×200 vertices, MIDES needed 13 levels in comparison to 15 levels in the worst case of MIES. The number of iterations needed to complete the maximum independent set was comparable with the one of MIS (Table 1). The MIDES algorithm shows a better reduction factor than MIES, as can be seen in Figure 9.

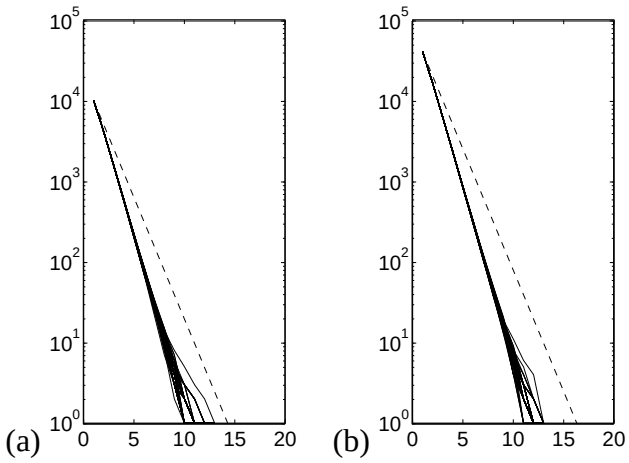


Figure 9: MIDES Algorithm: Number of vertices (y-axis) for the graph of size (a) 100×100, and (b) 200×200. x-axis: number of levels. The slope of the lines depicts the reduction factor. Solid lines for test results and dashed line for reduction factor 2.0

5 Conclusion

Experiments with stochastic decimation using maximal independent vertex sets (MIS) showed a problematic behavior on large images. After an initial phase of strong reduction, the reduction decreases dramatically. This is due to the evolution of larger and larger variations between the vertex degrees in the contracted graphs. To overcome this problem we proposed a method, MIES, based on matchings which guarantees a reduction factor of 2.0. As in the case of independent vertex sets, the method based on matchings does not allow to control the directions of the contractions. The second method, MIDES, that we proposed and tested is based on directed edges and allows to control the directions of the contractions. The experiments showed a non-decreasing reduction that was even stronger than the one obtained from the method based on matchings. Future work will focus on understanding and proving the good performance of the method based on directed edges.

References

[1] M. Bister, J. Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9):605–617, 1990.

[2] Mark J. Burge and Walter G. Kropatsch. A minimal line property preserving representation of line images. *Computing*, 62:355 – 368, 1999.

[3] N. Christofides. *Graph theory - an algorithmic approach*. Academic Press, New York, 1975.

[4] Reinhard Diestel. *Graph Theory*. Springer, New York, 1997.

[5] Roland Glantz and Walter G. Kropatsch. Guided relinking of graph pyramids. In *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'2000 and SPR'2000*, Lecture Notes in Computer Science, Alicante, Spain, August 1999. Springer, Berlin Heidelberg, New York. submitted.

[6] Jean-Michel Jolion. Data driven decimation of graphs. *Proc. 3th IAPR Int. Workshop on Graph based Representation*. pages 105–114, Capri, Italy, 2001.

[7] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366 – 374, 1995.

[8] Walter G. Kropatsch and Mark Burge. Minimizing the Topological Structure of Line Images. In Adnan Amin, Dov Dori, Pavel Pudil, and Herbert Freeman, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR'98 and SPR'98*, volume Vol. 1451 of *Lecture Notes in Computer Science*, pages 149–158, Sydney, Australia, August 1998. Springer, Berlin Heidelberg, New York.

[9] Walter G. Kropatsch, Aleš Leonardis, and Horst Bischof. Hierarchical, Adaptive and Robust Methods for Image Understanding. *Surveys on Mathematics for Industry*, No.9:1–47, 1999.

[10] Peter Meer. Stochastic image pyramids. *CVGIP*, 45:269 – 294, 1989.

[11] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tesselations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):pp.307–316, April 1991.

Feature Extraction using an Iterative Scheme within a Hierarchical Framework

M. Melki and J.M. Jolion

Laboratoire Reconnaissance de Formes et Vision

Bât. J. Verne, INSA Lyon, 69621 Villeurbanne Cedex, France

tel: 33 4 72 43 87 59 / fax: 33 4 72 43 80 97

e-mail: {melki,jolion}@rfv.insa-lyon.fr

Abstract

This paper introduces the main principles of a novel approach for feature extraction (mainly symbolic) within a hierarchical graph based framework. This is just a preliminary work and emphasize is made on the formalism, the properties we would like to achieve and some *ad-hoc* illustrations.

1 Introduction

As computer power increases, researchers use more complex tools for Pattern Recognition and Image Analysis. Graph theory is such a framework. Along the overall processes from a stimulus to its interpretation, graphs are used for several distinct tasks: hierarchical graphs for image segmentation and for control of perceptual strategies, graph matching for recognition and image understanding, graph manipulation for clustering, conceptual graphs for representation of relational and structural knowledge, involving time explicitly represented in the graphs.

For instance, many works have already been done over the past few years showing that graphs are very well suited for image segmentation and more generally for image analysis. Graphs are efficient as a processing and representational scheme in pattern recognition and image processing when complex and irregularly sampled data need to be synthesized

It takes its basis in the challenge of the TC15 technical committee of IAPR and more particularly in the ”Increasing the intelligence of a pixel-based graph” challenge. The underlying goal is to provide the graph based techniques with some learning and adaptation properties like the neural networks.

Our goal in this study is to propose a novel framework This is a very preliminary work. We do not attempt to present a complete framework.

Thus the paper is organized as follows: in section 2, we recall the main principles of feature extraction in images using hierarchical graphs. Section 3 gives more details of the kind of symbols we intend to work with. Section 4 introduces the main steps of hierarchical scheme.

Section 5 will then focus on the adaptation scheme within this framework. Section 6 draws some trends toward hierarchical symbolic features by means of a simple illustration.

2 Hierarchical graphs: a short recall

A hierarchical graph is a set of graphs built from the base up to the apex using a decimation process.

Let $G = (N, E)$ be a graph where N stands for the set of nodes and E for the set of edges. Let $\delta_{ij} \stackrel{\text{def}}{=} (i, j) \in E$. Let $V(i)$ be the neighborhood of the node i defined as $\{j \in N : \delta_{ij}\}$. Note that we assume that a given node i is a not member of its neighborhood. Each node, i, is associated with a value x_i and a set of local information. In the following, we will consider the edges of this graph as not valuated.

A decimation process transforms the graph $G^{(k)}$ in $G^{(k+1)}$ such that $|N^{(k+1)|} < |N^{(k)}|$.

P. Meer proposed in [10] to constraint this process with two rules.

Rule 1 : The decimation must be maximal, *i.e.* $\forall (i, j) \in E^{(k)}$, i and j cannot both belong to $N^{(k+1)}$.

Rule 2 : Any node of $N^{(k)}$ must be linked to a node in $N^{(k+1)}$, *i.e.* $\forall i \in N^{(k)}, (i \in N^{k+1} \vee V(i) \bigcap N^{(k+1)} \neq \emptyset)$.

It can be shown that finding $G^{(k+1)}$ with these two rules is equivalent to find the stable of $G^{(k)}$. Several powerful algorithms have been proposed to solve this search.

This tool has already been widely used in image analysis for feature extraction [8, 6, 10].

Different contraction kernels or decimation functions yield different hierarchies of graphs. Our goal is to interact within a given hierarchy, locally adapting the decimation function to some *a priori* long term goal.

3 Some symbolic features

We are here concerned with symbolic features. Given an image made of numerical values, we first need to extract symbolic tokens as relevant as possible. For instance if the initial stimulus is a curve, one can use the n-cell description language [9].

The 3×3 masks introduced by Canning *et al.* [3] are also such tokens. Figure 1 shows an example on the so-called ”cul de sac” image. The mask related to a part of this image (an house) is shown. A given pixel can be associated to several (between one and eight) masks depending on the local distribution of gray levels. Any mask is a symbolic token valuated by the local contrast. In order to be used in a hierarchical scheme, one need a description language which mix these elements in order to exhibit more and more complex symbolic features. The 3×3 masks can be merged in order to define other symbolic features of more complex nature like edge segments, sharp corners, large corners, concavities and enclosures as shown in [7] as well as texture patterns as shown in [4]. A more complex structure is derived from low-level structures. Our goal is to set up this strategy in a hierarchical process.

4 Hierarchical graph of symbolic features

The construction of a hierarchy is controlled by two processes: the selection of the surviving nodes which will constitute the next level, and the reduction which set how the nodes of the current level decide to link with the surviving nodes. These two processes are clearly some kind of degrees of freedom for any algorithm working on the hierarchy. The selection is most of the time simply a local maxima search and a widely used reduction rule is the nearest neighbor rule.

For a given application, these rules are set and not adaptable to the result of the feature extraction. This is the point we are working on.

Let us first reformulate the construction of the hierarchy in a more general and especially more symbolic framework.
Let n by a node of $G^{(k)}$. We assume that n is associated some symbolic descriptions (for instance parts of a curve) $S_0(n), S_1(n) \dots S_{l(n)}(n)$ and a set of rules $R_0, R_1 \dots R_\alpha$ [1].
Step 1 : Hypotheses generation

n asks its neighbors to get their description. Then based on these information and its set of rules, it builds some new symbols involving some of its neighbor's symbols. Any new symbol is associated some characteristics and at least a confidence value. So any new symbol will be considered as a valuated hypothesis, $\mu(H_0(n)), \mu(H_1(n)) \dots \mu(H_{h(n)}(n))$ where μ stands for the confidence measure function, H_i for the new symbol, *i.e.* up to now being an hypothesis, and $h(n)$ for the number of hypotheses. The confidence is related to the initial confidence of the symbols and to the hypothesis generation itself.

Even though dual-graph contractions preserve topology, our approach is not necessarily based on dual-graph contractions. Hence, the neighborhood relationship of higher level graphs does not always have the same meaning as the spatial neighborhood of the receptive fields. More, the receptive field associated to any node, i.e. the set of pixels in the initial image which belong to this node, is not guaranteed to consist of one connected component. Each symbol must thus carry some information related to its spatial location which is most of the time of importance for the generation of the new hypotheses.
Step 2 : Selection

As the selection of the surviving node is based on the local maxima of the value x_n, we must choose among several strategies:

- Best first : Any node only keeps its best hypothesis. $x_n = Max\{\mu(H_i(n)) : i = 1, \dots, h(n)\}$.
- Diversity first : the more hypotheses a node has, the better it is. $x_n = \sum_{i=1}^{i=h(n)} \mu(H_i(n))$ or $x_n = h(n)$.
- Goal-directed first : the system has a goal, $\mathcal{G}$, and the node keeps only the best hypothesis related to this goal. $x_n = Max\{\mu(H_i(n)) \times \rho(H_i(n), \mathcal{G}) : i = 1, \dots, h(n)\}$ where $\rho()$ stands for the plausibility of an hypothesis for a given goal. For instance, we can look for crosses or linear features ... Note that if we assume a probabilistic framework for

[1] Note that these rules may be some kind of common knowledge for the nodes and may be not dependent on the node.

the confidence μ and the plausibility ρ, then we can set $\mu(H_i(n)) \times \rho(H_i(n)$ as $P(H_i) \times P(\mathcal{G}|H_i) = P(H_i, \mathcal{G})$ (we estimate the plausibility by some *a posteriori* likelihood)

Step 3 : Reduction

The reduction scheme is derived from the retained hypotheses associated to the surviving nodes, e.g. local maxima of x_n satisfying the two Meer's rules.

A non surviving node in $G^{(k)}$ is considered as a child of a node of $G^{(k+1)}$ if and only if one of the symbols of this node was involved in the determination of a symbol of the surviving node. Note that as in the adaptive pyramid, a non surviving node is not forced to join at least one of the surviving node in its neighborhood if it considers that no hypothesis relates to it. Based on its local confidence, this node will disappear or be artificially kept up to the apex, being considered as a final step of a feature extraction.

One of the question we have to answer is to define the set of symbols that is associated to this node among the following alternatives:

- the symbols associated to this particular node at the previous level;
- a compromise of the symbols associated to the non surviving nodes.

Another question is related to the symbol we want to extract. Indeed, we cannot assume that the spatial distribution of these symbols is dense. This results in two disadvantages. First, the iterative process proposed by P. Meer in order to extract the surviving nodes is convergent if and only if we can extract new surviving nodes during each new iteration. If the spatial density of symbols is sparse, one can obtain a subgraph without any hypothesis (a subgraph such that any node is associated a null value). In order to overcome this problem, a global control is required in order to detect such case. If so, we propose to assign a random value to the node of the subgraph. The selection process can thus go on and the two rules will be satisfied. However, this particular class of surviving nodes can of course participate to the next selection step but must not perturb it. That is why they will be assigned a "dummy symbol".

Second, a non surviving node may not find a surviving node in its neighborhood with a non null intersection between its list of hypotheses and the retain list. In this case, we will assign this node to the surviving node but with a "dummy link". The second rule is thus validated and the node will not interact with the retained hypotheses.

It is indeed of importance to keep these two classes of nodes in the hierarchy (an alternative would have been to remove them) because this bottom-up process is just a first step. As we assume some adaptiveness, the fact that a node will be classified as a "dummy" node may change during the overall process.

The decimation process then continues up to the apex of the hierarchy. It stops when no more hypothesis can be set from the remaining nodes in the graph.

5 Adaptation in hierarchical graphs

Up to now, the hierarchy has been presented in a bottom-up fashion. Let us now focus on the top-down part of this framework which has not been yet fully studied in the literature (except some trends in [6] for shape description).

When the bottom-up process stops, we end up with a set of hypotheses (e.g. valuated symbolic features). We assume an extern control which can classify the hypotheses regarding a goal in three classes:

- not related to the goal ;
- to be improved ;
- validating the goal.

The symbolic features related to the third classes are just kept as they are. The symbolic features of the first and second classes must be improved in order to better achieve the goal. This information is mapped down in the hierarchy. In order to take advantage of this information, the hierarchical structure must be able to adapt itself. Which parts can be adapted ?

First, the plausibility of any local hypothesis which survives during the first bottom-up extraction can be decreased if they are related to a final hypothesis of the first class. On the contrary, hypothesis which are related to a final hypothesis of the second and third classes can be increased in order to better pop out. This kind of process is similar to the relinking pyramid introduced by Hong *et al.* in [5]. Note that Spann used a stochastic relinking procedure which could be adapted to take into account a particular configuration or behavior [11]. One can also see some similarity to the neural network approach. An equivalence between a hierarchy of graphs and a neural network has already been shown in [1] for the particular case of image pyramid. These studies showed that a Hopfield network can be used for constructing a non regular pyramid. They also showed that the curve pyramid can be implemented using a neural network. However, the inside of a neural network is mostly like a black box and only a set of weights can be adapted. In our case, we want to adapt the rules.

Second, the common set of rules can be valuated. Indeed, the sub-hierarchical graphs associated to the final symbolic feature of class two and three can be used to derive a valuation of the rules, *i.e.* a rule will be emphasized if it has been successfully used to produce a symbolic feature in accordance with the underlying goal. Note that this valuation can be level dependent.

Synchronous *versus* asynchronous top-down. Another important question we have to deal with is the exact nature of the top-down process. We could go down the hierarchy till the base and then start a new bottom-up process. However the knowledge accumulated during the previous steps would be lost. An alternative is to go down the hierarchy until one node can benefit from the upward information. Then the bottom-up process starts again.

Which tools ? The adaptation process can be viewed as an optimization problem thus some of the classic tools will be studied in order to define the more appropriate one among the MDL technique, the genetic algorithm, the classic feed-forward from neural nets . . .

6 Illustration

We will here limit ourselves to two basic primitives :

- Lines : a line is represented by the symbol $L(x_1, x_2)$, where x_1 and x_2 stand for the two end-points of the line,

- Intersection : when two lines intersect, we represent the intersection by $X(x_1, x_2, x_3, x_4)$, where (x_1, x_2) and (x_3, x_4) stand for the end-points of the two intersecting lines.

An end-point is either a pixel location (coordinates) or a link to another end-point (in that case, we assume this link to be symmetric). We will use Greek letters, for example $\{L_1(x_1, \alpha), L_2(\alpha, x_2)\}$, when two end-points are linked. So here, the second end-point of L_1 and the first end-point of L_2 are linked.

6.1 A reduction hypothesis

In this simple illustration, we will assume only one reduction hypothesis, in order to limit the complexity of the reduction process. In the following, $x_i = x_j$ is equivalent to x_i and x_j have the same coordinates. The hypotheses generation is as follows:

- Make the union of the primitives from the symbols over the neighborhood,
- Merge lines : for every two lines $L(x_1, x_2)$ and $L(x_3, x_4)$ in the list :
  - If both end-points are identical, e.g. $x_2 = x_3$ and $x_4 = x_1$, remove the two lines, and insert the closed line $L(\alpha, \alpha)$ instead,
  - If lines share an end-point, e.g. $x_2 = x_3$, remove the two lines, and insert the merged line $L(x_1, x_4)$ instead.
- Merge intersections and lines : for every intersection $X(x_1, x_2, x_3, x_4)$ and line $L(x_5, x_6)$
  - If the two end-points of the line belong to the intersection, e.g. $x_1 = x_5$ and $x_2 = x_6$, remove the line, and link the two end-points of the intersection : the intersection becomes $X(\alpha, \alpha, x_3, x_4)$,
  - If the intersection and the line share an end-point, for instance $x_1 = x_5$, remove the line, and replace x_1 by x_5 in the intersection which becomes $X(x_5, x_2, x_3, x_5)$.
- At last, merge intersections : for every two intersection primitives $X(x_1, x_2, x_3, x_4)$ and $X(x_5, x_6, x_7, x_8)$: for each end-points the two intersections share, link those two end-points (for instance, if $x_2 = x_7$, the two intersections will become $X(x_1, \alpha, x_3, x_4)$ and $X(x_5, x_6, \alpha, x_8)$).

The confident value associated with the resulting symbol will be the number of intersection primitives contained in that symbol. Note that we assume strict equivalence between end-points. One can also use some fuzzy equivalence resulting in fuzzy symbols.

6.2 An example of pyramid construction

Figure 2 represents the input image we will reduce in this part.

First, each pixel is associated a symbol consisting of just one primitive : for most pixels, it will be lines, but for the four pixels where the lines cross, it will be intersections (this information can be found locally). So, we will get something equivalent to figure 4. In the first steps,

we will only merge lines, which means that the global structure will not change, until the nodes containing intersections become neighbors (when all lines between them have been absorbed). The structure of the pyramid when this happens can be seen on figure 5.

When the nodes containing intersections become neighbors, some of them won't be able to survive (due to reduction constraints). However, the surviving nodes will keep the useful information contained by the non-surviving nodes : the symbols associated to the surviving nodes will no more be simple one-primitive symbols, but more complex multi-primitives symbols, as shown at figure 6 (where the black dots represent linked end-points).

At the next level, only one complex node survives, keeping the information from all the complex nodes of the previous level (figure 7). Note that the links between the four intersections have been correctly retrieved, and the last links will be made once the node merges with the last lines. At last, the apex of the hierarchy consists of one node, which representation is the expected representation for the figure (see figure 3).

6.3 Discussion

The previous example aimed at showing how symbolic pyramids can be used to extract *complex* relevant information, which cannot be done by using numeric pyramids.

In our example, the complexity was the number of primitives stored in each node : at lower levels, nodes have only a one-primitive symbol, whereas at higher levels, nodes might have multi-primitive symbols, where the primitives are usually linked by some of their end-points, so that they represent complex patterns in the picture. A numeric pyramid would have lost a lot of information when the nodes representing intersections would have been merged.

Note that in this simple illustration, the symbols themselves do not change when we go up into the pyramid : they are sets of primitives at every level of the pyramid (the sets become more and more complex though). In more complex models, we can have distinct symbols at the lower and higher levels (for example, with a geometric representation, we could have angles and lines in lower levels, and curves, polylines, polygons and circles in higher levels). In such case, we would also have the same evolution in the reduction rules used in the different levels.

At last, one important point is the way we can control which reduction rule is applied. In our example, we only used one reduction rule. But we could for instance add one rule to extract any closed pattern into one closed line (a line which both end-points are linked to each other). Using that rule, level $n + 2$ we would become like figure 8 (where the circle and the four lines next to it are in the same node), and the final symbol would be figure 9. One of our goals consists in learning properly how to use the information we get from a reduction to modify the importance of the rules (i.e. implement a top down process), in order to adapt the reduction regarding to the data and the problem we want to solve.

Our main approach is not related to the Brooks theory, *i.e.* intelligence without representation [2] but mostly in the Marr paradigm. Indeed, we would like to learn a set of intermediate representations and the adaptation processes needed to transform one representation into another.

The proposal introduced in this paper just set some trends toward a new method and asked more questions than it gave answers. Our goal is, in the context of this workshop, to gain from

our community some advice on this long term project.

References

[1] Bischof H. and Kropatsch W., "Neural Networks and Image Pyramids", *Pattern Recognition 1992*, edited by Bischof H. and Kropastch W., W.G.Oldenbourg, Germany, 1992, pp.249-260.

[2] Brooks R.A., "Intelligence without representation", Artificial Intelligence, 47, 1991.

[3] Canning J., Kim J.J., Netanyahu N.S. and Rosenfeld A., "Symbolic pixel labeling for curvilinear feature detection", *Pattern Recognition Letters*, Vol. 8, 1998, pp.299-308.

[4] Duperthuy C. and Jolion J.M., "Toward a generalized primal sketch", *8th Int. Workshop on Theoretical Foundations of Computer Vision*, Dagstuhl, Germany, 1996, Springer Verlag, Advances in Computing, 1997, pp.109-118.

[5] Hong T.H., Narayanan K.A., Peleg S., Rosenfeld R. and Silberberg T. "Image Smoothing and Segmentation by Multiresolution Pixel Linking: Further Experiments and Extensions", *IEEE trans. on Syst. Man, Cybern.*, Vol. 12, 1982, pp.611-622.

[6] Jolion J.M. and Montanvert A. "The adaptive pyramid: a framework for 2D image analysis", *Computer Vision, Graphics and Image Processing: Image Understanding*, Vol. 55, No 3, 1992, pp.339-348.

[7] Jolion J.M., "Concavity detection using mask-based approach", *SSPR'98, 7th IAPR Int. Workshop on Structural and Syntactical Pattern Recognition*, Sydney, Aout 1998, Lecture Notes in Computer Science, A.Amin, D.Dori, P.Pudil, H.Freeman (Eds), Springer, pp.302-311.

[8] Kropatsch W.G., "Building Irregular Pyramids by Dual Graph Contraction", *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142, No 6, 1995, pp.366-374.

[9] Kropatsch W.G., "Property Preserving Hierarchical Graph Transformations". In C. Arcelli, L.P. Cordella and G. Sanniti di Baja, editors, *Advances in Visual Form Analysis*, World Scientific Publishing Company, 1997, pp.340-349.

[10] Meer P., "Stochastic image pyramids, *Computer Vision, Graphics, and Image Processing*, Vol. 45, No 3, 1989, pp.269-294.

[11] Spann M., "Figure/Ground Separation using Stochastic Pyramid Relinking", *Pattern Recognition*, Vol. 24, 1991, pp.993-1002.

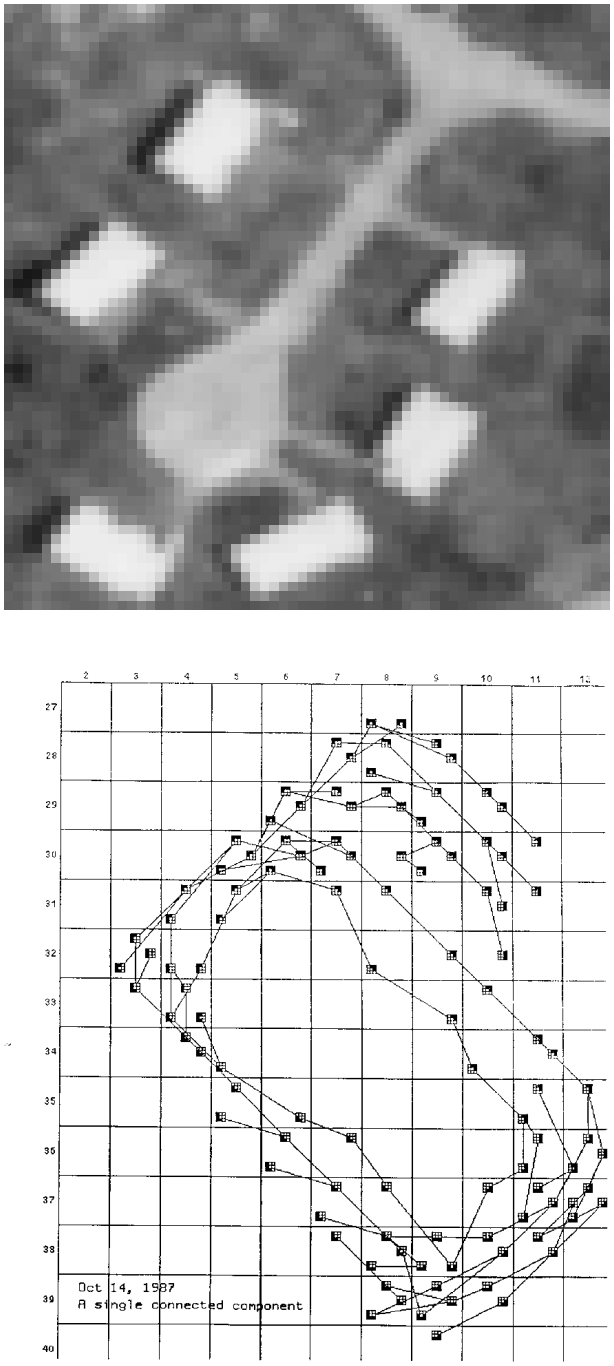


Figure 1: The ”cul de sac” aerial grey level image and a detail of the detected masks for one of the house. A given pixel is associated with several masks. The links relate the masks which are mutually consistent.

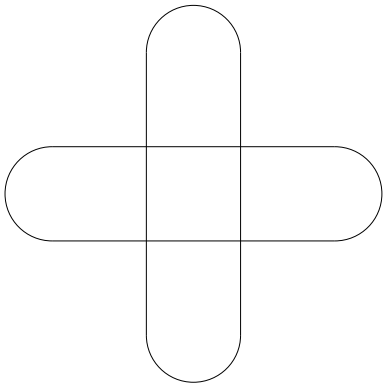


Figure 2: A test figure

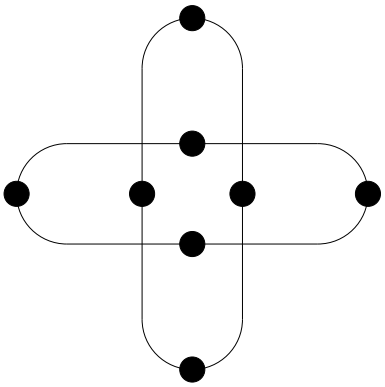


Figure 3: Its representation

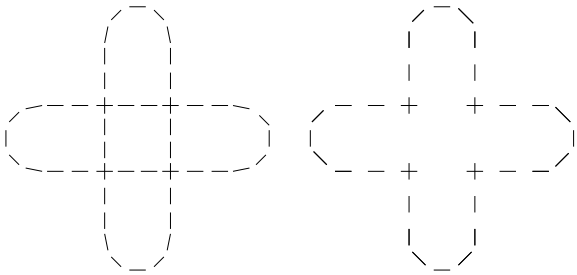


Figure 4: Level 1

Figure 5: Level n

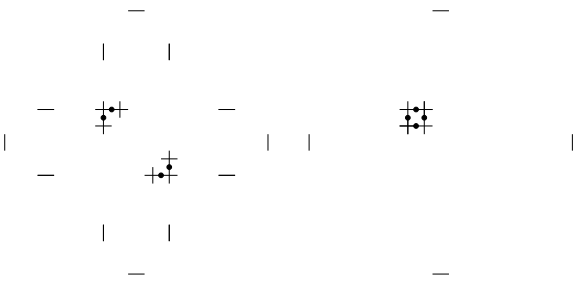


Figure 6: Level $n+1$

Figure 7: Level $n+2$

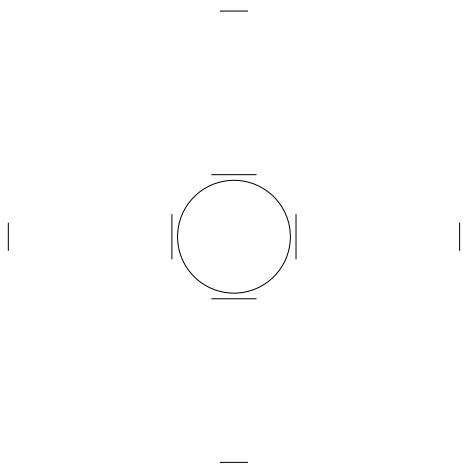


Figure 8: Level $n+2$, using closing rule

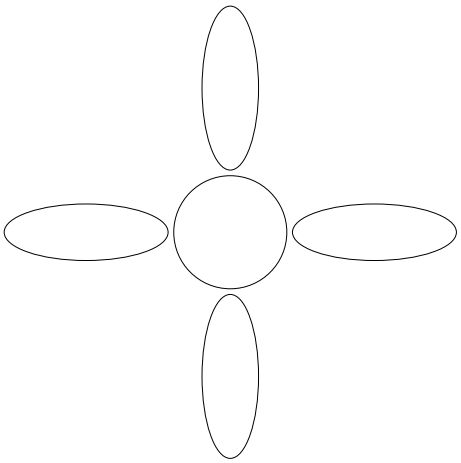


Figure 9: Final representation with closing rule

Randomized RANSAC *

Jiří Matas, Ondřej Chum
Center for Machine Perception, Faculty of Electrical Engineering
Czech Technical University, Technicka 2, Prague, Czech Republic
tel: +420 2 2435 7637, fax: +420 2 2435 7385
e-mail: [matas,chum]@cmp.felk.cvut.cz

Abstract

Many computer vision algorithms include a robust estimation step where model parameters are computed from a data set containing a significant proportion of outliers. The RANSAC algorithm is possibly the most widely used robust estimator in the field of computer vision. In the paper we show that under a broad range of conditions, RANSAC efficiency is significantly improved if *hypothesis evaluation is randomized*.

A new randomized (hypothesis evaluation) version of RANSAC, R-RANSAC, is introduced. Computational savings are achieved by typically evaluating only a fraction of data points for models contaminated with outliers. The idea is implemented in a two-step evaluation procedure. A mathematically tractable class of statistical preverification tests for test samples is introduced. For this class we derive an approximate relation for optimal setting of its single parameter. The proposed pre-test is verified on both synthetic data and real-world problems and a significant increase in speed is shown.

1 Introduction

Many computer vision algorithms include a robust estimation step where model parameters are computed from a data set containing a significant proportion of outliers. The RANSAC[1] algorithm introduced by Fishler and Bolles in 1981 [2] is possibly the most widely used robust estimator in the field of computer vision. RANSAC has been applied in the context of short baseline stereo [11, 13], wide baseline stereo matching [8, 14, 10], motion segmentation [11], mosaicing [6], detection of geometric primitives [1], robust eigenimage matching [4] and elsewhere.

The structure of the RANSAC algorithm is simple but powerful. Repeatedly, subsets are randomly selected from the input data and model parameters fitting the sample are

*The authors were supported by the European Union under project IST-2001-32184, the Czech Ministry of Education under project MSM 212300013 and by The Grant Agency of the Czech Republic under project GACR 102/02/1539.

[1] **RAN**dom **SA**mple **C**onsensus

computed. The size of the random samples is the smallest sufficient for determining model parameters. In a second step, the quality of the model parameters is evaluated on the full data set. Different cost functions may be used [12] for the evaluation, the standard being the number of inliers, i.e. the number of data points consistent with the model. The process is terminated when the likelihood of finding a better model becomes low. The strength of the method stems from the fact that it is sufficient to select a single random sample not contaminated by outliers to find a good solution. Depending on the complexity of the model (the size of random samples) RANSAC can handle contamination levels well above 50%, which is commonly assumed to be a practical limit in robust statistics [9].

The speed of RANSAC depends on two factors. Firstly, the level of contamination determines the number of random samples that have to be taken to guarantee a certain confidence in the optimality of the solution. Secondly, the time spent evaluating the quality of each of the hypothesized model parameters is proportional to the size N of the data set. Typically, a very large number of erroneous model parameters obtained from contaminated samples are evaluated. Such models are consistent with only a small fraction of the data. This observation can be exploited to significantly increase the speed of the RANSAC algorithm.

As the main contribution of this paper, we show that under a broad range of conditions, RANSAC efficiency is significantly improved if *hypothesis evaluation is randomized*. The core idea of the Randomized (hypothesis evaluation) RANSAC is that most model parameter hypotheses evaluated are influenced by outliers. For such erroneous models, it is sufficient to test only a small number of data points d from the total of N points ($d \ll N$) to conclude, with high confidence, that they do not correspond to the sought solution. The idea is implemented in a two-step evaluation procedure. First, a statistical test is performed on d randomly selected data points. The final evaluation on all N data points is carried out only if the pre-test is passed. The increase in speed of the modified RANSAC depends on the likelihoods of the two types of errors made in the pre-test: 1. rejection of an uncontaminated model and 2. acceptance of a contaminated model. Since RANSAC is already a randomized algorithm, the randomization of model evaluation does not change the nature of the solution – it is only correct with a certain probability. However, the same confidence in the solution is obtained in, on average, a shorter time.

Finding an optimal pre-test with the fastest average behaviour is naturally desirable, but very complex. Instead we introduce in Section 3 a mathematically tractable class of pre-tests based on small test samples. For this class we derive an approximate relation for optimal setting of its single parameter. The proposed pre-tests are assessed on both synthetic data and real-world problems and performance improvements are demonstrated.

The structure of this paper is as follows. First, in Section 2, the concept of evaluation with pre-tests is introduced and formulae describing the total complexity of the algorithm are derived. Both the number of samples drawn and the amount of time spent on evaluation of a hypothesized model are discussed in detail. In Section 3, the *d-out-of-d* class of pre-test is introduced and analyzed. In Section 4 both simulated and real experiments are presented and their results discussed. The paper is concluded in Section 5 and plans for future work are discussed.

2 Randomized RANSAC

In this section, the time complexity of the RANSAC algorithm is expressed as a function of quantities that characterise the input data. We start the presentation by introducing the most important symbols used. The set of all data points is denoted U, the number of data points $N = |U|$, and ε represents the fraction of inliers contained in the data points. The symbol m is the size of the sample, i.e. the number of data points necessary to compute model parameters.

Let us first express the total time spent in the R-RANSAC procedure. The time needed to verify the consistency of one data point with the hypothesized parameters was chosen as a unit of time. The average time spent in R-RANSAC in number of verified data points is

$$J = k(\bar{t} + t_M), \tag{1}$$

where k is the number of samples drawn, $\bar{t}$ is the average number of verified data points within one model evaluation, and t_M is the time necessary to compute the parameter of the model from the selected sample. Note that t_M from (1) is a constant independent of both N and ε.

From (1) we can see, that the average time spent in R-RANSAC depends on both the number of samples drawn k and the average time required to process each sample. The analysis of these two components follows.

The number of tested hypothesis, which is equal to the number of samples, depends on the termination condition. Two different termination criteria may be adopted in RANSAC. The hypothesize-verify loop is either stopped after evaluation of more samples than expected on average before a good (uncontaminated) sample is selected or, alternatively, the number of samples is chosen to ensure that the probability that a better-than-currently-best sample is missed is lower than a predefined confidence level. We show that the stopping times for the two cases, average-driven and confidence-driven, differ only by a multiplicative factor and hence the optimal value in the proposed test is reached with the same parameters.

Since the sample is selected without repetition, the probability of taking a good sample is

$$P_I = \frac{\left(\begin{array}{c} I \\ m \end{array} \right)}{\left(\begin{array}{c} N \\ m \end{array} \right)} = \frac{I!\,(N-m)!}{(I-m)!\,N!} = \prod_{j=0}^{m-1} \frac{I-j}{N-j},$$

where $I = \varepsilon N$ stands for the number of inliers. For $N \gg m$ a simple and accurate approximation is obtained

$$P_I \approx \varepsilon^m, \tag{2}$$

which is exactly correct for sampling with repetition. The average number of samples taken before the first uncontaminated is given by (from properties of the geometric distribution)

$$\bar{k} = \frac{1}{\varepsilon^m \alpha}. \tag{3}$$

Note that for the randomised version of RANSAC the number of samples is higher than or equal to the standard version, because a valid solution may be rejected in a preliminary test

In:	$U = \{x_i\}$	set of data points, $\|U\| = N$
	$f : S \to p$	computes model parameters from a data point sample
	$\rho(p, x)$	the cost function for a single data point
Out:	p^*	parameters of the model maximizing the cost function

$k := 0$

Repeat until P{better solution exists} $< \eta$
(a function of C^* and no. of steps k)

$k := k + 1$

I. Hypothesis

(1) select randomly set $S_k \subset U$, $|S_k| = m$
(2) compute parameters $p_k = f(S_k)$

II. Preliminary test

(3) perform test based on $d \ll N$ data points
(4) continue verification only if the test is passed

III. Evaluation

(5) compute cost $C_k = \sum_{x \in U} \rho(p_k, x)$
(6) if $C^* < C_k$ then $C^* := C_k$, $p^* := p_k$

Table 1: *Summary of* RANSAC *and* R-RANSAC *algorithms. The step II is added to* RANSAC *to randomize its evaluation.*

with probability $1 - \alpha$. In the confidence-driven sampling, at least k samples have to be taken to reduce the probability of missing a good sample below a predefined confidence level η. Thus we get, as in [11]),

$$\eta = (1 - \varepsilon^m \, \alpha)^k, \tag{4}$$

and solving for k leads to

$$k = \frac{\log \eta}{\log (1 - \varepsilon^m \, \alpha)}. \tag{5}$$

Since $(1 - x)$ is the first order Taylor expansion of e^{-x} at zero, and $(1 - x) \leq e^{-x}$, we have

$$\begin{aligned} \eta \;=\; (1 - \varepsilon^m \, \alpha)^k \quad &\leq \quad e^{-\varepsilon^m \, \alpha \, k} \\ \ln \eta \quad &\leq \quad -\varepsilon^m \, \alpha \, k \\ \frac{-\ln \eta}{\varepsilon^m \, \alpha} \quad &\geq \quad k \end{aligned}$$

We see, that $k \leq \bar{k}(-\ln \eta)$, where $-\ln \eta$ is a predefined constant, so all formulae obtained for the η-confidence driven case can be trivially modified to cover the average case.

The Number of data points points tested. So far we have seen that introduction of a preliminary test has *increased the number of samples drawn*. For the pre-test to make sense, this effect must be more than offset by the reduction in the average number of data points tested per hypothesis. There are two cases to be considered. First, with probability P_I an uncontaminated ('good') sample is drawn. Then the preverification test is passed with probability α and all N data points are verified. Else, with probability $1-\alpha$, this good sample is rejected and only $\bar{t}_\alpha$ data points are on average tested. In the second case, a contaminated ('bad') sample is drawn, and this happens with probability $1-P_I$. Again either the pre-verification step is passed, but this time with a *different probability* β, and the full test with all N data points is carried out, or with probability $1-\beta$, only $\bar{t}_\beta$ data points are tested in the preverification test.

Here β stands for the probability, that a bad sample passes the preverification test. Note that it is important that $\beta \ll \alpha$, i.e. a bad (contaminated) sample is consistent with a smaller number of data points than a good sample. Forming a weighted average of the four cases, the formula for the average number of tests per sample is obtained:

$$\bar{t}(d) = P_I\,(\alpha N + (1-\alpha)\bar{t}_\alpha) + (1-P_I)(\beta N + (1-\beta)\bar{t}_\beta). \tag{6}$$

Values of α, β, $\bar{t}_\alpha$, and $\bar{t}_\beta$ depend on the type of preverification test.

3 The $T_{d,d}$ Test

In this section we introduce a simple and thus mathematically tractable class of preverification tests. Despite its simplicity, we show in the simulations and experiments of Section 4 its potential. The test we will analyze is defined as follows: Pass the test if all d data points out of d randomly selected are consistent with the hypothesized model. In the rest of this section we derive the optimal value for d. But first of all we have to derive constants introduced in the previous section

$$\alpha = \varepsilon^d \qquad \text{and} \qquad \beta = \delta^d,$$

where δ is the probability that a data point is consistent with a "random" model. Since we do not need to test all d, just to find first failure, the average time spent in the preverification test is

$$\bar{t}_\alpha = \sum_{i=1}^{d} i\,(1-\varepsilon)\,\varepsilon^{i-1} \qquad \text{and} \qquad \bar{t}_\beta = \sum_{i=1}^{d} i\,(1-\delta)\,\delta^{i-1}$$

Since

$$\sum_{i=1}^{d} i(1-x)x^{i-1} \le \sum_{i=1}^{\infty} i(1-x)x^{i-1} = \frac{1}{1-x}, \tag{7}$$

we have

$$\bar{t}_\alpha \le \frac{1}{1-\varepsilon} \qquad \text{and} \qquad \bar{t}_\beta \le \frac{1}{1-\delta}.$$

The relationship we get after substituting these into (6)

$$\bar{t}(d) \approx \varepsilon^m \left(\varepsilon^d N + \frac{1-\varepsilon^d}{1-\varepsilon} \right) + (1-\varepsilon^m) \left(\delta^d N + \frac{1-\delta^d}{1-\delta} \right)$$

is too complicated, so we incorporate the following approximations

$$(1-\varepsilon^m)\frac{1-\delta^d}{1-\delta} \approx 1,$$

$$(1-\varepsilon^m)\delta^d N \approx \delta^d N, \text{ and}$$

$$\varepsilon^d N \gg \frac{1-\varepsilon^d}{1-\varepsilon}.$$

After applying these approximations, we have

$$\bar{t}(d) \approx N\,\delta^d + 1 + \varepsilon^{m+d}\,N \tag{8}$$

The average time spent in R-RANSAC in number of verified data points is then approximately

$$J(T_{d,d}) \approx \frac{1}{\varepsilon^m\,\varepsilon^d}\left(N\,\delta^d + \varepsilon^{m+d}\,N + 1 + t_M\right) \tag{9}$$

We are looking for the minimum of $J(T_{d,d})$ and hence we can solve $\frac{\partial J(T_{d,d})}{\partial d} = 0$ for d and we get optimal length of the $T_{d,d}$ test as follows

$$d^* \approx \frac{\ln\left(\frac{\ln\varepsilon(t_M+1)}{N\,(\ln\delta - \ln\varepsilon)}\right)}{\ln\delta}. \tag{10}$$

The value of d_{opt} must be an integer greater or equal to zero, so it could be obtained as

$$d_{opt} = \max(0, \arg\min_{d\in\{\lfloor d^*\rfloor,\lceil d^*\rceil\}} J(T_{d,d})). \tag{11}$$

Since the cost function $J(T_{d,d})$ has only one extreme and for $d \to \pm\infty$ we have $J(T_{d,d}) \to \infty$, we can say that R-RANSAC is faster than the standard RANSAC if

$$J(T_{0,0}) > J(T_{1,1}).$$

From this equation we get

$$N > (t_M + 1)\frac{1-\varepsilon}{\varepsilon-\delta}. \tag{12}$$

4 Experiments

In this section are experiments that show the usefulness of the new randomised RANSAC algorithm with a preverification tests. The speed-up is demonstrated on the problem of epipolar geometry estimation. Three experiments are conducted on data from a synthetic, a short (standard) and wide-baseline stereo matching problem. Results of these experiments are summarized in tables 2, 3, and 4 respectively. The structure of the tables is the following. The first column shows the length d of the $T_{d,d}$ test, where $d = 0$ means standard RANSAC. The number of samples, each consisting of 7 point-to-point correspondences, that were used for model parameter estimation is given in the second column. Since the seven-point algorithm [3] for computation of the fundamental matrix may lead to one or three solutions, the next column, labeled 'models', shows the number of hypothesized

d	samples	models	tests	inliers	time
0	1866	4569	6821218	600	25.0
1	4717	11536	16311	600	6.0
2	11849	28962	33841	600	15.1

Table 2: *Synthetic experiment on 1500 correspondences, 40% of inliers, 30 repetitions.*

d	samples	models	tests	inliers	time
0	480	1146	766875	343	2.6
1	960	2301	83953	342	1.4

Table 3: *Short baseline experiment on 676 tentative correspondences.*

fundamental matrices. The 'tests' column displays the number of point-to-point correspondences evaluated during the procedure. In the penultimate column, the average number of inliers detected is given. The last column is rather informative and shows the time in seconds taken by the algorithm. This is strongly dependent on the implementation.

Synthetic experiment. 1500 correspondences were generated, 900 outliers and 600 inliers. Since the run-time of both RANSAC and R-RANSAC is a random variable, the programs were executed 30 times and averages were taken. Result are shown in the Table 2. Since the number of correspondences is large, the standard RANSAC algorithm spends a long time verifying all correspondence as can be seen in column 'tests'.

Short baseline experiment was conducted on the images from a standard dataset of the Leuven castle [7]. There were 676 tentative correspondences formed by the Harris interest points followed by cross-correlation of its neigbourhood. The tentative correspondences contained approximately 60% of inliers. Looking at Table 3 we see that approximately twice as many fundamental matrices were hypothesized in R-RANSAC, but more than nine times less correspondences were evaluated.

Wide baseline experiment on the BOOKSHELF dataset. The tentative correspondences were formed as follows. Discriminative regions (MSERs, SECs) [5] were detected. Robust similarity functions on the affine invariant description were used to establish the mutually nearest pair of regions. Point correspondences were obtained as centers of gravity of those regions. There were less then 40% of inliers among the correspondences.

5 Conclusion

In this paper, we presented a new R-RANSAC algorithm, which increased the speed of model parameter estimation under a broad range of conditions, due to the hypothesis evaluation

d	samples	models	tests	inliers	time
0	3094	7582	3078184	161	12.9
1	6366	15583	178217	164	8.7

Table 4: *Wide baseline experiment on 413 tentative correspondences.*

Figure 1: *Short baseline image set*

Figure 2: *Wide baseline image set*

being randomized. For samples contaminated by outliers, it was shown that it was sufficient to test only a small number of data points $d \ll N$ to conclude with high confidence that they do not correspond to the sought solution. The idea was implemented in a two-step evaluation procedure (Table 1). We also introduced in Section 3 a mathematically tractable class of pre-tests based on small test samples. For this class we derived an approximate relation for optimal setting of its single parameter. The proposed pre-test was verified on both synthetic data and real-world problems and a significant increase in speed was observed. The task for the future is to design the optimal preverification test.

References

[1] J. Clarke, S. Carlsson, and A. Zisserman. Detecting and tracking linear features efficiently. In *Proc. 7th BMVC*, pages 415–424, 1996.

[2] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, June 1981.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[4] A. Leonardis, H. Bischof, and R. Ebensberger. Robust recognition using eigenimages. Technical Report TR-47, PRIP, TU Wien, 1997.

[5] Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. Wide-baseline stereo from distinguished regions. Research Report CTU–CMP–2001–33, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, November 2001.

[6] P. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. In *Proc. BMVC*, pages 616– 62, 2000.

[7] M. Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, ESAT-PSI, K.U.Leuven, 1999.

[8] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. International Conference on Computer Vision*, pages 754–760, 1998.

[9] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[10] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*, July 2001.

[11] P. H. S. Torr. *Outlier Detection and Motion Segmentation*. PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.

[12] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[13] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *CVIU*, 71(3):312–333, September 1998.

[14] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. 11th British Machine Vision Conference*, 2000.

A Layered Parametric Fitting Method for Sparse Data

Daniel Beresford and Adrian Hilton
Centre for Vision, Speech and Signal Processing
University of Surrery, Guildford, Surrey, GU2 7XH
e-mail: d.beresford@eim.surrey.ac.uk

Abstract

This paper examines how a parameterised model can be fitted to a sparse unstructured 3D data set. A model constructed from two layers is employed in the fitting process. The underlying layer acts as a Control Layer whose vertices are the parameters involved in the fitting process. Defined by the Control Layer is a High Resolution Layer which is the surface to be fitted to the 3D data set. A non-linear least-squares minimisation technique finds the best fit for the model given the data by altering the parameters. Using a priori knowledge of the data, sparse data sets can be reconstructed as unknown data regions are recovered through careful selection of the model. This method has been applied to the reconstruction of simple primitives with a view to using it on more complex structures such as the human face.

$Key\ Words$: sparse data, layered parametric model, a priori knowledge.

1 Introduction

Capturing and reconstructing objects from images with little texture information is a difficult problem. This problem can be avoided by using a laser scanner or information such as the silhouette of the object. However building a model from silhouettes usually requires the object to be placed on a turntable providing accurate information about angle of rotation. The background also needs to be extractable. Laser scanners are a popular commercial tool but are expensive and the data can be noisy.

An alternative is to reconstruct an object from a set of images. The problem then arises of acquiring enough information from the images to build the object. This paper deals with reconstructing an object from the 3D information obtained from a set of images. The problem has now become one of reconstructing an object from a sparse amount of 3D information. To aid the reconstruction the assumption will be made that the general form of the target object is known i.e. some initial structure of the object can be supplied.

Recovering an object from a sparse 3D data set is the principal aim here following on from a previous piece of work where sparse data was recovered from a Model Driven Bundle Adjustment method based on [5].

What is sought after is a method to fit a single model to a set of sparse data points. Also included is some explanation of how once an initial fit has been obtained improvements can be made to the parameterisation through analysis of local fitting errors. As a priori knowledge is assumed some restrictions on the variation of the initial model can be made, and although this slows down the fitting process it helps prevent unlikely fits occuring.

An extension to this method of using apriori knowledge would be to incorporate statistical knowledge gained from many similar objects as has been shown in [1], where a database of object variation has been obtained.

1.1 Impetus behind the method

The approach uses a reduced parametric model as described in Section 2. The reasons for adopting this approach are two fold. Firstly only a small number of parameters are used to describe the shape of the model [10]. Secondly as it is assumed that the general structure of the data is known a priori but the amount of data is limited the shape of the model can be used to 'interpolate' across the unmeasured data regions. However even though the main aim is to fit to sparse data sets the method can also be applied to dense 3D data. As the structure of the model is known a priori this will provide some advantages as for example the recovered object could then be animated based on the predefined parameter set applied to the initial model.

1.2 Knowledge a priori

Many methods of reconstruction from 3D data sets produce good surface reconstructions [6] [11] [9] but require dense point measurements. They also provide little information about the structure of the reconstructed object. Some form of parameterisation [10] can provide an initialisation of the objects shape [4] [8]. An overview of surface reconstruction methods can be found in [12]. Acknowledging the general structure of the object allows the initial choice of parameter positions to be carefully chosen beforehand. This not only allows fitting to the object to occur rapidly due to the model already being close to the data, but owing to structural similarities between the model and the object that the data represents, the density of the data set need not be high.

2 Structure of Model

The basis of the method is a Control Layer containing few vertices, which are used as the parameters within the fitting process. These Control vertices are joined in a triangulated mesh to form a simple surface. The vertices control a surface with a higher density of vertices. Once the relationship between the Control Layer and this High Resolution Layer is defined then the High Resolution surface can be altered through the movement of the vertices of the Control Layer. Above the High Resolution surface sits the data which is in the form of 3D unstructured data. Figure 1 shows a simplified representation of the relationships between the layers of the model and the data.

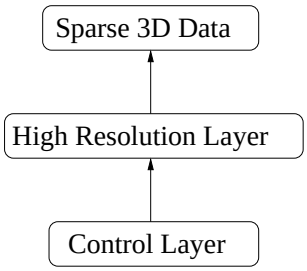


Figure 1: chart showing connection between the Layered Model and Data

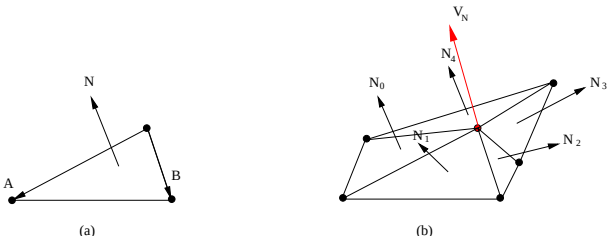


Figure 2: Vertex normal calculated from surrounding mesh-face normals

At this stage it needs to be pointed out that the High Resolution mesh and the Control mesh are initially created as follows. A dense 3D surface is created which bears some relation to the data set. This dense surface will be the High Resolution Layer, vertices on this dense mesh are then chosen and form the Control Layer. In the following section it will be shown how the High Resolution mesh is controlled by the Control Layer.

2.1 Volumetric Model

The relationship between the Control Layer and the High Resolution Layer is defined through a simple volumetric method [13]. The steps in the creation of the volume will be defined below but it is important to understand the overall objective which is to link the High Resolution surface to the simple Control Layer through some simple parameters. These parameters being the vertices of the Control Layer.

2.1.1 From Vertex Normal to Volume

The first step is to calculate the normals of the vertices in the Control Layer. This is carried out by calculating the normals of the faces of the mesh surrounding this vertex, weighting their values based on the angle each face makes with the vertex under consideration, i.e. the angle the face makes at that vertex, and finding the average, as shown in figure 2.

The vertex normals are then used to create a volume for each mesh face as shown in figure 3 (a). The length of the vertex normals can be controlled to ensure only relevant data is included in the volumes, this means outliers are removed because points outside the volume defined by the vertex normals are considered, this is important in the fitting process explained in Section

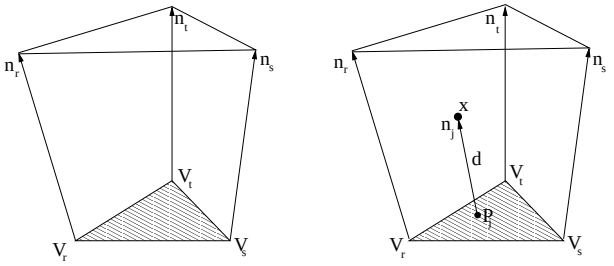


Figure 3: point to surface mapping

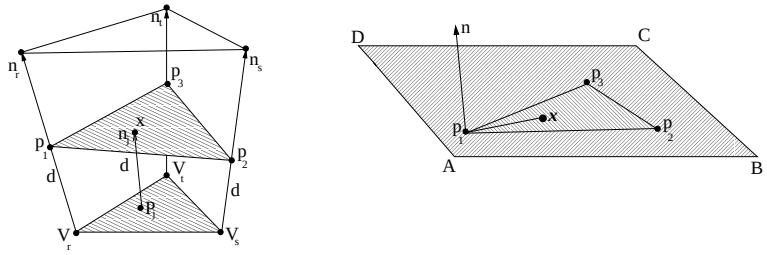


Figure 4: calculating coefficients

3. Also shown in diagram 3 is a point from the High Resolution surface falling inside the volume. This point can now be defined by the Control Layer using the vertices (V_r, V_s, V_t), their normals (n_r, n_s, n_t) and the distance d along their normals which is the Euclidean distance of the vertex x from the Control surface along the normal n_j from a point p_j on the Control Layer.

2.1.2 Defining a High Resolution Layer Vertex with respect to the Control Layer

A vertex on the High Resolution Layer can be defined by a point on the surface of the Control Layer and its normal through the use of barycentric coordinates, equation 1.

$$\begin{aligned} p_j &= \alpha V_r + \beta V_s + \gamma V_t \\ n_j &= \alpha n_r + \beta n_s + \gamma n_t \end{aligned} \tag{1}$$

$$\begin{aligned} x &= p_j + dn_j \\ x &= \alpha(V_r + dn_r) + \beta(V_s + dn_s) + \gamma(V_t + dn_t) \end{aligned} \tag{2}$$

Mapping a High Resolution vertex x onto the Control Layer is defined in equation 2, where d is the Euclidean distance of point x along the normal n_j from the point p_j on the Control Layer. The next problem is to solve for the unknown coefficients (α, β, d) where $\gamma = (1 - \alpha - \beta)$ which is shown in section 2.1.3.

First d is solved for and once this is known points (p_1, p_2, p_3) (see figure 4) can be calculated and then it is just a matter of finding the barycentric coordinates of triangle (p_1, p_2, p_3) for point x.

2.1.3 Solving for d

To calculate d the plane equation (4), is required. The plane equation is solved by considering the points $(p_1, p_2, p_3 \ and \ x)$ which all lie on the same plane and where x is a known High Resolution point, $p_1 = (V_r + dn_r),\ p_2 = (V_s + dn_s),\ p_3 = (V_t + dn_t)$. The normal n can be calculated from the cross product of two vectors on the plane such as $(p_{12} \times p_{13})$.

Equation 3 can be simplified to a cubic equation in terms of d which can then be solved either analytically or numerically. Each vertex on the High Resolution Layer will have its own value of d and barycentric coordinates (α, β, γ). These barycentric coordinates are the same for the projection of the High Resolution vertex onto the Control Layer, that is point p_j in figure 4. Once the coefficients are known for every point on the High Resolution surface this surface can be altered through the manipulation of the Control Layer.

$$\begin{aligned} 0 &= n \cdot (p_1 - x) \\ n &= p_{12} \times p_{13} \\ n &= [(V_r + dn_r) - (V_s + dn_s)] \times [(V_r + dn_r) - (V_t + dn_t)] \\ 0 &= [(V_r + dn_r) - (V_s + dn_s)] \times [(V_r + dn_r) - (V_t + dn_t)]((V_r + dn_r) - x) \end{aligned} \quad (3)$$

$$n \cdot (p_1 - x) = 0 \quad (4)$$

So far a simplified definition of a High Resolution surface by a Control Layer has been achieved. The next phase is to apply this within the process of fitting a High Resolution model to a 3D point data set.

3 The Fitting Process Levenberg-Marquart Least Squares Minimisation

The previous section described the parameterisation of the High Resolution Layer with respect to the Control Layer. This section will explain how this parameterisation can be employed in fitting the High Resolution surface to a 3D point set of scattered data.

Figure 5 demonstrates the setup of the two layers for the fitting process.

The fitting process uses energy minimisation to fit the High Resolution surface to the 3D data set. As in the Control Layer to High Resolution Layer the connection between the High Resolution Layer and the 3D data points employs a parameterisation, in this case based on the vertices of the High Resolution surface. This is shown in figure 5 (b) where X is the data point, and (P_0, P_1, P_2) is the High Resolution Layer with the normals (PN_0, PN_1, PN_2) creating the volume. The distance along the normal from the High Resolution Layer to the data point (shown in figure 5 (b) as X) is the distance to be minimised. This distance is from the 3D data point X to its projection onto the High Resolution Layer X_{pr}. The minimum of the energy equation 5 is searched for, where σ is the variance of the data and ϕ represents the parameters which are

the vertices of the Control Layer. The aim is to find the set of parameters that give the highest probability for the data set to have occured i.e. the minimum energy.

The Levenberg-Marquart algorithm requires the gradient to be calculated from equation 6, and the approximate Hessian matrix from equation 7 which supplies how far to descend along the gradient. The second derivative terms are ignored here, as is stated in [14], due to these terms being destabilizing. Given an estimate of a (the initial Control Vertices positions), equation 8 calculates an increment δa. The values of b, A and a are held in matrix form. A varying factor λ is applied to the diagonal terms in the approximate Hessian matrix and either increases or decreases for a subsequent iteration depending on an increase or decrease in the error between data and the model for that iteration.

$$\epsilon(\phi) = \sum_{i=1}^{N} \left[\frac{X_i - X_{pr}(\phi)_i}{\sigma_i} \right]^2 \tag{5}$$

$$b = \frac{\partial \epsilon}{\partial \phi_k} = \sum_{i=1}^{N} \frac{[X_i - X_{pr}(\phi)_i]}{\sigma_i^2} \frac{\partial X_{pr}(\phi)_i}{\partial \phi_k} \tag{6}$$

$$A = \frac{\partial^2 \epsilon}{\partial \phi_k \partial \phi_l} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left[\frac{\partial X_{pr\ i}}{\partial (\phi)_k} \frac{\partial X_{pr\ i}}{\partial (\phi)_l} \right] \tag{7}$$

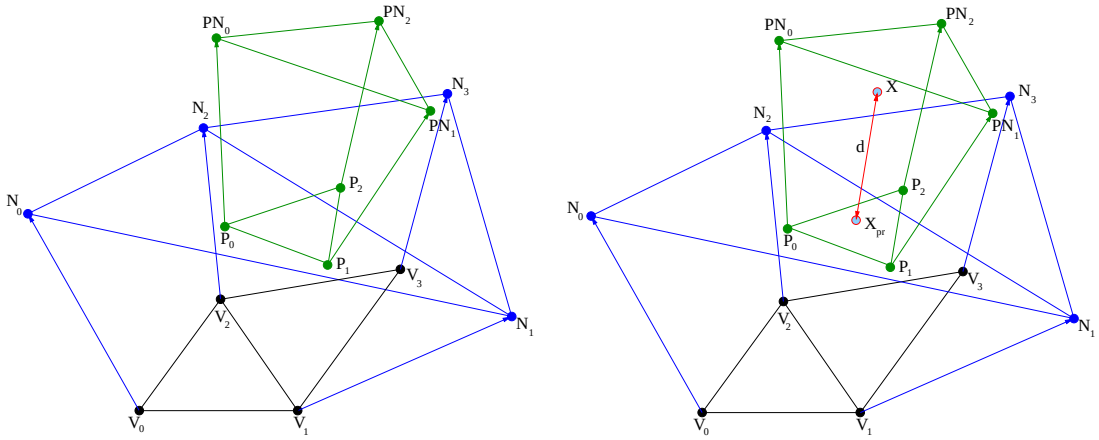
$$(A\ +\ \lambda I)\delta a\ =\ -b \tag{8}$$

The problem is set out in a least-squares form. Extreme outliers are eliminated as the volume defined by the normals of the vertices for each face in the mesh can be scaled according to the data. The Levenberg-Marquart method being a standard non-linear least squares technique is employed due to the fact that the equation to be minimised depends nonlinearly on the Control Layer parameters. The minimisation proceeds iteratively with trial values for the parameters. The Levenberg-Marquart method is used to move quickly to a solution by switching smoothly between the steepest descent method, which is used far from the minimum, to the inverse-Hessian method when the minimum is approached.

Spring Model

As there is a priori knowledge for the structure of the data set the initial model used can be biased towards that structure. To preserve the structure of the model and prevent points moving towards 'unlikely' positions a spring energy is built into the Control Layer [6]. The role of the spring energy is to retain the general shape of the model while still allowing the fitting process to alter the inital model towards the data set. This alters the energy minimisation equation by including a spring energy term shown in equation 9, where K represents the spring constant and δ the change in length of the spring.

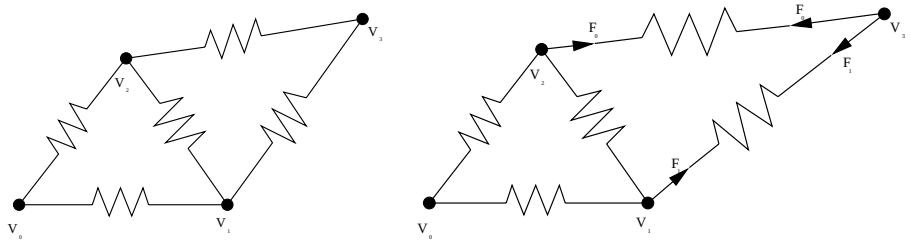
$$\epsilon = \epsilon_{dist}(X, \phi) + \epsilon_{spring}(K, \delta) \tag{9}$$



(a) Control Layer Volume and High Resolution Layer Volume

(b) Control Layer Volume, High Resolution Layer Volume and 3D data point

Figure 5: Layered Model



(a) Initial Control Point positions

(b) Shifted Control Point with spring forces

Figure 6: Spring Model

4 Illustration of Model and Fitting

A simple example of the fitting process is shown in figure 8. Which shows the data set as red points and the High Resolution Layer starting as a sphere and being deformed through the parameterisation of the Control Layer shown in figure 7. Three more simple examples are given in Figures 9, 10, 11. These figures show the High Resolution model on the left with the desired fit in the middle and the actual fit on the right.

5 Future Work

The fitting process is only concerned with obtaining a good geometric fit but this says nothing about how the structure of the model (defined in the High Resolution Layer) is fitting to the data.

Figure 7: Control Layer for Sphere to Cube fitting

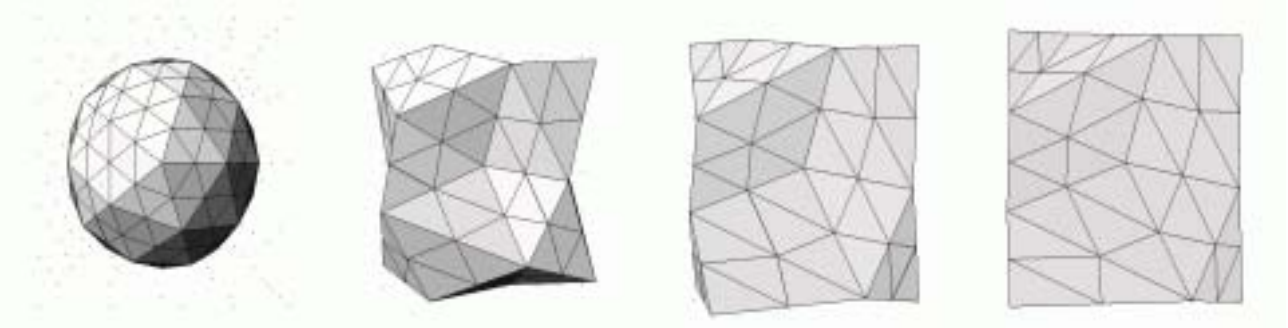


Figure 8: Fitting a Sphere to a Cube

Figure 9: High Resolution Model - Desired Fit - Actual Fit

Figure 10: High Resolution Model - Desired Fit - Actual Fit

Figure 11: High Resolution Model - Desired Fit - Actual Fit

To test this the change in curvature of the model [3] [7] [2] as it fits to the data can be calculated at the vertices and compared to the curvature of the data set at specific points of interest which depend on the data under consideration.

Local Minima and Local Errors in Fitting

A problem with the current setup of the method is the tendency to fall into local minima. To overcome this a validation scheme has been created whereby a sample of the data set is removed from the data and used solely to calculate the error in the fitting process. This will provide a good evaluation of how well the model is fitting to the data.

A point will be reached where the best fit for those given parameters is obtained. However the fit may be very bad visually and have a low probability that this data set could occur with these parameters. Therefore an improvement in the parameterisation is required. Within each volume of the Control Layer the fitting error will vary. For Volumes where the fitting error is high that Control Layer mesh triangle can be subdivided, the corresponding High Resolution mesh triangles can also be subdivided if required. The desired outcome through this increased parameterisation would be a reduced local error for this Control Volume and therefore a closer fit to the data.

6 Conclusions

Presented is a novel aproach to parametric fitting. The volumetric, layered model, reduces the number of parameters involved in the fitting. The ability to preserve discontinuities, inherent in the volumetric model allows areas of objects where large changes in curvature occur to be modelled. Most surface fitting methods require dense data but here sparse data is used with a priori knowledge for unknown regions. The a priori knowledge allows parameter positions to be specified in a reasonable fashion and in regions of the object where more detail will occur the High Resolution Layer can be given a correspondingly more densely defined mesh.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Siggraph 1999*, pages 187–194, 1999.

[2] Herve Delingette. Simplex Meshes: a General Representation for 3D Shape Reconstruction. Technical report, INRIA, 1994.

[3] D.Hilbert and S.Cohn-Vossen. *Geometry and the Imagination*. AMS Chelsea Publishing, 1990.

[4] Eric Bardinet, Laurent D. Cohen and Nicholas Ayache. A Parametric Deformable Model to Fit Unstructured 3D Data. *Computer Vision and Image Understanding*, pages 39–54, 1998.

[5] Pascal Fua. Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities. Technical report, INRIA, 1991.

[6] Hugues Hoppe, Tony DeRose Tom Duchamp, John McDonald and Wemer Stuetzle. Surface Reconstruction from Unorganized Points. *Computer Graphics*, pages 71–78, 1992.

[7] Jan Koenderink. *Solid Shape*. The MIT Press, 1990.

[8] C.W. Liao and G. Medioni. Surface approximation of a cloud of 3d points. *GMIP*, 57(1):67–74, January 1995.

[9] Nina Amenta Marshall Bern and David Eppstein. The Crust and the B-Skeleton : Combinatorial Curve Reconstruction. Technical report, Computer Sciences, University of Texas, Austin, 1997.

[10] Frederic I. Parke. Parameterized Models for Facial Animation. *IEEE Computer Graphics and Applications*, pages 61–68, November 1982.

[11] Radim Šára and Ruzena Bajcsy. Fish-scales: Representing fuzzy manifolds. In Sharat Chandran and Uday Desai, editors, *Proc. 6th International Conference on Computer Vision*, pages 811–817, New Delhi, India, January 1998. IEEE Computer Society, Narosa Publishing House.

[12] I. Soderkvist. Introductory overview of surface reconstruction methods, 1999.

[13] Wei Sun, Adrian Hilton, Raymond Smith and John Illingworth. Building Layered Animation Models from Captured Data. Technical report, Centre for Vision Speech and Signal Processing, University of Surrey, 2000.

[14] W.H.Press, B.P.Flannery, S.A.Teukolsky, and W.T.Vetterling. *Numerical Recipes, the Art of Scientific Computing*. Cambridge University Press, 1986.

Superellipsoids Gaining Momentum

Aleš Jaklič and Franc Solina

Computer Vision Laboratory

University of Ljubljana, Faculty of Computer and Information Science

Tržaška cesta 25, SI-1000 Ljubljana, Slovenia

phone: +386 1 4768 878, fax: +386 1 4264 647

e-mail: ales.jaklic@fri.uni-lj.si,franc.solina@fri.uni-lj.si

Abstract

We derive a closed form expressions for two-dimensional Cartesian moment m_{pq} of order $(p+q)$ of a superellipse and the three-dimensional Cartesian moment m_{pqr} of order $(p+q+r)$ of a superellipsoid in their respective canonical coordinate systems.

Moments of order n in coordinate system that is rigidly transformed from canonical coordinate system can be computed as a linear combination of moments in canonical coordinate system of order lower or equal to n . Additionally, moments of objects that are compositions of superellipsoids can be computed as simple sums of moments of individual parts.

To demonstrate practical application of derived results we register pairs of range images based on moments of recovered compositions of recovered superellipsoids. We use standard techniques to find center of gravity and principal axes while third-order moments are used to resolve four-way ambiguity. Experimental results show expected improvement of recovered rigid transformation as compared to the registration based on moments of raw range image data. Beside object pose estimation the presented method can be used for object recognition with moments and/or moment invariants as object features.

keywords: 3D moments, superellipsoid, transformations of moments, pose estimation

1 Introduction and Motivation

Moment-based techniques have a well established tradition in object recognition and pose estimation [10]. Initial two-dimensional moment invariants techniques were extended to three-dimensions [11, 8, 9] and three-dimensional moments were used for object-recognition [5].

Although algorithms and methods for segmentation and recovery of superellipsoids exist (see survey in [6]) moment-based methods have not been applied to such representations. Numerical integration was proposed to compute volume and moments of inertia for superellipsoids [15]. However, numerical integration must be performed for each pair of values of shape parameters ϵ_1 and ϵ_3 as well as for each order of moment. Closed form expressions for computation of

moments would thus allow computationally efficient application of moment-based techniques to objects represented as compositions of superellipsoids.

Recovery of superellipsoids from a single view range image is underconstrained and even additional constraint of minimal volume [12], does not guarantee a precise model for a single superellipsoid like object [14]. In order to obtain a precise model several range images taken from different viewpoints have to be combined into a single data set. Many registration and range data fusion algorithms are based on some form of local minimizaton and require a good initial estimate of the transformatation [2, 13, 3, 4]. The moment based method presented in this paper could provide such an estimate.

The paper is organized as follows: in section 2 we derive moments of superellipses and based on that moments of superellipsoids in their respective canonical coordinate systems. Section 3 derives expressions used in computation of moments in a coordinate system which is produced by a rigid transformation of the canonical coordinate system. Sections 4 and 5 present the registration algorithm used and the experimental results, respectively.

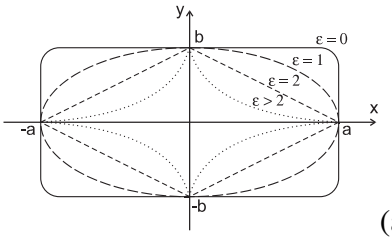
2 Moments of Superellipses and Superellipsoids

A superellipse is defined as a closed curve in $\mathbb{R}^2$ (see Figure 1 (a)), with parameters a, b, and ϵ

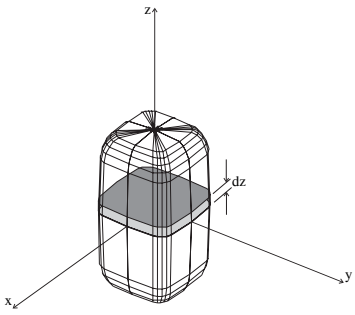
$$\mathbf{r}(\omega) \equiv \begin{bmatrix} x(\omega) \\ y(\omega) \end{bmatrix} \equiv \begin{bmatrix} a(\cos\omega)^{\epsilon} \\ b(\sin\omega)^{\epsilon} \end{bmatrix} \qquad -\pi \leq \omega < \pi \quad , \tag{1}$$

while a superellipsoid is defined as a closed surface in $\mathbb{R}^3$ (see Figure 1 (b)), with parameters a, b, c, ϵ_1, and ϵ_2 [1]

$$\mathbf{r}(\eta,\omega) \equiv \begin{bmatrix} x(\eta,\omega) \\ y(\eta,\omega) \\ z(\eta,\omega) \end{bmatrix} \equiv \begin{bmatrix} a(\cos\eta)^{\epsilon_1}(\cos\omega)^{\epsilon_2} \\ b(\cos\eta)^{\epsilon_1}(\sin\omega)^{\epsilon_2} \\ c(\sin\eta)^{\epsilon_1} \end{bmatrix} \qquad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi. \end{matrix} \tag{2}$$



(a)



(b)

Figure 1: (a) Superellipses for different values of parameter ϵ. (b) Geometrical interpretation of a superellipsoid as a stack of superellipses with infinitesimal thickness dz, their size being modulated by another superellipse.

2.1 Two-dimensional Cartesian Moment of Order $(p+q)$

The two-dimensional Cartesian moment of order $n = (p+q)$ is defined as

$$m_{pq} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x,y) dx dy \ . \tag{3}$$

Since we are interested in moments of a superellipse we set $f(x,y) = 1$ inside the superellipse and $f(x,y) = 0$ outside. Due to the symmetry of a superellipse with respect to x and y axis and origin of the coordinate system it is easy to note that:

$$p \text{ is odd} \vee q \text{ is odd} \Longrightarrow m_{pq} = 0 \ , \tag{4}$$

while for the case of p and q both being even the moment can be computed using a new coordinate system with coordinates r and ω instead of x and y. The transformation between the two systems is parameterized by a and ϵ and given by

$$\begin{aligned} x &= ar(\cos\omega)^\epsilon \ , \\ y &= br(\sin\omega)^\epsilon \ , \end{aligned} \tag{5}$$

with determinant of Jacobian matrix for the transformation

$$|\mathbf{J}| = abr\epsilon(\sin\omega)^{\epsilon-1}(\cos\omega)^{\epsilon-1} \ . \tag{6}$$

Since the p and q are both even, we can reduce the computation of the integral to the first quadrant

$$\begin{aligned} m_{pq} &= \int_{-b}^{b}\int_{-a}^{a} x^p y^q f(x,y) dx dy = 4\int_0^b \int_0^a x^p y^q f(x,y) dx dy \\ &= 4\int_0^{\pi/2}\int_0^1 (ar(\cos\omega)^\epsilon)^p (br(\sin\omega)^\epsilon)^q |\mathbf{J}| dr d\omega \\ &= \frac{4}{p+q+2} a^{p+1} b^{q+1} \epsilon \int_0^{\pi/2} (\sin\omega)^{(q+1)\epsilon-1} (\cos\omega)^{(p+1)\epsilon-1} d\omega \\ &= \frac{2}{p+q+2} a^{p+1} b^{q+1} \epsilon B((q+1)\frac{\epsilon}{2}, (p+1)\frac{\epsilon}{2}) \ , \end{aligned} \tag{7}$$

where beta function $B(x,y)$ is defined as

$$B(x,y) \equiv 2\int_0^{\pi/2} (\sin\phi)^{2x-1} (\cos\phi)^{2y-1} d\phi = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \ . \tag{8}$$

Table 1: Area and moments of inertia for superellipses of various shapes computed from (7) and using limit (29) for cases with $\epsilon = 0$.

	$\epsilon = 0$ (rectangle)	$\epsilon = 1$ (ellipse)	$\epsilon = 2$ (rhomb)
Area (m_{00})	$4ab$	πab	$2ab$
Moment of inertia about the x axis (m_{02})	$\frac{4}{3}ab^3$	$\frac{\pi}{4}ab^3$	$\frac{1}{3}ab^3$
Moment of inertia about the y axis (m_{20})	$\frac{4}{3}a^3b$	$\frac{\pi}{4}a^3b$	$\frac{1}{3}a^3b$

2.2 Three-dimensional Cartesian Moment of Order $(p + q + r)$

Three-dimensional Cartesian moment of order $n = (p + q + r)$ is defined as

$$m_{pqr} \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q z^r f(x, y, z) dx dy dz \ . \tag{9}$$

Again we set $f(x, y, z) = 1$ inside the superellipsoid and $f(x, y, z) = 0$ outside the superellipsoid. The moment can be expressed with the two-dimensional moments m_{pq} in the plane parallel to the xy plane as (see Figure 1 (b)):

$$m_{pqr} = \int_{-c}^{+c} \int_{-b}^{+b} \int_{-a}^{+a} x^p y^q z^r f(x, y, z) dx dy dz = \int_{-c}^{+c} z^r m_{pq}(z) dz \ . \tag{10}$$

Intersection of a plane parallel to xy with the superellipsoid is a superellipse with parameters $a(z)$, $b(z)$, and ϵ_1. From (4) and the symmetry of superellipsoid with respect to the xy plane it follows that

$$p \text{ is odd} \vee q \text{ is odd} \vee r \text{ is odd} \Longrightarrow m_{pqr} = 0, \tag{11}$$

and for the case when all of p, q, and r are even:

$$m_{pqr} = \int_{-c}^{+c} z^r m_{pq}(z) dz = 2 \int_0^{\pi/2} z(\eta)^r m_{pq}(\eta) \dot{z}(\eta) d\eta$$

$$= 2 \int_0^{\pi/2} (c(\sin\eta)^{\epsilon_1})^r \left(\frac{2}{p+q+2} a^{p+1}(\eta) b^{q+1}(\eta) \epsilon_2 B((q+1)\frac{\epsilon_2}{2}, (p+1)\frac{\epsilon_2}{2}) \right) c\epsilon_1 (\sin\eta)^{\epsilon_1 - 1} \cos\eta d\eta$$

$$= \frac{4}{p+q+2} c^{r+1} \epsilon_1 \epsilon_2 B((q+1)\frac{\epsilon_2}{2}, (p+1)\frac{\epsilon_2}{2}) \int_0^{\pi/2} (\sin\eta)^{\epsilon_1(r+1)-1} (a(\cos\eta)^{\epsilon_1})^{p+1} (b(\cos\eta)^{\epsilon_1})^{q+1} \cos\eta d\eta$$

$$= \frac{4}{p+q+2} a^{p+1} b^{q+1} c^{r+1} \epsilon_1 \epsilon_2 B((q+1)\frac{\epsilon_2}{2}, (p+1)\frac{\epsilon_2}{2}) \int_0^{\pi/2} (\sin\eta)^{\epsilon_1(r+1)-1} (\cos\eta)^{\epsilon_1(p+q+2)+1} d\eta$$

$$= \frac{2}{p+q+2} a^{p+1} b^{q+1} c^{r+1} \epsilon_1 \epsilon_2 B((r+1)\frac{\epsilon_1}{2}, (p+q+2)\frac{\epsilon_1}{2} + 1) B((q+1)\frac{\epsilon_2}{2}, (p+1)\frac{\epsilon_2}{2}) \tag{16}$$

Table 1 and Table 2 show the values of derived expressions for two and three-dimensional moments for some common geometric shapes. The computed expressions correspond exactly to the well-known expressions derived by direct integration for those specific shapes [9].

Table 2: Volume and moments of inertia for superellipsoids of various shapes computed from (16) and using limits (29), (30) for cases where $\epsilon_1 = 0$ or $\epsilon_2 = 0$.

	$\epsilon_1 = 0$ $\epsilon_2 = 0$ (plate)	$\epsilon_1 = 0$ $\epsilon_2 = 1$ (elliptical cylinder)	$\epsilon_1 = 1$ $\epsilon_1 = 1$ (ellipsoid)
Volume (m_{000})	$8abc$	$2\pi abc$	$\frac{4}{3}\pi abc$
Moment of inertia about the x axis ($m_{020} + m_{002}$)	$\frac{8}{3}abc(b^2 + c^2)$	$\pi abc(\frac{1}{2}b^2 + \frac{2}{3}c^2)$	$\frac{4}{15}\pi abc(b^2 + c^2)$
Moment of inertia about the y axis ($m_{200} + m_{002}$)	$\frac{8}{3}abc(a^2 + c^2)$	$\pi abc(\frac{1}{2}a^2 + \frac{2}{3}c^2)$	$\frac{4}{15}\pi abc(a^2 + c^2)$
Moment of inertia about the z axis ($m_{200} + m_{020}$)	$\frac{8}{3}abc(a^2 + b^2)$	$\pi abc(\frac{1}{2}a^2 + \frac{1}{2}b^2)$	$\frac{4}{15}\pi abc(a^2 + b^2)$

3 Transformation of Moments

Compositions of superellipsoids are expressed with transformations that relate a canonical coordinate system of each superellipsoid ($\mathbf{x}$) to a common global coordinate system ($\mathbf{x}^G$), usually the coordinate system of the range scanner. Moments of such non-penetrating compositions of superellipsoids can be computed as simple sums of moments of each superellipsoid. Without loss of generality, a rigid transformation can be decoupled into rotation followed by translation

$$\mathbf{x}^G = \mathbf{T}\mathbf{x} = \mathbf{T}_{tra}\mathbf{T}_{rot}\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} \tag{17}$$

For the case of pure translation, and using binomial theorem it follows

$$\begin{aligned} m^G_{pqr} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x+p_x)^p(y+p_y)^q(z+p_z)^r f(x+p_x, y+p_y, z+p_z)dxdydz \\ &= \sum_{i=0}^{p}\sum_{j=0}^{q}\sum_{k=0}^{r}(i,p-i)!\,(i,q-j)!\,(k,r-k)!\,p_x^{p-i}p_y^{q-j}p_z^{r-k}\;\; m_{ijk} \end{aligned} \tag{18}$$

Note that for translation moment of order n in translated coordinate system is a combination of moments of order less or equal to n in the original coordinate system.

For the case of pure rotation, we can use the multinomial theorem to expand the power terms

$$\begin{aligned} m^G_{pqr} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(n_x x + o_x y + a_x z)^p(n_y x + o_y y + a_y z)^q(n_z x + o_z y + a_z z)^r f(x,y,z)dxdydz \\ &= \sum_{\substack{i+j+k \\ =p}}\sum_{\substack{l+m+n \\ =q}}\sum_{\substack{t+u+v \\ =r}}(i,j,k)!(l,m,n)!(t,u,v)!n_x^i o_x^j a_x^k n_y^l o_y^m a_y^n n_z^t o_z^u a_z^v\; m_{(i+l+t)(j+m+u)(k+n+v)} \end{aligned} \tag{19}$$

Note that for rotation moment of order n in rotated coordinate system is a combination of moments of the same order n in the original coordinate system.

Table 3: Moments of a superellipsoid in global coordinate system expressed with parameters of the rigid transformation and moments of the superellipsoid in canonical coordinate system

0th order moment

$$m^G_{000} = m_{000}$$

1st order moments

$$\begin{aligned} m^G_{100} &= p_x m_{000} \\ m^G_{010} &= p_y m_{000} \\ m^G_{001} &= p_z m_{000} \end{aligned}$$

2nd order moments

$$\begin{aligned} m^G_{200} &= p_x^2 m_{000} + n_x^2 m_{200} + o_x^2 m_{020} + a_x^2 m_{002} \\ m^G_{110} &= p_x p_y m_{000} + n_x n_y m_{200} + o_x o_y m_{020} + a_x a_y m_{002} \\ m^G_{101} &= p_x p_z m_{000} + n_x n_z m_{200} + o_x o_z m_{020} + a_x a_z m_{002} \\ m^G_{020} &= p_y^2 m_{000} + n_y^2 m_{200} + o_y^2 m_{020} + a_y^2 m_{002} \\ m^G_{011} &= p_y p_z m_{000} + n_y n_z m_{200} + o_y o_z m_{020} + a_y a_z m_{002} \\ m^G_{002} &= p_z^2 m_{000} + n_z^2 m_{200} + o_z^2 m_{020} + a_z^2 m_{002} \end{aligned}$$

3nd order moments

$$\begin{aligned} m^G_{300} &= p_x^3 m_{000} + 3 p_x m_{200} \\ m^G_{210} &= p_x^2 p_y m_{000} + p_y m_{200} \\ m^G_{201} &= p_x^2 p_z m_{000} + p_z m_{200} \\ m^G_{120} &= p_x p_y^2 m_{000} + p_x m_{020} \\ m^G_{111} &= p_x p_y p_z m_{000} \\ m^G_{102} &= p_x p_z^2 m_{000} + p_x m_{002} \\ m^G_{030} &= p_y^3 m_{000} + 3 p_y m_{020} \\ m^G_{021} &= p_y^2 p_z m_{000} + p_z m_{020} \\ m^G_{012} &= p_y p_z^2 m_{000} + p_y m_{002} \\ m^G_{003} &= p_z^3 m_{000} + 3 p_z m_{002} \end{aligned}$$

Symmetry of superellipsoids simplifies the computation of expressions (18) and (19) since most moments are equal to 0 in the canonical coordinate system. Table 3 contains expressions for moments of superellipsoids up to the third order.

4 Range image registration

The basic idea of range image registration based on moments is to construct a coordinate frame, which is rigidly attached to the object in each image [10, 5, 8]. After constructing the two frames, we know their relationship to the global coordinate system and thus we also know the rigid transformation between the two frames, which is also the rigid transformation of the object. We will name the constructed frames the canonical frames.

A canonical frame has its origin in the center of gravity of the object. In such a frame the first order moments are equal to 0. The axes of the coordinate system are aligned along the axes of minimal and maximal moment of inertia. Both, the center of gravity as well as the axes of minimal and maximal moment of inertia are invariant to rigid transformation of the coordinate system and are intrinsically related to the space occupancy distribution of the object.

Since we are dealing only with right hand Cartesian coordinate frames we uniquely determine the remaining third axis by fixing two axes of the coordinate system. For our work we freely selected the x and the z axes to correspond to the minimal and to the maximal moment of inertia, respectively. Note however that the moments of inertia are invariant to rotation of the coordinate frame for 180° about any of the coordinate axes. This leads to four possible orientations of the canonical coordinate frame: the constructed one, and one for each rotation of 180° about the x, y, and z axis respectively, as shown in Figure 2.

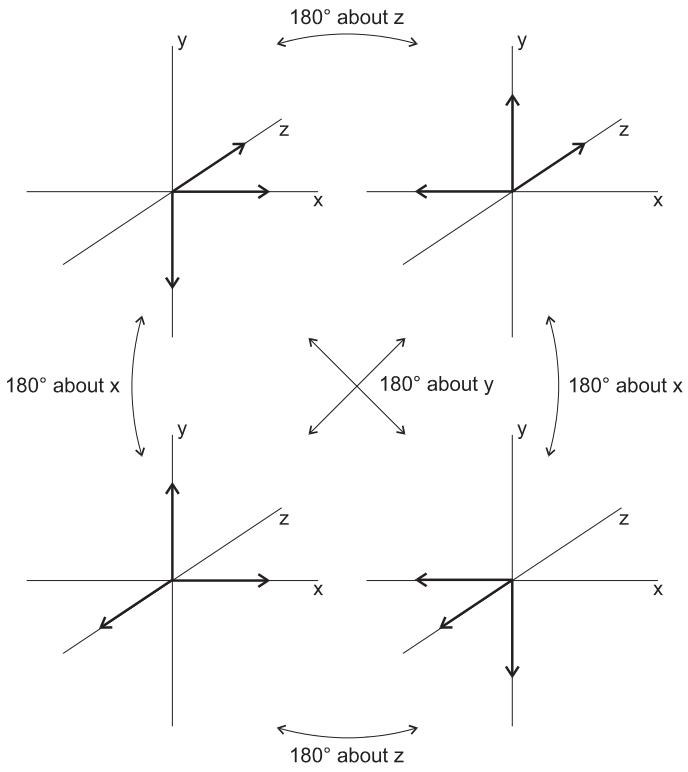


Figure 2: Four different right hand Cartesian coordinate frames with their axes aligned along given lines. Each one of them generates the remaining three by rotations of 180° about all the axes.

4.1 Resolving 4-way ambiguity

A search for the most distant point on the object from the origin of the coordinate system along the principal axes was proposed in [5], and the use of third order moments in [8], to resolve the 4-way ambiguity. The presented approach is similar to [8], but with much simpler derivation.

It is instructive to determine how solid moments of the same object computed in the four coordinate frames are related. Rotation of coordinate system (x, y, z) about the x axis produces new coordinate system (x', y', z') where $x' = x$, $y' = -y$, and $z' = -z$, the determinant of the Jacobian matrix for the transformation equals 1 so moments computed in the coordinate systems rotated about the x, y, and z axis are related to moments in original coordinate systems as follows:

$$M^{\mathrm{x}}_{pqr} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p(-y)^q(-z)^r dxdydz = (-1)^{q+r} M_{pqr} \tag{20}$$

$$M^{\mathrm{y}}_{pqr} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (-x)^p y^q(-z)^r dxdydz = (-1)^{p+r} M_{pqr} \tag{21}$$

$$M^{\mathrm{z}}_{pqr} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (-x)^p(-y)^q z^r dxdydz = (-1)^{p+q} M_{pqr} \tag{22}$$

Given a pair of vectors with third-order moments as components, we can transform the first vector to the three vectors corresponding to rotations about all the axes and then select the vector that is closest to the second vector to determine the rotation. Note that if due to object shape third-order moments vanish or the vector is equidistant to several vectors, higher order moments may be used in the same way. It follows directly from the selection of coordinate systems that moments of order lower than three cannot be used to resolve this ambiguity.

5 Experimental Results

In the first experiment we recovered rigid transformation between two range image views of a pile of stones. Algorithm described in [7, 6] was used to recover superellipsoid models from range images. To visualize the quality of recovered estimate of rigid transformation we overlaid the recovered models from view2 over the range image view1 Figure 3 (e) and recovered models from view1 over the range image view2 (f).

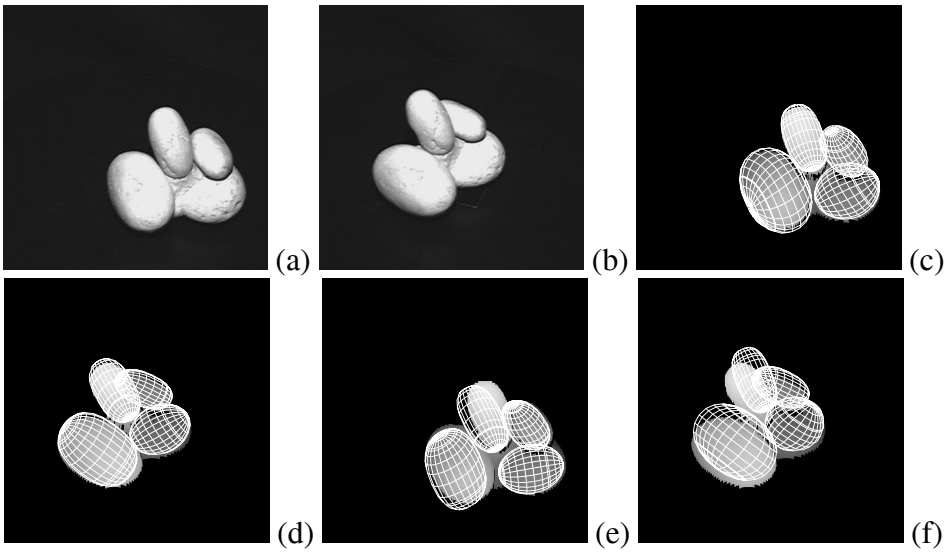


Figure 3: Registration of two real range images based on recovered superellipsoids (a) intensity image view1, (b) intensity image view2, (c) range image view1 with recovered superellipsoids (d) range image2 with recovered superellipsoids (e) models recovered from view2 overlaid over range image view1 using the recovered transformation (f) models recovered from view1 overlaid over range image2 using the recovered transformation

In the second experiment we generated a set of synthetic range images with known transformations among different views. The results presented in Figures 4–7 compare precision of estimates computed from moments of recovered models to estimates based on moments of range image data points. In the later case we approximated the integrals present in calculations of moments with sums of coordinates of range data points. The residual rigid transformation that transforms an estimated transformation to the true transformation was used as a quantitative measure of precision of recovered transformation.

6 Conclusions

We derived a closed form expressions for two-dimensional Cartesian moment m_{pq} of order $(p+q)$ of a superellipse and the three-dimensional Cartesian moment m_{pqr} of order $(p+q+r)$ of a superellipsoid. These results can be directly used to compute zeroth, first, and second order moments with well known physical meaning as area or volume, center of gravity and moments of inertia as well as to compute higher order moments used in applications of various moment invariants. To demonstrate the correctness of derived expressions we computed area and moments of inertia for standard two-dimensional shapes (rectangle, ellipse, rhomb) and volume and moments of inertia for standard three-dimensional shapes (plate, elliptical cylinder, ellipsoid).

Moments of order n in a coordinate system that is rigidly transformed from canonical coordinate system can be computed as a linear combination of moments in canonical coordinate system of order lower or equal to n . Additionally, moments of objects that are compositions of superellipsoids can be computed as simple sums of moments of individual parts. Since many moments of superellipsoids in their canonical coordinate system are equal to 0 transformation equations can be significantly reduced in the number of terms.

Feasibility of the proposed method was demonstrated with a registration of two real range views with unknown rigid transformation. Experiments with synthetic range images and know ground truth transformation showed significant better performance of range image registration based on moments of recovered superellipsoid models as compared to registration based on moments of range image data points. This is due to reduced effects of self-occlusion of parts and independence of density of range image data points.

The presented method can be directly used for object recognition with moments and/or moment invariants as object features.

A Beta and Gamma Functions

Beta function is defined as

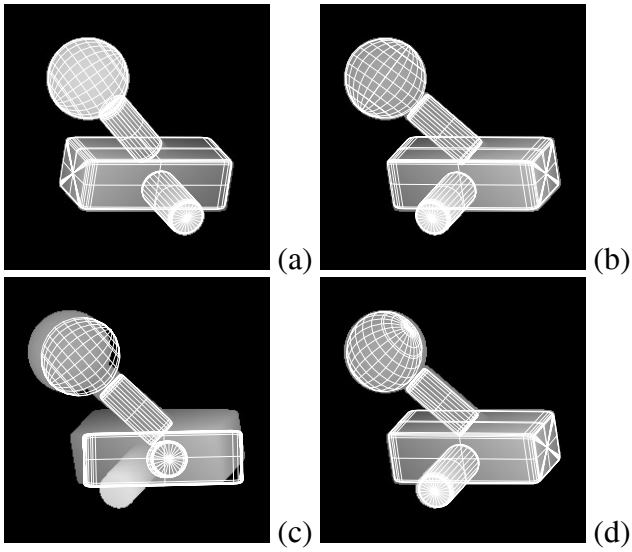
$$B(x,y) \equiv 2\int_0^{\pi/2} (\sin\phi)^{2x-1}(\cos\phi)^{2y-1}d\phi = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \tag{23}$$

For completeness we provide well know equalities for the gamma function used in further derivations

$$\Gamma(x+1) = x\Gamma(x) \tag{24}$$
$$\Gamma(n) = (n-1)! \tag{25}$$
$$\Gamma(1/2) = \sqrt{\pi} \tag{26}$$

From (24) and (26) follows that half integer arguments ($n = 1, 2, 3, ...$)

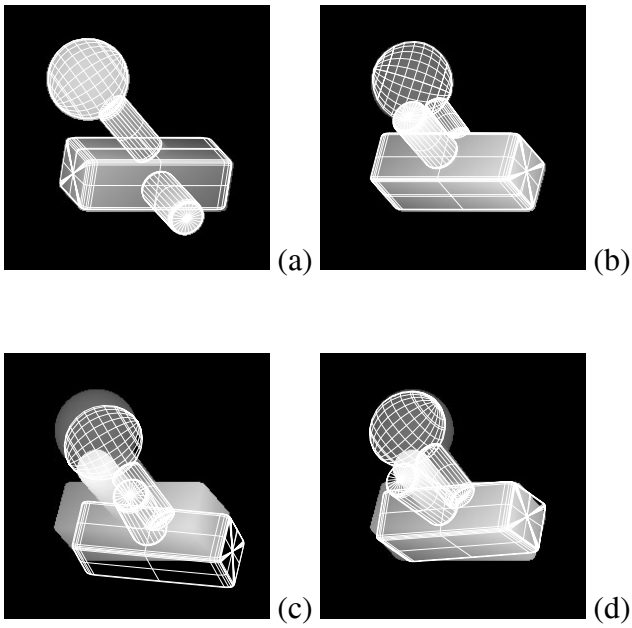
$$\Gamma(1/2+n) = \frac{1\cdot 3\cdot 5\cdot\cdots\cdot(2n-1)}{2^n}\sqrt{\pi}\ . \tag{27}$$



$$T^{image}_{res\_v1\_v2} = \begin{bmatrix} 0.90736 & -0.125109 & -0.401306 & 68.21 \\ -0.040771 & 0.923988 & -0.38024 & 71.8426 \\ 0.418374 & 0.361377 & 0.833289 & -75.9138 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{model}_{res\_v1\_v2} = \begin{bmatrix} 0.999874 & -0.00606522 & -0.0146585 & 0.839206 \\ 0.00611055 & 0.999977 & 0.00309931 & -0.742111 \\ 0.0146394 & -0.00318671 & 0.999887 & -2.1701 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

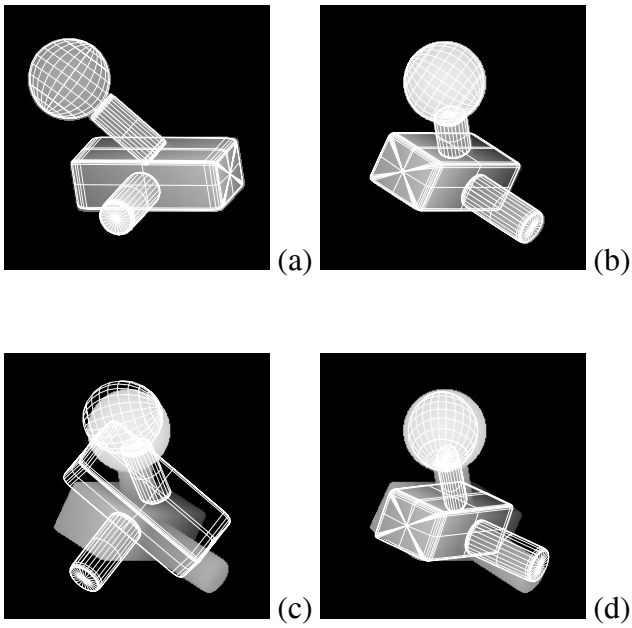
Figure 4: Registration of view1 to view2: (a) recovered model1 from view1, (b) recovered model2 from view2, (c) registration of model1 to view2 based on raw image data, (d) registration of model1 to view2 based on recovered models.



$$T^{image}_{res\_v1\_v5} = \begin{bmatrix} 0.981616 & -0.0551466 & -0.182724 & 20.2942 \\ 0.0698197 & 0.994748 & 0.0748542 & 1.82368 \\ 0.177639 & -0.0862351 & 0.98031 & -29.4717 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{model}_{res\_v1\_v5} = \begin{bmatrix} 0.991026 & 0.113205 & 0.071085 & -23.9595 \\ -0.118093 & 0.990616 & 0.0688275 & 8.30731 \\ -0.0626247 & -0.0766031 & 0.995092 & 14.738 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

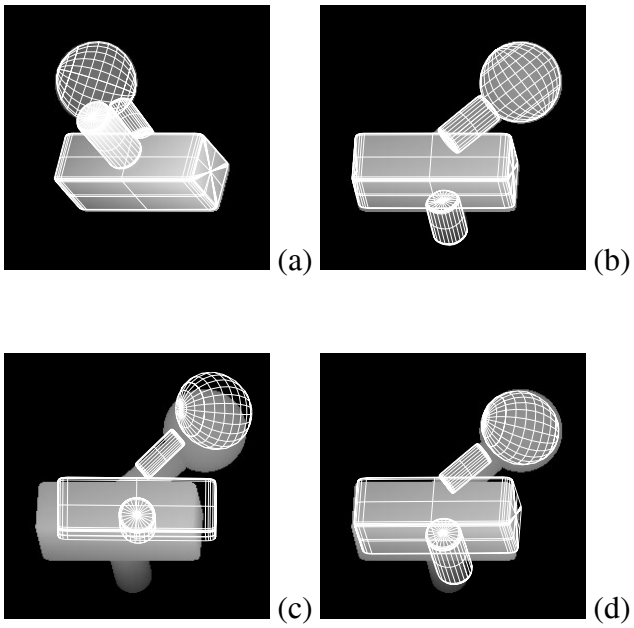
Figure 5: Registration of view1 to view5: (a) recovered model1 from view1, (b) recovered model5 from view5, (c) registration of model1 to view5 based on raw image data, (d) registration of model1 to view5 based on recovered models.



$$T^{image}_{res\_v2\_v6} = \begin{bmatrix} 0.158142 & -0.722261 & 0.673298 & 101.093 \\ 0.568553 & 0.624104 & 0.535951 & -121.143 \\ -0.807304 & 0.298049 & 0.509342 & 120.115 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{model}_{res\_v2\_v6} = \begin{bmatrix} 0.989199 & 0.126035 & -0.0748251 & 0.136176 \\ -0.126321 & 0.99199 & 0.000914878 & 16.7077 \\ 0.0743425 & 0.00854769 & 0.997197 & -10.3948 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 6: Registration of view2 to view6: (a) recovered model2 from view2, (b) recovered model6 from view6, (c) registration of model2 to view6 based on raw image data, (d) registration of model2 to view6 based on recovered models.



$$T^{image}_{res\_v5\_v7} = \begin{bmatrix} 0.991834 & 0.040307 & -0.121004 & 1.4898 \\ 0.00438126 & 0.937422 & 0.34817 & -68.0966 \\ 0.127467 & -0.345859 & 0.929589 & -3.5777 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{model}_{res\_v5\_v7} = \begin{bmatrix} 0.999646 & 0.00852136 & 0.025215 & -6.88662 \\ -0.00892507 & 0.999833 & 0.015955 & -6.65775 \\ -0.0250731 & -0.0161757 & 0.999555 & -0.385114 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 7: Registration of view5 to view7: (a) recovered model5 from view5, (b) recovered model7 from view7, (c) registration of model5 to view7 based on raw image data, (d) registration of model5 to view7 based on recovered models.

Bellow we derive the intermediate result frequently used in computing moments of superellipses and superellipsoids:

$$B(x, x+1) = \frac{\Gamma(x)\Gamma(x+1)}{\Gamma(2x+1)} = \frac{x\Gamma(x)\Gamma(x)}{2x\Gamma(x+1)} = \frac{1}{2}B(x, x). \tag{28}$$

Since $\Gamma(x)$ approaches $+\infty$ for $x \to 0^+$ and $-\infty$ for $x \to 0^-$ we have to compute the limits for the beta function terms for the case of rectangular shapes.

$$\begin{aligned} \lim_{\epsilon\to 0} \epsilon B(a\epsilon, b\epsilon) &= \lim_{\epsilon\to 0} \epsilon \frac{\Gamma(a\epsilon)\Gamma(b\epsilon)}{\Gamma((a+b)\epsilon)} = \lim_{\epsilon\to 0} \epsilon \frac{\frac{\Gamma(a\epsilon+1)}{a\epsilon}\frac{\Gamma(b\epsilon+1)}{b\epsilon}}{\frac{\Gamma((a+b)\epsilon+1)}{(a+b)\epsilon}} \\ &= (\frac{1}{a} + \frac{1}{b}) \lim_{\epsilon\to 0} \frac{\Gamma(a\epsilon+1)\Gamma(b\epsilon+1)}{\Gamma((a+b)\epsilon+1)} = \frac{1}{a} + \frac{1}{b} \end{aligned} \tag{29}$$

and

$$\begin{aligned} \lim_{\epsilon\to 0} \epsilon B(a\epsilon, b\epsilon+1) &= \lim_{\epsilon\to 0} \epsilon \frac{\frac{\Gamma(a\epsilon+1)}{a\epsilon}\Gamma(b\epsilon+1)}{\Gamma((a+b)\epsilon+1)} \\ &= \frac{1}{a} \lim_{\epsilon\to 0} \frac{\Gamma(a\epsilon+1)\Gamma(b\epsilon+1)}{\Gamma((a+b)\epsilon+1)} = \frac{1}{a}. \end{aligned} \tag{30}$$

References

[1] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, January 1981.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[3] G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.

[4] C. Dorai, J. Weng, and A. K. Jain. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1131–1138, 1997.

[5] J. M. Galvez and M. Canton. Normalization and shape recognition of three-dimensional objects by 3D moments. *Pattern Recognition*, 26(5):667–681, 1993.

[6] A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecth, 2000. ISBN 0-7923-6601-8.

[7] A. Leonardis, A. Jaklič, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(11):1289–1295, November 1997.

[8] C. Lo and H. Don. 3d moment forms: Their construction and application to object identification and positioning. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 11(10):1053–1064, October 1989.

[9] A. G. Mamistvalov. n-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(8):819–831, August 1998.

[10] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing*, 54(5):438–460, 1992.

[11] F. A. Sadjadi and E. L. Hall. Three-dimensional moment invariants. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-2(2):127–136, March 1980.

[12] F. Solina and R. Bajcsy. Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

[13] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH'94 Computer Graphics Proceedings, Annual Conference Series*, pages 311–247, 1994.

[14] P. Whaite and F. P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049, October 1991.

[15] M. Y. Zarrugh. Display and inertia parameters of superellipsoids as generalized constructive solid geometry primitives. In *Proceedings of the 1985 ASME International Computers in Engineering Conference and Exhibition*, volume 1, pages 317–328, 1985.

Kernel representation of the Kesler construction for Multi-class SVM classification *

Vojtěch Franc, Václav Hlaváč

Czech Technical University, Faculty of Electrical Engineering

Department of Cybernetics, Center for Machine Perception

121 35 Prague 2, Karlovo náměstí 13, Czech Republic

phone: +420 2 24357637, fax: +420 2 24357385

e-mail: {xfrancv,hlavac}@cmp.felk.cvut.cz

Abstract

We propose a transformation from the multi-class SVM classification problem to the single-class SVM problem which is more convenient for optimization. The proposed transformation is based on simplifying the original problem and employing the Kesler construction which can be carried out by the use of properly defined kernel only. The experiments conducted indicate that the proposed method is comparable with the one-against-all decomposition solved by the state-of-the-art SMO algorithm.

1 Introduction

The standard Support Vector Machines (SVM) [8] are designed for dichotomic classification problem (two classes only, called also binary classification). The multi-class classification problem is commonly solved by a decomposition to several binary problems for which the standard SVM can be used. For instance, one-against-all (1-a-a) decomposition is often applied. In this case the classification problem to k classes is decomposed to k dichotomic decisions $f_m(x)$, $m \in K = \{1, \ldots, k\}$, where the rule $f_m(x)$ separates training data of the m-th class from the other training patterns. The classification of a pattern x is performed according to maximal value of functions $f_m(x)$, $m \in K$, i.e., the label of x is computed as $\mathrm{argmax}_{m \in K} f_m(x)$.

For the SVM, however, the multi-class problem can be solved directly [8, 9]. Let us consider that we are given labelled training patterns $\{(x_i, y_i) \colon i \in I\}$, where a pattern x_i is from an n-dimensional space $\mathcal{X}$ and its label attains a value from a set K. The $I = \{1, \ldots, l\}$ denotes a

*Our research was by the European Union under project IST-2001- 32184, by the Czech Ministry of Education under projects MSM 212300013, MSMT Kontakt ME412, and by the Grant Agency of the Czech Republic under project GACR 102/00/1679.

set of indices. The linear classification rules $f_m(x) = \langle w_m, x\rangle + b_m$, $m \in K$ (the dot product is denoted by $\langle ., .\rangle$) can be found directly by solving the multi-class SVM problem

$$\begin{array}{c} \min\limits_{w,b,\xi} \frac{1}{2} \sum\limits_{m\in K} ||w_m||^2 + C \cdot \sum\limits_{i\in I} \sum\limits_{m\in K\setminus\{y_i\}} (\xi_i^m)^d \ , \\ \text{s.t.} \quad \langle w_{y_i}, x_i\rangle + b_{y_i} - (\langle w_m, x_i\rangle + b_m) \geq 1 - \xi_i^m \ , \\ \xi_i^m \geq 0 \ , \quad i \in I \ , m \in K \setminus \{y_i\} \ . \end{array} \tag{1}$$

Similarly to the dichotomic SVM, the minimization of the sum of norms $\|w_m\|^2$ leads to maximization of the margin between classes. For a non-separable case, the sum of $(\xi_i^m)^d$ weighted by a regularization constant C means that the cost function penalizes misclassification of training data. The linear $(d = 1)$ or quadratic $(d = 2)$ cost functions are often used.

To employ kernel functions [8] into non-linear classification rules $f_m(x)$, one has to formulate a dual form of the multi-class SVM decision (1) which is defined as [8, 9]

$$\min_{\alpha} \sum_{i\in I}\sum_{j\in I}(\tfrac{1}{2}c_j^{y_i} A_i A_j - \sum_{m\in K} \alpha_i^m \alpha_j^{y_i} + \tfrac{1}{2} \sum_{m\in K} \alpha_i^n \alpha_j^m) k(x_i, x_j) - 2 \sum_{i\in I} \sum_{m\in K} \alpha_i^m \ ,$$

$$\begin{array}{c} \text{s.t.} \quad \sum\limits_{i\in I} \alpha_i^m = \sum\limits_{i\in I} c_i^m A_i \ , m \in K \ , \\ 0 \leq \alpha_i^m \leq C \ , \alpha_i^{y_i} = 0, \\ A_i = \sum\limits_{m\in K} \alpha_i^m \ , c_j^{y_i} = \begin{cases} 1 & \text{if} \quad y_i = y_j \ , \\ 0 & \text{if} \quad y_i \neq y_j \ , \end{cases} \\ i \in I \ , m \in K \ . \end{array} \tag{2}$$

The dual problem (2) has $k \cdot l$ variables and l of them are always zero. The number of variables is too large in practical problems and consequently it is very difficult to solve the dual quadratic problem directly. There is a solution which employs a decomposition method and solves series of smaller quadratic problems. However, the constraints of the problem (2) are too complicated to allow direct use of efficient decomposition methods developed for dichotomic decision problems, e.g., the Sequential Minimal Optimizer (SMO) algorithm [6].

We propose (i) to modify slightly the original problem (1) by adding the term $(1/2) \sum_{m\in K} b_m^2$ to the objective function, and (ii) to transform the modified problem to the single-class SVM problem which is considerably simpler than the previous formulation. Efficient algorithms can be used to solve the new problem. Moreover, the proposed transformation can be performed by the properly defined kernel function only. The addition of the $(1/2)\, b$ term in the objective function was suggested by Mangasarian [5] for the dichotomic problem. Solutions of the modified problem mostly coincides with the solutions of the original problem [5]. According to the Mangasarian: "For $1,000$ randomly generated problems (dimension $n = 5$ and 40 patterns in the training set) with same C, only 34 cases had solution vector w that differed by more than 0.001 in their 2-norm".

The following section describes proposed approach in details.

2 From multi-class SVM to single-class SVM

We consider modified multi-class SVM where the $(1/2)\,b^2$ is added to the objective function of the (1) which leads to

$$\begin{aligned} &\min_{w,b,\xi} \tfrac{1}{2} \sum_{m\in K} (||w_m||^2 + b^2) + C \cdot \sum_{i\in I} \sum_{m\in K\setminus\{y_i\}} (\xi_i^m)^d \,, \\ &\text{s.t.} \\ &\langle w_{y_i}, x_i\rangle + b_{y_i} - (\langle w_m, x_i\rangle + b_m) \geq 1 - \xi_i^m \,, \\ &\xi_i^m \geq 0 \,, \quad i \in I \,, m \in K \setminus \{y_i\} \,. \end{aligned} \tag{3}$$

We name the problem (3) defined above as the multi-class BSVM problem (B stands for the added bias). Next we introduce a transformation which translates the multi-class BSVM problem (3) to the single-class SVM problem. The single-class SVM problem is defined as

$$\begin{aligned} &\min_{w,\xi} \tfrac{1}{2}||w||^2 + C \sum_{i\in I} (\xi_i)^d \,, \\ &\text{s.t.} \quad \langle w, z_i\rangle \geq 1 - \xi_i \,, \quad i \in I \,. \end{aligned} \tag{4}$$

This problem (4) can be already solved by algorithms which are considerably simpler than the original problems (1) or (3). The dual form of the problem (4) with the linear cost function $d = 1$ is

$$\begin{aligned} &\max_{\alpha} \sum_{i\in I} \alpha_i - \tfrac{1}{2} \sum_{i\in I} \sum_{j\in I} \alpha_i \cdot \alpha_j \cdot k(z_i, z_j) \,, \\ &\text{s.t.} \qquad 0 \leq \alpha_i \leq C \,, \quad i \in I \,, \end{aligned} \tag{5}$$

where $k(z_i, z_j)$ was substituted for the dot products $\langle z_i, z_j\rangle$. The case with the quadratic cost function $d = 2$ can be solved as the separable case using the kernel function $k'(x_i, x_j) = k(x_i, x_j) + \delta_{i,j} \cdot \frac{1}{2C}$. The dual form of the separable case is the same as the problem (5) up to the constraints which simplify to $0 \leq \alpha_i$. We will describe two simple algorithms for solving the single-class SVM problem in Section 3.

The transformation from the multi-class BSVM problem to the single-class SVM problem is based on the Kesler's construction [1]. This construction maps the input n-dimensional space $\mathcal{X}$ to a new $(n+1)\cdot k$-dimensional space $\mathcal{Y}$ where the multi-class problem appears as the single-class problem. Each training pattern x_i is mapped to new $(k-1)$ patterns z_i^m, $m \in K \setminus \{y_i\}$ defined as follows. Let us assume that coordinates of z_i^m are divided into k slots. If each slot $z_i^m(j)$, $j \in K$ has $n+1$ coordinates then

$$z_i^m(j) = \begin{cases} [x_i, 1] \,, & \text{for} \quad j = y_i \,, \\ -[x_i, 1] \,, & \text{for} \quad j = m \,, \\ 0 \,, & \text{otherwise.} \end{cases} \tag{6}$$

We seek a vector w composed of vectors $w_1, \ldots, w_k$ and thresholds $b_1, \ldots, b_k$ in the new space $\mathcal{Y}$ as

$$w = [[w_1, b_1], [w_2, b_2], \ldots, [w_k, b_k]] \,. \tag{7}$$

For instance, when $k = 4$ and $y_i = 3$ then the vectors z_i^m, $m = 1, 2, 4$ are constructed as

$$\begin{array}{rcccccc} z_i^1 & = & [& -[x_i, 1] & 0 & [x_i, 1] & 0 &] \\ z_i^2 & = & [& 0 & -[x_i, 1] & [x_i, 1] & 0 &] \\ z_i^4 & = & [& 0 & 0 & [x_i, 1] & -[x_i, 1] &] \end{array}$$

Performing the transformation (6) we obtain a set $\{z_i^m : i \in I\, m \in K \setminus \{y_i\}\}$ containing $(k-1) \cdot l$ vectors. Each constraint of the multi-class BSVM problem can be expressed as $\langle w, z_i^m \rangle \geq 1 - \xi_i^m$ using the transformed vectors. It is obvious that by substituting w to the objective function of the single-class SVM problem the objective function (4) becomes equivalent to the objective function (3) of the multi-class BSVM. Consequently, the multi-class BSVM problem can be equivalently expressed as the single-class SVM problem,

$$\begin{array}{l} \min\limits_{w} \frac{1}{2}||w||^2 + C \cdot \sum\limits_{i \in I} \sum\limits_{m \in K \setminus \{y_i\}} (\xi_i^m)^d \,, \\ \text{s.t.} \quad \langle w, z_i^m \rangle \geq 1 - \xi_i^m \,, \\ \quad i \in I \,, m \in K \setminus \{y_i\} \,. \end{array} \tag{8}$$

At a first look the introduced transformation seems to be intractable because of increased dimension. However, in the dual form in which the data appears in terms of dot products only the transformation can be performed by introducing a properly defined kernel function.

Let z_i^m and z_j^n be two vectors from $\mathcal{Y}$ created by the transformation (6). Note that the vector z_i^m has the y_i-th coordinate slot equal to $[x_i, 1]$, the m-th slot equal to $-[x_i, 1]$, and remaining coordinates equal to zero. The vector z_j^n is created likewise. Consequently, the dot product $\langle z_i^m, z_j^n \rangle$ is equal to the sum of dot products between $[x_i, 1]$ and $[x_j, 1]$ which occupy the same coordinate slot. The sign of these dot products is positive if $y_i = y_j$ or $m = n$ and negative if $y_i = n$ or $y_j = n$. If all the numbers y_i, y_j, m, and n differ then the dot product is equal to zero. The construction of the dot product $\langle z_i^m, z_j^n \rangle$ can be easily expressed using the Kronecker delta, i.e., $\delta(i, j) = 1$ for $i = j$, and $\delta(i, j) = 0$ for $i \neq j$. The dot product between z_i^m and z_j^n is

$$\langle z_i^m, z_j^n \rangle = (\langle x_i, x_j \rangle + 1) \cdot (\delta(y_i, y_j) + \delta(m, n) - \delta(y_i, n) - \delta(y_j, m)) \,.$$

The dot products $\langle x_i, x_j \rangle$ are replaced by the kernel function $k(x_i, x_j)$ in the non-linear case. The kernel function $k'(z_i^m, z_j^n)$ involving transformations (6) and non-linear case is constructed as

$$k'(z_i^m, z_j^n) = (k(x_i, x_j) + 1) \cdot (\delta(y_i, y_j) + \delta(m, n) - \delta(y_i, n) - \delta(y_j, m)) \,. \tag{9}$$

It implies that solving the dual form (5) of the single-class SVM problem with the kernel (9) is equivalent to solving the dual form of the multi-class BSVM problem (3). As the result of the dual single-class problem we obtain a set of α_i^m, $i = 1, \ldots$, $m = 1, \ldots, k$, $m \neq y_i$ multipliers corresponding to the transformed vectors z_i^m. These multipliers α_i^m determine the vectors w_m and thresholds b_m which can be obtained by reverting the transform (7).

The normal vector w in the transformed space $\mathcal{Y}$ is equal to $w = \sum_{i \in I} \sum_{m \in K \setminus \{y_i\}} z_i^m \alpha_i^m$. The vector $w_j \in \mathcal{X}$ occupies the j-th coordinate slot and is determined by the weighted sum of

vectors z_i^m which have the non-zero j-th coordinate slot, so that

$$\begin{aligned} w_j &= \sum_{i \in I} \sum_{m \in K \setminus \{y_i\}} x_i \alpha_i^m (\delta(j, y_i) - \delta(j, m)) \,, \\ b_j &= \sum_{i \in I} \sum_{m \in K \setminus \{y_i\}} \alpha_i^m (\delta(j, y_i) - \delta(j, m)) \,, \end{aligned}$$

holds. To classify the pattern x in the non-linear case there is need to evaluate $f_j = \langle w_j, \phi(x) \rangle + b_j$ which is equal to

$$f_j(x) = \sum_{i \in I} k(x_i, x) \sum_{m \in K \setminus \{y_i\}} \alpha_i^m (\delta(j, y_i) - \delta(j, m)) + b_j \,.$$

3 Algorithms to the single-class SVM problem

The introduced kernel allows us to solve the multi-class BSVM problem by the use of algorithms solving the single-class SVM problem. Many efficient optimization algorithms for the two-class problem can be readily modified to solve the one-class problem. We have conducted several experiments (see Section 4) using the modified Sequential Minimal Optimizer (SMO) [6] and the kernel Schlesinger-Kozinec algorithm [3].

The SMO for the single-class SVM problem can modify only one Lagrangian at a time since the dual form does not contain the equality constrains. The framework of the modified algorithm is preserved from the original one.

The kernel Schlesinger-Kozinec algorithm solves the two-class SVM problem with quadratic cost function. This problem is transformed to the equivalent problem where the nearest points from the convex hulls are sought. This transformed probllem can be solved by a simple iterative procedure. The nearest point from the origin to one convex hull is sought in the modification to the single-class SVM problem. We used the modified kernel Schlesinger-Kozinec's algorithm to train the multi-class BSVM problem with quadratic cost function and the modified SMO algorithm for the linear cost function.

The implementation of both algorithms in Matlab is available [2].

4 Experiments

We tested the proposed method on the benchmark data sets selected from the UCI data repository [7] and Statlog data collection. We scaled all the data to range $[-1, 1]$. Table 1 summarizes the data sets used.

As a comparative approach we used the one-against-all decomposition and the SMO [6] algorithm for learning the decomposed dichotomic SVM problems which we denote 1-a-a SMO. To solve the single-class problem obtained employing the proposed kernel we used (i) the simplified SMO algorithm denoted as M-1-SMO and (ii) the kernel Schlesinger-Kozinec algorithm denoted as M-1-KSK both mentioned in Section 3.

We trained the classifiers using the Radial Basis Function (RBF) kernel $k(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2 \cdot \sigma}}$ with the $\sigma = \{2^{-3}, 2^{-2}, \ldots, 2^3\}$ and the regularization constant $C = \{2^0, 2^1, \ldots, 2^7\}$. Each

Table 1: Benchmark datasets used for testing.

	number of patterns	number of classes	number of attributes
iris	150	3	4
wine	178	3	13
glass	214	6	13
thyroid	215	3	3

Table 2: Results of comparison on the benchmark datasets. Measured: testing classification error **CE** $[\%]$, training **time** $[s]$ and number of support vectors **SVs**.

		1-a-a SMO	M-1-SMO	M-1-KSK
iris	CE (C,σ)	2.7 $(2^7,2^0)$	**2.0**$(2^5,2^0)$	**2.0** $(2^4,2^0)$
	time	**0.12**	0.22	0.44
	SVs	**17**	30	19
wine	CE (C,σ)	**1.1** $(2^5,2^3)$	2.3 $(2^6,2^3)$	1.7 $(2^1,2^1)$
	time	**0.2**	0.67	0.40
	SVs	54	**37**	54
glass	CE(C,σ)	37.0 $(2^5,2^{-1})$	**28.7** $(2^3,2^{-2})$	31.1 $(2^0,2^{-2})$
	time	14.10	4.06	**1.37**
	SVs	**150**	167	177
thyroid	CE(C,σ)	2.3 $(2^4,2^0)$	2.7 $(2^1,2^{-1})$	**1.8**$(2^0,2^{-1})$
	time	0.41	**0.13**	0.31
	SVs	**35**	43	66

from the 7×8 pairs of (σ, C) was evaluated using 10-fold cross validation method. The parameters which yielded the best average testing error rate are enlisted in Table 2. We also measured average values of (i) the number of support vectors and (ii) the training time on training time on Pentium PIII/750Mhz and (ii) number of support vectors.

5 Conclusions and future work

We propose a transformation from the multi-class SVM classification problem (1) to the single-class SVM problem (4) for which efficient optimization algorithms exist. First the original problem is slightly modified by adding the term $(1/2)\sum_{m\in K} b_m^2$ (similarly to Mangasarian [5] in the dichotomic problem). Then the modified problem is transformed to the single-class SVM problem which is carried out by the use of a properly defined kernel function only.

The experiments conducted indicate that the proposed method is comparable with the one-against-all decomposition solved by the state-of-the-art SMO algorithm. It is worthwhile to

investigate the proposed kernel with other efficient algorithms which can solve the single-class problem, e.g. the Nearest Point Algorithm [4] or the Successive Overrelaxation (SOR) algorithm [5].

References

[1] O. Duda, R., E. Hart, P., and G. Stork, D. *Pattern Recognition*. John Wiley & Sons, 2000.

[2] V. Franc and V. Hlavač. Statistical pattern recognition toolbox for Matlab, 2000-2001. `http://cmp.felk.cvut.cz`.

[3] Vojtěch Franc and Václav Hlaváč. A simple learning algorithm for maximal margin classifier. In A. Leonardis and H. Bischof, editors, *Kernel and Subspace Methods for Computer Vision*, pages 1–11, Vienna, Austria, August 2001. TU Vienna.

[4] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11(1):124–136, January 2000.

[5] L. Mangasarian, O. and R. Musicant, D. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10(5), 1999.

[6] J.C. Platt. Fast training of support vectors machines using sequential minimal optimization. In B. Scholkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods*. MIT Press, Cambridge, MA., USA, 1998.

[7] UCI-benchmark repository of artifical and real data sets. University of California Irvine, `http://www.ics.uci.edu/~mlearn`.

[8] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[9] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

Gradient Eigenspaces for Robust Recognition*

Horst Wildenauer[1], Thomas Melzer[1] and Horst Bischof[2]

[1]Pattern Recognition and Image Processing Group 183/2

Institut for Computer Aided Automation

Vienna University of Technology

Favoritenstrasse 9/1832, A-1040 Vienna, Austria

e-mail: {wilde, melzer}@prip.tuwien.ac.at

[2]ICG

Technical University Graz

Institute for Computer Graphics and Vision

Inffeldgasse 16 2. OG, A-8010 Graz, Austria

e-mail: bis@icg.tugraz.ac.at

Abstract

In the recent literature, gradient-based (filtered) eigenspaces have been used as a means to achieve illumination insensitivity. In this paper, we show that filtered eigenspaces are also inherently robust w.r.t. (non-Gaussian) noise and occlusions. We argue that this robustness stems essentially from the *sparseness of representation* and insensitivity w.r.t. shifts in the mean value. This is also demonstrated experimentally using examples from the field of object recognition and pose estimation.

1 Introduction

Since its inception in the early 1990s, the eigenspace approach to object recognition has received much interest in the vision and object recognition community and is still a very active research topic. It has successfully been employed in various applications such as face recognition [7], illumination planning [4], visual inspection and even visual servoing [5].

The key idea is to represent each object as collection of labelled $(m \times n)$-images, which show the object under envisaged viewing conditions. In the eigenspace approach, PCA is performed on a set of N mean-normalized training images

$$\begin{aligned} \tilde{\mathbf{X}} &= (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N) \\ &= (\mathbf{x}_1 - \hat{\mathbf{m}}, \ldots, \mathbf{x}_N - \hat{\mathbf{m}}) \end{aligned} \tag{1}$$

*This work was supported by the FWF under grant no. P13981-INF.

(whereby $\hat{\mathbf{m}}$ is the estimated mean and $\mathbf{x}_i, \hat{\mathbf{m}} \in \mathbb{R}^{(m \times n)}$) to find a low-dimensional representation of the object. If the eigenvectors of the estimated covariance matrix $\frac{1}{N-1}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ are given by $(\mathbf{e}_1, \ldots, \mathbf{e}_N)$ (whereby we assume that the $\mathbf{e}_i$ are sorted in decreasing order according to their eigenvalues), each mean-normalized input image $\tilde{\mathbf{x}}$ is approximated as linear combination of the first $k << N$ eigenvectors (eigenimages):

$$\tilde{\mathbf{x}} \approx \hat{\mathbf{x}} = \sum_{i=1}^{k} c_i \mathbf{e}_i. \tag{2}$$

The coefficients $\mathbf{c} = (c_1, \ldots, c_k)$ of the linear expansion of $\tilde{\mathbf{x}}$ in terms of $(\mathbf{e}_1, \ldots, \mathbf{e}_k)$ are the eigenspace representation of $\tilde{\mathbf{x}}$.

The purpose of this feature extraction step is twofold: First, it reduces the amount of data needed to represent a single observation (i.e., image) by a factor $\frac{k}{N}$. Second, PCA identifies the directions that explain best the variability between different object views (in the correlation sense) and thus carry the information most relevant to a subsequent classification stage; indeed, it can be shown that among all orthonormal transformations, PCA is optimal in the sense that it minimizes, in the mean square sense, the expected reconstruction error between the original signal $\tilde{\mathbf{x}}$ and the signal $\hat{\mathbf{x}}$ reconstructed from its low-dimensional representation $\mathbf{c}$.

Traditionally, the coefficients c_i in Eq. 2 are obtained as the projections of $\tilde{\mathbf{x}}$ onto the first k eigenimages. Although this approach is conceptually simple and elegant, it can not deal with noise and occlusions. Similarly, it can not handle changes in the object's appearance due to varying illumination, unless these illuminations effects are explicitly incorporated into the eigenspace model (however, this would require that each object pose is acquired under all possible illumination conditions, which is not feasible in most cases).

In this paper, we will discuss an eigenspace approach based on gradient filters, which is not only insensitive to varying illumination, but also inherently robust w.r.t. noise and occlusions. In section 2, we will briefly review the relevant theory, while in section 3, we will demonstrate the performance of our approach experimentally. Conclusions are given in Section 4.

2 Gradient-Based Eigenspaces

The fact that edges are relatively insensitive to the effects of varying illumination has motivated several different implementations of gradient-based eigenspaces (e.g., [8, 2]). Our approach, which was introduced in [1], is based on the observation that a filtered eigenspace representation can be obtained **by filtering the original eigenimages** rather than performing PCA on filtered versions of the training images: Let f be a linear filter and let $*$ denote the convolution operator. Since convolution is a distributive (linear) operation, we have (cf. Eq. 2)

$$(f * \tilde{\mathbf{x}}) \approx \sum_{i=1}^{k} c_i (f * \mathbf{e}_i). \tag{3}$$

Note that Eq. 3 implies that the coefficient vector $\mathbf{c}$ in Eq. 2 can be retrieved from the filtered input image and the filtered eigenimages. However, since the filtering process will, in general,

not preserve the orthogonality of the eigenimages, the coefficients can no longer be obtained as projections of the input image onto the eigenimages. Instead, we compute them by solving (in the least squares sense) the overdetermined system of $(m \times n)$ linear equations Eq. 3. In our experiments, we used a set of six steerable gradient filters [6]. Since, in our approach, the coefficients are not affected by the choice of the filter, we can combine the resulting six sets of equations into one large system consisting of $m \times n \times 6$ equations and solve them simultaneously.

In [1], it was shown that filtered PCA, combined with the robust approach discussed in [3], is insensitive to illumination effects and can tolerate significant levels of noise and occlusion. In this paper, we focus on the properties of filtered PCA itself and show that a gradient-based eigenspace representation, in addition to being illumination insensitive, is also **inherently robust** w.r.t. noise and occlusions.

Figure 1: The 5 objects used in the experiments.

3 Experiments

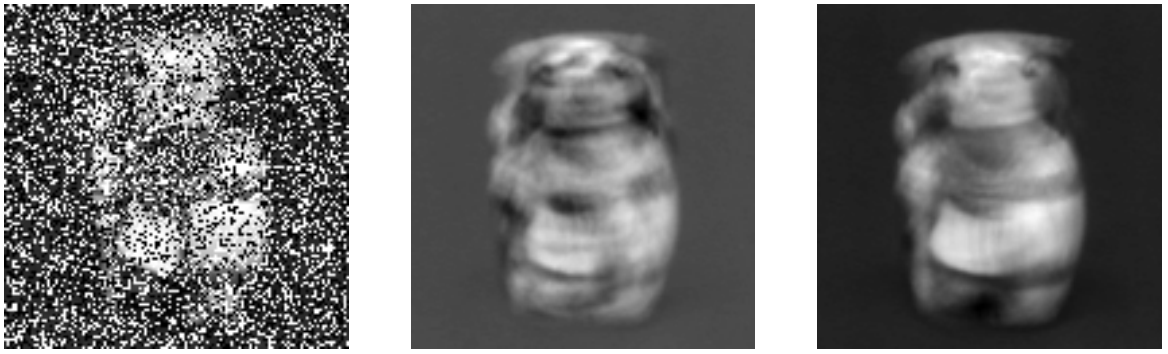
We have extensively evaluated our algorithm on a database of 5 objects (Fig. 1), whereby each object is represented by 72 views; these views were acquired in 5 degree-increments by placing the object on a turntable. Only 36 views were used to build the eigenspace; unless stated otherwise, the eigenspace dimension (number of eigenvectors) was 30. As performance criteria, we used the coefficient error (L2-distance between true and recovered coefficients) and the recognition rate.

Salt and Pepper Noise To have a systematic performance evaluation we took the total set of 360 images and added salt and pepper noise ranging from 0 to 90 percent. This experiment was repeated 10 times.

An example of a reconstruction of a noisy image using the coefficients obtained by the standard and the filtered eigenspace method can be seen in Figure 2. Fig. 3(a) shows the average coefficient error and its standard deviation for the filtered and the standard approach. In Fig. 3(b) the obtained recognition rates are depicted.

Occlusions In this experiment, we used again all 360 images and distorted them with homogeneous occlusions with grayvalues ranging from 0 to 1 (maximum brightness). The size of the occlusions was increased from 0 to 90 percent of the object's size.

In Figure 4 the reconstruction of a 40% occluded image using the standard and the filtered method can be seen. Fig. 5 shows the coefficient errors obtained with the filtered and standard method, while the corresponding recognition rates can be seen in Fig. 6. Note the high sensitivity of the standard approach w.r.t. the grayvalue of the occlusion.

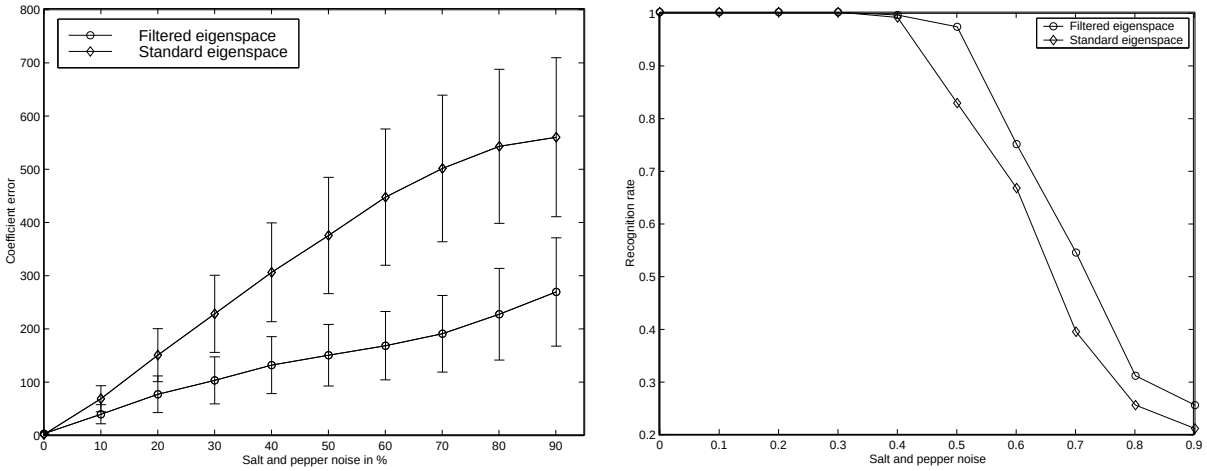


(a) Image with salt and pepper noise

(b) Reconstruction of (a) using the standard method

(c) Reconstruction of (a) using the filtered method

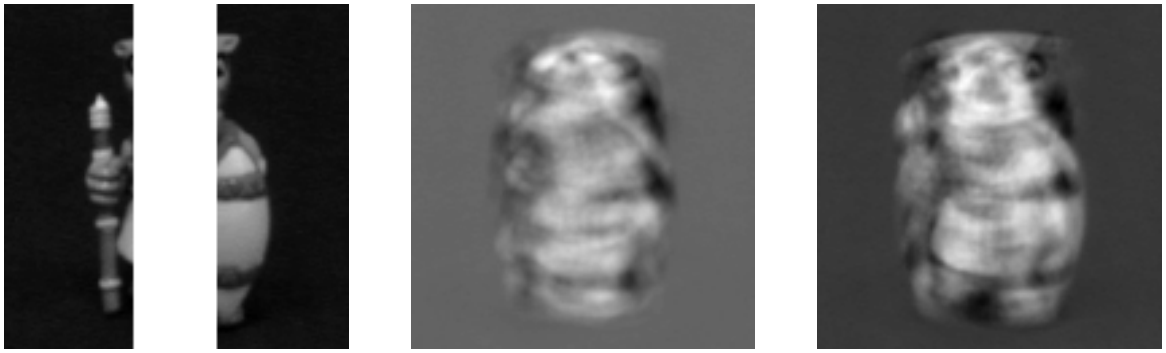
Figure 2: Demonstration of the influence of 50% salt and pepper noise on the standard and the filtered eigenspace.



(a) Coefficient error

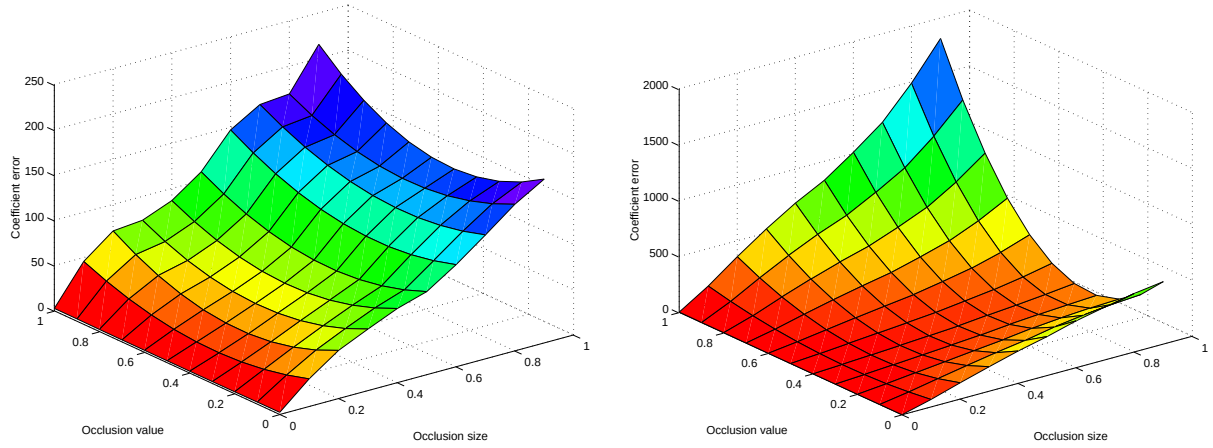
(b) Recognition rate

Figure 3: Influence of salt and pepper noise on the standard and the filtered eigenspace.



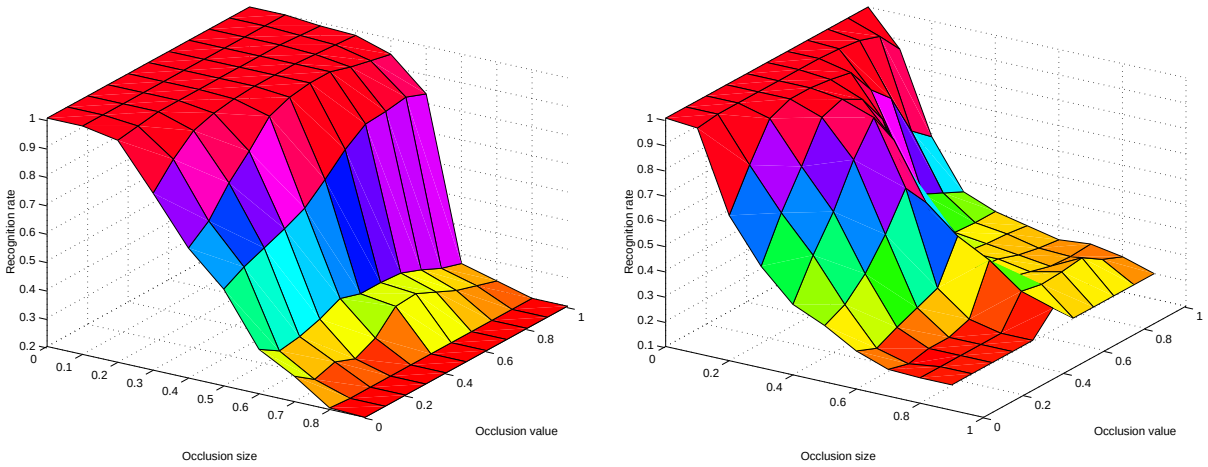
(a) Occluded input image (b) Reconstruction of (a) using the standard method (c) Reconstruction of (a) using the filtered method

Figure 4: Demonstration of the effects of a 40% occlusion on the standard and the filtered eigenspace.



(a) Filtered eigenspace (b) Standard eigenspace

Figure 5: Comparison of coefficient errors for occluded images.



(a) Filtered eigenspace (b) Standard eigenspace

Figure 6: Comparison of recognition rates for occluded images.

4 Discussion

In the previous section, we have shown experimentally that our version of filtered PCA compares favorably to the standard eigenspace approach. The experiments indicate that filtered PCA is not only illumination-insensitive (as discussed in [1]), but also inherently robust w.r.t. non-Gaussian noise and occlusions. This robustness stems essentially from the following two properties of filtered PCA:

Sparseness of Representation The gradient-filtered eigenimages and the gradient-filtered input images encode high-frequency information. However, unless the appearance of the training objects is dominated by high-frequency texture or structure elements, the number of such edge pixels will be small compared to the size of the eigenimages. Consequently, distortions in the input image due to occlusions or noise are likely to result in spurious gradients that will not coincide with the true edges encoded in the eigenspace model. Due to the sparse nature of gradient maps, such spurious gradients will be orthogonal (or almost orthogonal) to the eigenspace and thus will have no (or only small) influence on the solution vector $\mathbf{c}$.

Insensitivity w.r.t. the Mean It has been standard practice in the eigenspace approach to rescale the input image $\mathbf{x}$ to unit length before subtracting the mean-image. This makes the approach insensitive to changes in the overall brightness level and ensures that the pre-stored mean-image $\hat{\mathbf{m}}$ and the actual input image are of comparable magnitude. However, uneven illumination, saturation effects or large occlusions will pull away the computed scale factor $s = \frac{1}{||\mathbf{x}||}$ from its true value, which can make the subtraction of $\hat{\mathbf{m}}$ practically meaningless (for too large s) or, worse, make the relative magnitude of the input image so small that $\tilde{\mathbf{x}} = s\mathbf{x} - \hat{\mathbf{m}}$ is dominated by $\hat{\mathbf{m}}$, and not by the actual observation.

Filtered PCA is far less susceptible to such *mean-effects* than other eigenspaces approaches. This is due to the fact that gradient filters basically compute the difference of gray-values in a small local neighborhood; the mean, however, will be relatively homogeneous in such small neighborhoods and therefore will be canceled by the subtraction.

References

[1] Horst Bischof, Horst Wildenauer, and Aleš Leonardis. Illumination insensitive eigenspaces. In *Proc. of IEEE International Conference on Computer Vision*, pages 233–238, 2001.

[2] J. Hornegger, H. Niemann, and R. Riesack. Appearance-based object recognition using optimal feature transforms. *Pattern Recognition*, 33(2):209–224, 2000.

[3] Aleš Leonardis and Horst Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 1:99–118, 2000.

[4] Hiroshi Murase and Shree K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, December 1994.

[5] Shree K. Nayar, Sameer A. Nene, and Hiroshi Murase. Subspace methods for robot vision. *IEEE Trans. Robotics and Automation*, 12(5):750–758, October 1996.

[6] E. Simoncelli and H. Farrid. Steerable wegde filters for local orientation analysis. In *IEEE Trans. on Image Processing*, pages 1–15, 1996.

[7] Matthew Turk and Alexander P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 4(1):71–86, 1991.

[8] A. Yilmaz and M. Gokmen. Eigenhill vs. eigenface and eigenedge. *Pattern Recognition*, 34(1):181–184, 2001.

An Improved Energy Minimization for Deformable Templates

Andrea Scaggiante* Massimo Zampato* Samuele Dal Bello° Giuseppe Marchiori°

* Tecnomare S. p. A.

San Marco 3584,

Venice

Tel. +39041796711

Fax +390415230363

° Consorzio RFX

EURATOM-ENEA Association for Fusion

Corso Stati Uniti 4 Padova

Tel. +390498295000

Fax +390498700718

e-mail: scaggiante.a@tecnomare.it, zampato.m@tecnomare.it, dalbello@igi.pd.cnr.it, marchiori@igi.pd.cnr.it

Abstract

This paper addresses the development of a new approach to Deformable Template technique for accurate shape extraction. By implementing a compact and efficient minimization of the template energy, important improvements are obtained concerning with outlier elimination, optical distortion compensation and easy and precise initialization. An exhaustive projective model for planar feature is described, highlighting its employment in a visual servoing application.

1 Introduction

Accurate feature extraction is a mandatory requirement for some Computer Vision applications such as visual servoing. Deformable Template (DT, see [5, 11]) has proven to be a reliable technique for this kind of problems. This approach is rooted in Active Contour Models (ACM or “snakes”, see [6]), consisting of a sampled curve that deforms and adjusts its shape to match the contour of an object in the scene. Bending is driven by the minimization of an energy that depends on the image and the type of target features (edges, lines, uniform region border, etc.). In order to make ACM less sensitive to noise and occlusions, the a priori knowledge of the shape to be extracted has been introduced in the DT approach. Different solutions have been proposed for the model representation, energy minimization, and considered image features.

DT is becoming more and more widespread as an approach to extraction and tracking of complex shapes.

The use of DT for the feature extraction determines a set of advantages resumed here below.

- Introducing the a priori knowledge about the feature shape enhances the algorithm robustness, avoiding the template to fit to local minima in the energy minimization process.
- At the end of the extraction phase, a point in the 2D template can be associated to a point in the 3D feature. This is really useful for applications for 3D-measurement where image segmentation is only a step of the measurement process (e.g. the target pose estimation).

Meaningful parts of the feature can be identified too: for instance, we can extract the points belonging to a circular border of the target without any intervening elaboration.

- The state configuration of a DT can be described by a limited number of parameters instead of an undetermined number of control nodes without reducing the fitting accuracy to the actual feature (on the contrary for an ACM the smaller is the node number, the smaller is the fitting accuracy)

DT has to be implemented for the extraction of well-known features. This type of target is typical of an industrial environment where manifolds have rigid and well-defined shapes. A lot of these 3D features lay on a single plane so that their perspective projection can be modeled by a homographic transformation.

This paper describes a Deformable Template algorithm developed as part of a monocular vision system for the pose estimation of planar objects. This system will be employed in the semi-automatic control of a mechanical arm [3] used for the maintenance of the graphite first wall of the nuclear fusion experiment RFX [4]. The experiment is run by a Consortium including ENEA, CNR and Padua University in the framework of the European research programme on the thermonuclear fusion. Graphite tiles cover the inner surface of the vacuum vessel and have to be removed and substituted in case of plasma damage. The end-effector of the arm is to be exactly driven to the clamping key in the center of the tile (see Figure 1) with a frontal attitude. The template to be extracted consists of the internal circle and the two rectangular holes. A following elaboration step determines the pose of the target. Figure 2 shows a second type of target, consisting of a bush in a vessel stiffening ring where a new tile is to be inserted.

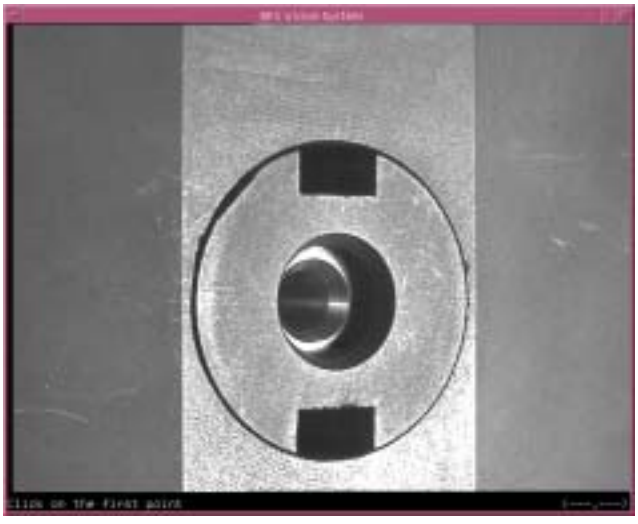


Figure 1 Planar feature of the graphite tile key

For such an application the feature extraction has to be very precise and natural accuracy ensured by DT classical technique could not be enough: few pixel error in the extraction could affect pose result. Furthermore the available image could exhibit some noise elements, e.g. not very sharp edges, reflecting and rough surfaces. The proposed DT approach introduces important improvements aiming at enhancing the accuracy of the method, in order to concern

with outlier discarding, optical distortion, easy and precise manual initialization, efficient energy minimization.
Section 2 describes the type of implemented model. The energy minimization technique, the approach to outlier discarding and the initialization procedure are described in section 3. Section 4 deals with the compensation of the lens distortion. Section 5 reports some test result and section 6 gathers the conclusions.

Figure 2 Planar feature of the bush inserted in the vessel stiffening ring

2 The projective model

Let $\mathbf{X} = [X \quad Y \quad Z \quad 1]^T$ be a point in the absolute reference system, $\mathbf{u} = [u \quad v]^T$ its pixel co-ordinates, R_{3x3} the rotation and t_{3x1} the translation from the co-ordinates in the absolute reference system to the co-ordinates in the camera reference system, fu and fv the focal lengths, r_0 and c_0 the co-ordinates of the principal point. By using the pin hole model of a camera we have

$$\begin{bmatrix} s \cdot u \\ s \cdot v \\ s \end{bmatrix} = \begin{bmatrix} fu & 0 & r_0 \\ 0 & fv & c_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [R_{3x3} \quad t_{3x1}] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

A planar shape can be described by a list of points $\mathbf{X}_i = [X_i \quad Y_i \quad 1]^T$ belonging to its border so that, setting $Z = 0$ in Equation (1) we get[1]:

$$\begin{bmatrix} s \cdot u_i \\ s \cdot v_i \\ s \end{bmatrix} = \begin{bmatrix} fu & 0 & r_0 \\ 0 & fv & c_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [R_{3x3} \quad t_{3x1}] \cdot \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} s \cdot u_i \\ s \cdot v_i \\ s \end{bmatrix} = \begin{bmatrix} fu & 0 & r_0 \\ 0 & fv & c_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{11} & R_{12} & t_{11} \\ R_{21} & R_{22} & t_{21} \\ R_{31} & R_{31} & t_{31} \end{bmatrix} \cdot \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} s \cdot u_i \\ s \cdot v_i \\ s \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{4}$$

Equation (4) proves the image of a planar feature is fully described by a homography; we call this homography the projective model. Points $\mathbf{X}_i$ are the model of our template, nodes $\mathbf{u}_i$ are a particular template configuration; determining this configuration is equivalent to finding out the corresponding matrix $\mathbf{A}$.

3 An improved minimization technique

The ACM minimization approach starts from dynamics: the active contour "lives" on a potential surface and iteratively falls down towards a potential valley.
As far as DT is concerned, the minimization is achieved in a parameter space (see [9, 10]). Different modifications of this technique have been successively introduced. Supposing searching edge features, the forces acting on the nodes can be obtained finding the Point with the Max value of the Module of the intensity Gradient (MMGP) along the normal direction to the template at each node of the template itself in order to attract it towards the MMGPs. This approach is motivated by the aperture problem: the component of motion along an edge can't be locally identified (see [8, 2]). This approach determines a first difference from a complete physical model; in the same direction, the energy to minimize can be changed: it can depend no more on the gradient module but on the direction of the gradient itself and the template

[1] R_{ij} and t_{ij} are the element (i, j) of matrices R_{3x3} and t_{3x1} respectively

normal vector (see [7]). This choice reduces the dependence of the extraction on the strength of the image feature edges.

Following this evolution, our template introduces some improvements to classical minimization approach. The model parameters are updated realizing a least square minimization of the distances between each node $\mathbf{u}_i$ and the corresponding MMGP $\mathbf{u}_{\max i} = \begin{bmatrix} u_{\max i} & v_{\max i} \end{bmatrix}^T$.

By resolving Equation (4) with respect to parameters a_{ij} of matrix $\mathbf{A}$, we have

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X_i u_i & -Y_i u_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -X_i v_i & -Y_i v_i \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{bmatrix} \tag{5}$$

By substituting $\mathbf{u}_{\max i}$ to $\mathbf{u}_i$ in (5) we have

$$\begin{bmatrix} u_{\max i} \\ v_{\max i} \end{bmatrix} = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X_i u_{\max i} & -Y_i u_{\max i} \\ 0 & 0 & 0 & X_i & Y_i & 1 & -X_i v_{\max i} & -Y_i v_{\max i} \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{bmatrix} \tag{6}$$

These relations are collected in a global linear system that is solved to compute matrix $\mathbf{A}$. $\mathbf{A}$ is then applied to each model point $\mathbf{X}_i$ to calculate the new nodes of the template $\mathbf{u}_i$. These steps are repeated iteratively. The final configuration of the template is the configuration corresponding to the minimum value of the following energy

$$E = -\sum_i \left| \mathbf{n}_{T\,i} \bullet \mathbf{n}_{G_i} \right| \tag{7}$$

$\mathbf{n}_{T_i}$ is the unitary vector normal to the template and $\mathbf{n}_{G_i}$ is the unitary vector normal to the intensity gradient.

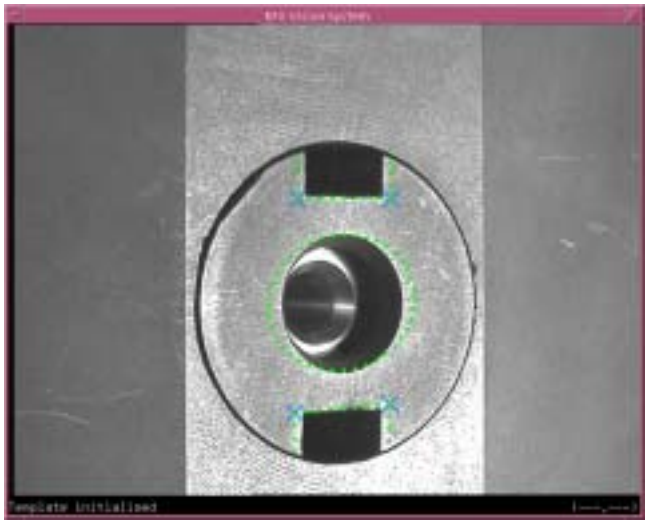


Figure 3 Example of initialization for a tile key template

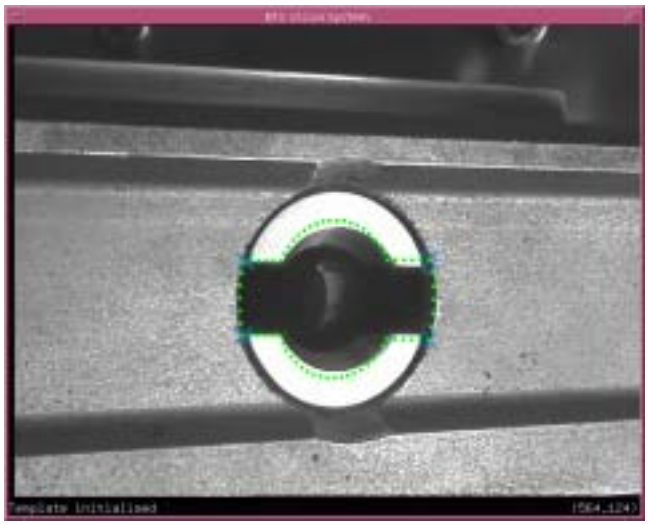


Figure 4 Example of initialization for a bush template

This technique allows discarding outlier MMGPs, so that feature extraction can be more accurate. For outlier discarding, our approach progressively reduces the max-allowed distance between a node and the corresponding MMGP. In this way the DT reaches far edges in the initial iterations but outliers do not draw it away when it is near to the target. The max-allowed distance is quadratically reduced to use more iterations for the final configuration improvement. It is worth noting that using the classical dynamic minimization outliers can not be discarded because in that case the template node distribution has to be symmetrically balanced.

The algorithm can be accurately initialized by putting the co-ordinates of four reference points selected by an operator into Equation (5).

Figure 3 and Figure 4 show two examples of initialization for the considered templates.

4 Taking optical distortion into account

Usually, classical implementations of the DT overlook lens distortion. The proposed approach allows taking it into account by modifying the minimization procedure. Next section describes the extension of the pinhole camera model so as to deal with distortion compensation.

4.1 Camera model with optical distortion compensation

The optical distortion correction can be considered by modifying the pin hole model of Equation (1) as follows:

$$\begin{bmatrix} s\cdot x \\ s\cdot y \\ s \end{bmatrix} = [R_{3x3} \quad t_{3x1}]\cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{cases} x = r + (g_0 + g_2)\cdot r^2 + g_3\cdot r\cdot w + g_0\cdot w^2 + k\cdot r\cdot\left(r^2 + w^2\right) \\ y = w + g_1\cdot r^2 + g_2\cdot r\cdot w + (g_1 + g_3)\cdot w^2 + k\cdot w\cdot\left(r^2 + w^2\right) \end{cases}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} fu & 0 \\ 0 & fv \end{bmatrix}\cdot\begin{bmatrix} r \\ w \end{bmatrix} + \begin{bmatrix} r_0 \\ c_0 \end{bmatrix} \tag{8}$$

g_0, g_1, g_2, g_3 and k are the parameters describing the lens distortion. It is worth noting that any other model of lens distortion can substitute the used model without affecting our approach.

A pixel $\mathbf{u}$ can be transformed to the corresponding point $\mathbf{m}' = [r \quad w]^T$ in the distorted normalized co-ordinate space[2]; compensating the optical distortion we get $\mathbf{m} = [x \quad y]^T$ in the normalized co-ordinate space.

4.2 Deformable Template with optical distortion compensation

In order to compensate the optical distortion, the DT minimization approach can be modified as described in this section.

[2] The normalized co-ordinates are the co-ordinates in the image plane of a virtual camera with unitary focal length.

The projective Equations (4) and (5), that neglect the lens distortion, can be used with no approximation to link point $\mathbf{X}_i$ of the planar object (again we set $Z = 0$) to its normalized coordinates $\mathbf{m}_i$ by using a new matrix $\mathbf{A}'$:

$$\begin{bmatrix} s \cdot x_i \\ s \cdot y_i \\ s \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & t_{11} \\ R_{21} & R_{22} & t_{21} \\ R_{31} & R_{31} & t_{31} \end{bmatrix} \cdot \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \mathbf{A}' \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} X_i & Y_i & 1 & 0 & 0 & 0 & -X_i x_i & -Y_i x_i \\ 0 & 0 & 0 & X_i & Y_i & 1 & -X_i y_i & -Y_i y_i \end{bmatrix} \begin{bmatrix} a'_{11} \\ a'_{12} \\ a'_{13} \\ a'_{21} \\ a'_{22} \\ a'_{23} \\ a'_{31} \\ a'_{32} \end{bmatrix} \tag{10}$$

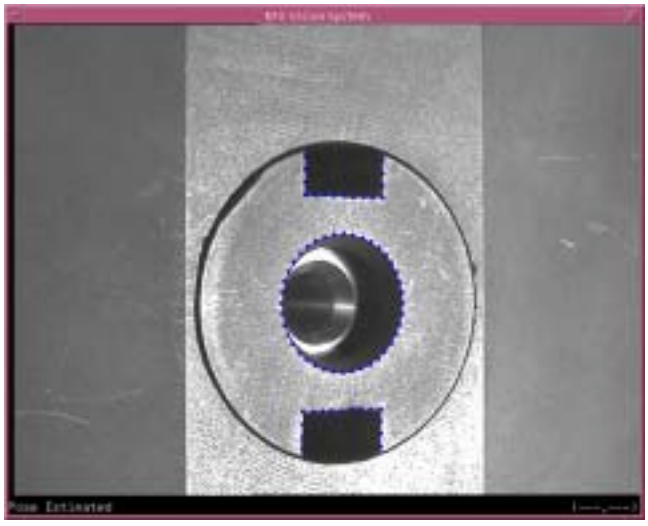


Figure 5 Example of final extraction for the tile key template

The algorithm steps can be described as follows:

- In the initialization step the reference points are transformed to the corresponding normalized points and used to estimate matrix $\mathbf{A}'$.
- For each $\mathbf{X}_i$ the corresponding $\mathbf{m}_i$ is then computed by applying matrix $\mathbf{A}'$.

- From $\mathbf{m}_i$ the corresponding $\mathbf{u}_i$ point is computed. $\mathbf{u}_i$ is used to find out its MMGP and the energy E
- The MMGPs are projected to the normalized co-ordinate space and used to compute new matrix $\mathbf{A}'$.

In the following iterations only last four steps are repeated until convergence is achieved.
Figure 5 and Figure 6 show final extraction results for the target templates.

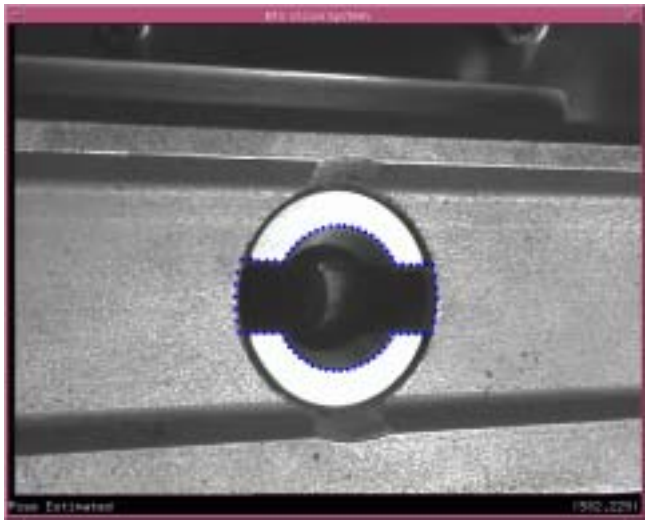


Figure 6 Example of final extraction for the bush template

5 Test results

Some tests have been performed in order to find the algorithm accuracy.

Test case	Mean Error [pixel]
Tile key 30°	1.01
Tile key 60°	1.14
Bush 30°	1.32
Bush 60°	1.19

Table 1 Test results

The poses of two targets at 30° and 60° with respect to the camera frame have been beforehand precisely measured. According to the a priori poses, the planar target points have been projected on the image plane by applying the model of Equation (8) to compare them to the image projection of the same points according to the estimated pose. An average error on the node set has been computed. Table 1 collects the errors found out in the four considered cases; for each case the mean error value on ten consecutive trials with different initializations is reported.

6 Conclusions

This paper has dealt with an improved Deformable Template technique for feature extraction. Main improvements of the proposed approach are the outlier rejection and the optical distortion compensation through a compact and efficient minimization technique. The algorithm has been actually developed for the extraction of a planar image with a template described by a projective model but it can be straightforwardly applied to different template types. An extension to tridimensional feature extraction is very interesting because both image target tracking and feature pose estimation could be achieved in a single elaboration step by using the pose of the target with respect to the camera reference frame as template model descriptor. Multiple camera data fusion could be also implemented in order to improve the precision and reliability of such a 3D technique.

References

[1] Bascle B. and Deriche R., ”Region tracking through image sequences”, INRIA Research No. 2439, December 1994.

[2] Drummond T.W. and Cipolla R., ”Visual tracking and control using Lie algebras”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp.652-657, Fort Collins, CO, USA, June 1999.

[3] Doria A., Finotello R., Gnesotto F., Marchiori G., Perfumo A., ”Construction and Testing of the RFX Remote Handling System”, *Proceedings of the 17th Symposium on Fusion Technology*, Roma, Italy, September 1992.

[4] Gnesotto F., Sonato P., Baker R.W., Elio F., Doria A., Fauri M., Fiorentin P., Marchiori G., Zollino G., ”The plasma system of RFX”, *Fusion Engineering and Design*, Vol. 25, No. 4, pp.335-372, January 1995

[5] Jolly M. P. D., Lakshmanan S. and Jain A. K. , ”Vehicle segmentation and classification using deformable templates”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 3, pp.293-308, March 1996.

[6] Kass M., Witkins A. and Terzopoulos D., ”Snakes: Active Contour Models ”, *International Journal of Computer Vision*, Vol.1, No. 4, pp.321-331, 1988

[7] Lai K.F., ”Deformable contours: modeling, extraction, detection and classification”, Ph.D. thesis, Electrical Engineering, University of Wisconsin-Madison, 1994.

[8] Martin F. and Horaud R., ”Multiple Camera Tracking of Rigid Objects”, INRIA Research No. 4268, September 2001.

[9] Scaggiante A., *Stima della posa mediante inseguimento e ricostruzione di caratteristiche ellittiche da una sequenza di viste monoscopiche,* Electrical Engineering thesis. University of Padua, 1998.

[10] Scaggiante A., Frezza R. and Zampato M., ”Identifying and Tracking Ellipses: a Technique based on Elliptical Deformable Templates”, *Proceedings of the Tenth International Conference on Image Analysis and Processing*, Venice, Italy, September 1999.

[11] Yuille A. and Hallinan P., *Active vision, chapter 2 Deformable templates,* pp.21-38. MITPress, 1992.

On Combining Shape from Silhouette and Shape from Structured Light *

Srdan Tosovic, Robert Sablatnig, and Martin Kampel
Vienna University of Technology,
Institute of Computer Aided Automation,
Pattern Recognition and Image Processing Group
Favoritenstr. 9, 183-2, A-1040 Vienna
e-mail: {tos,sab,kampel}@prip.tuwien.ac.at

Abstract

This paper presents an octree based method of three-dimensional reconstruction of objects using a combination of two different methods, Shape from Silhouette and Shape from Structured Light, focusing on reconstruction of archaeological vessels. Shape from Silhouette is a method suitable for reconstruction of objects with handles, whereas it is unable to reconstruct concavities on an object's surface, such as inside of a bowl. Shape from Structured Light can reconstruct such concavities, but it often creates incomplete models because of camera and light occlusions. The purpose of combining these two methods is to overcome the weaknesses of one method through the strengths of the other, making it possible to construct complete models of arbitrarily shaped objects. The construction is based on multiple views of an object using a turntable in front of stationary cameras. Results of the algorithm developed are presented for both synthetic and real objects.

1 Introduction

Shape from Silhouette is a method of automatic construction of a 3D model of an object based on a sequence of images of the object taken from multiple views, in which the object's silhouette represents the only interesting feature of the image [21, 18]. The object's silhouette in each input image corresponds to a conic volume in the object real-world space. A 3D model of the object can be built by intersecting the conic volumes from all views, which is also called *Space Carving* [12].

Shape from Silhouette can be applied on objects with variety of shapes, including objects with handles, like many archaeological vessels and sherds. However, concavities on an object's

*This work was partly supported by the Austrian Science Foundation (FWF) under grant P13385-INF, the European Union under grant IST-1999-20273 and the Austrian Federal Ministry of Education, Science and Culture.

surface remain invisible for this method, making it unusable for reconstruction of the inside of a bowl or a cup or the inner side of a sherd. Therefore, another method, Shape from Structured Light, is used to discover the concavities.

Shape from Structured Light is a method which constructs a surface model of an object based on projecting a sequence of well defined light patterns onto the object. The patterns can be in the form of coded light stripes [11] or a ray or plane of laser light [13]. The 3D coordinates of the points on the object's surface are recovered using active triangulation [3, 9].

There have been many works on construction of 3D models of objects from multiple views. Baker [1] used silhouettes of an object rotating on a turntable to construct a wire-frame model of the object. Martin and Aggarwal [14] constructed volume segment models from orthographic projection of silhouettes. Chien and Aggarwal [7] constructed an object's octree model from its three orthographic projections. Veenstra and Ahuja [23] extended this approach to thirteen standard orthographic views. Potmesil [18] created octree models using arbitrary views and perspective projection. For each of the views he constructs an octree representing the corresponding conic volume and then intersects all octrees. In contrast to this, Szeliski [21] first creates a low resolution octree model quickly and then refines this model interactively, by intersecting each new silhouette with the already existing model. The last two approaches project an octree node into the image plane to perform the intersection between the octree node and the object's silhouette. Srivastava and Ahuja [20] in contrast, perform the intersections in 3D-space. Niem [15] uses pillar-like volume elements (pillars) rather than octree for model representation. De Bonet and Viola [4] extended the idea of voxel reconstruction to transparent objects by introducing the Roxel algorithm — a responsibility weighted 3D volume reconstruction. Wong and Cipolla [26] use uncalibrated silhouette images and recover the camera positions and orientations from circular motions.

Most laser light based Shape from Structured Light methods use a camera, a calibrated laser ray or plane and a motion platform — usually a linear slide or a turntable. Borgese et al. [5] use a pair of standard video cameras, a laser pointer, and a special hardware that lets the laser spot be detected with high reliability and accuracy. By obtaining a single surface point at each step, this method implies a slow, sparse sampling of the surface. Liska [13] uses two lasers aligned to project the same plane, a camera and a turntable. Using two lasers eliminates some of the light occlusions but not the camera occlusions, resulting in incomplete models for many objects. Park et al. [17] built a DSLS (Dual Beam Structured Light) scanner, which uses a camera mounted on a linear slide and two non-overlapping laser planes, resulting in denser range images. Davis and Chen [8] use two calibrated fixed cameras viewing a static scene and an uncalibrated laser plane which is freely swept over the object.

The work of Szeliski [21] was used as a basis for the Shape from Silhouette and the work of Liska [13] as a basis for the Shape from Structured Light approach presented in this paper.

The paper is organized as follows. Section 2 describes the equipment used for acquisition. Section 3 describes the octree model representation and Section 4 presents the combination strategy proposed. Experimental results with both synthetic and real data are given in Section 5 and at the end of the paper conclusions are drawn and future work is outlined.

2 Acquisition System

The acquisition system consists of the following devices:

- a turntable (Figure 1a) with a diameter of 50 cm, whose desired position can be specified with an accuracy of $0.05°$ (however, the minimal relative rotation angle is $1.00°$).
- two monochrome CCD-cameras with a focal length of 16 mm and a resolution of 768x576 pixels. One camera (*Camera-1* in Figure 1) is used for acquiring the images of the object's silhouettes and the other (*Camera-2* in Figure 1) for the acquisition of the images of the laser light projected onto the object.
- a red laser (Figure 1d) used to project a light plane onto the object. The laser is equipped with a prism in order to span a plane out of the laser beam.
- a lamp (Figure 1e) used to back-light the scene for the acquisition of the silhouette of the object [10]. The object should be clearly distinguishable from the background independent from the object's shape or the type of its surface.

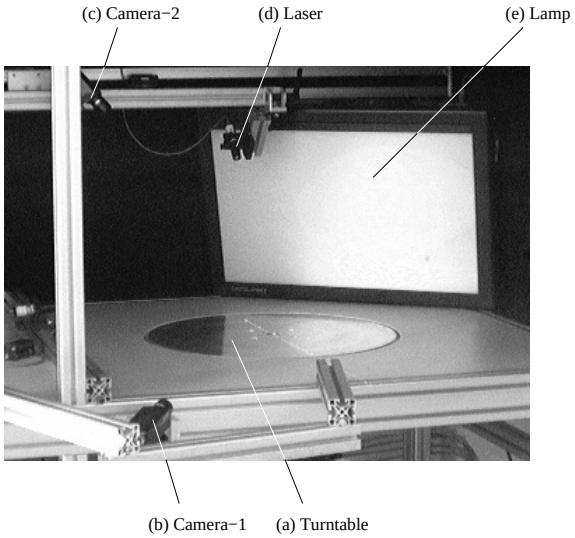


Figure 1: Acquisition System

Both cameras are placed about 50 cm away from the rotational axis of the turntable. Ideally the optical axis of the camera for acquiring object's silhouettes lies nearly in the rotational plane of the turntable, orthogonal to the rotational axis. The camera for acquiring the projection of the laser plane onto the object views the turntable from an angle of about $45°$. The laser is directed in such that the light plane it projects contains the rotational axis of the turntable. The second camera views the light plane also from an angle of about $45°$. The relative position of the two cameras to one another is not important, since the acquisition of the silhouettes and the acquisition of the laser light projection are independent from one another.

Prior to any acquisition, the system is calibrated in order to determine the inner and outer orientation of the camera and the rotational axis of the turntable. We used the calibration technique proposed by Roger Y. Tsai [22], for several reasons: it is efficient and accurate, lens

distortion can be taken into account but also ignored if desired, and there is a publicly available implementation on Internet [25]. In our experiments, the average calibration error was 0.5 pixel or less (measured in the image plane), which is sufficient for our approach, because the smallest unit processed in an image is 1 pixel.

3 Octree Model Representation

There are many different model representations in computer vision and computer graphics used. Here we will mention only the most important ones. Surface-based representations describe the surface of an object as a set of simple approximating patches, like planar or quadratic patches [2]. Generalized cylinder representation [19] defines a volume by a curved axis and a cross-section function at each point of the axis. Overlapping sphere representation [16] describes a volume as a set of arbitrarily located and sized spheres. Approaches such as these are efficient in representing a specific set of shapes but they are not flexible enough to describe arbitrary solid objects. Two of the most commonly used representations for solid volumes are boundary representation (B-Rep) [24] and constructive solid geometry (CSG) [24, 19].

An octree [6] is a tree-formed data structure used to represent 3-dimensional objects. Each node of an octree represents a cube subset of a 3-dimensional volume. A node of an octree which represents a 3D object is said to be:

- *black*, if the corresponding cube lies completely within the object
- *white*, if the corresponding cube lies completely within the background, i.e., has no intersection with the object
- *gray*, if the corresponding cube is a boundary cube, i.e., belongs partly to the object and partly to the background. In this case the node is divided into 8 child nodes (octants) representing 8 equally sized sub-cubes of the original cube

An octree as described above contains binary information in the leaf nodes and therefore it is called a binary octree, and it is suitable for representation of 3D objects where the shape of the object is the only object property that needs to be modeled by the octree. Non-binary octrees can contain other information in the leaf nodes, e.g., the cube color in RGB-space. For the 3D modeling approach presented in this work, a binary octree model is sufficient to represent 3D objects.

The octree representation has several advantages [6]: for a typical solid object it is an efficient representation, because of a large degree of coherence between neighboring volume elements (voxels), which means that a large piece of an object can be represented by a single octree node. Another advantage is the ease of performing geometrical transformations on a node, because they only need to be performed on the node's vertices. The disadvantage of octree models is that they digitize the space by representing it through cubes whose resolution depend on the maximal octree depth and therefore cannot have smooth surfaces. However, this is a problem with any kind of voxel-based volumetric representation.

4 Fusion of the Algorithms

As noted in Section 1, Shape from Silhouette defines a *volumetric* model of an object, whereas Shape from Structured Light defines a *surface* model of an object. The main problem that needs to be addressed in an attempt to combine these two methods is how to adapt the two representations to one another, i.e. how to build a common 3D model representation. This can be done in several ways:

- Build the *Shape from Silhouette*'s volumetric model and the *Shape from Structured Light*'s surface model independently from one another. Then, either convert the volumetric model to a surface model and use a combination of the two surface models to create the final representation or convert the surface model to a volumetric model and use a combination of the two volumetric models to create the final representation. Depending on the properties of the two models (e.g., whether they represent a subset or a superset of the object), their combination can mean their union or their intersection or some more complex operation.
- Use a common 3D model representation from the ground up, avoiding any model conversions. That means either design a volume based Shape from Structured Light algorithm or a surface based Shape from Silhouette algorithm.

Generally, the conversion of a surface model to a volumetric model is a complex task, because if the surface is not completely closed, it is hard to say whether a certain voxel lies inside or outside the object. With closed surfaces one could follow a line in 3D space starting from the voxel observed and going in any direction and count how many times the line intersects the surface. For an odd number of intersections one can say that the voxel belongs to the object. But even in this case there would be many special cases to handle, e.g. when the chosen line is tangential to the object's surface.

This reasoning lead us to the following conclusions:

- Building a separate Shape from Structured Light surface model and a Shape from Silhouette volumetric model followed by converting one model to the other and then combining them is mathematically complex and computationally costly.
- If we want to estimate the volume of an object using our model, any intermediate surface models should be avoided because of the problems of conversion to a volumetric model.

Therefore, our approach proposes building a single volumetric model from the ground up, using both underlying methods in each step (illustrated in Figure 2):

1. Binarize the acquired images for both Shape from Silhouette and Shape from Structured Light in such a way that the white image pixels *possibly* belong to the object and the black pixels *for sure* belong to the background (see Figure 2a). A silhouette binary image is created by extraction of the object's silhouette through simple thresholding of the image. The creation of a structured light binary image is more complex. Based on the known position of the laser, an input image (representing the intersection of the laser plane with the object's surface) is converted to an image approximating intersection of the laser plane with the whole object.
2. Build the initial octree, containing one single root node marked "black". (Figure 2b). This node is said to be at the level 0.

3. All black nodes of the `current level` are assumed to be in a linked list. If there are no nodes in the `current level`, the final model has been build so jump to Step 8. Otherwise, continue with Step 4.
4. Project the `current node` of the `current level` into all Shape from Silhouette binary images and intersect it with the image silhouettes of the object. As the result of the intersection the node can remain ”black” (if it lies within the object) or be set to ”white” (it lies outside the object) or ”gray” (it lies partly within and partly outside the object), see image on the left in Figure 2c. Note that the meaning of ”black” in the octree and in the binary images is inverted — a node is black if it's projection lies lies entirely in the white area of an image.
5. If the `current node` after Step 4 is not white, it is projected into the Shape from Structured Light binary image representing the nearest laser plane to the node (ideally the plane intersecting the node center) and intersected with the area representing the intersection of the object and the laser plane (image on the right in Figure 2c). Other structured light images, representing planes which do not intersect the `current node`, are irrelevant for determination of its color.
6. If the node is set to Gray it is divided into 8 child nodes of the `current level` $+ 1$, all of which are marked ”black”
7. Processing of the `current node` is finished. If there are more nodes in the `current level` set the `current node` to the next node and go back to Step 4. If all nodes of the `current level` have been processed, increment the `current level` and go to Step 3.
8. The final octree model has been built (Figure 2d).

5 Results

Experiments were performed with both synthetic and real objects. For synthetic objects we built a model of a virtual camera and laser and created input images in such a way that the images fit perfectly into the camera model. Doing so the accuracy of the models constructed can be analyzed, without impact of camera calibration errors. For both synthetic and real objects we compare the volume and the size of the bounding cuboid of the model with the volume and size of the bounding cuboid of the object.

As synthetic objects we created a virtual sphere with the radius $200\ mm$, and a virtual cuboid with dimensions $100 \times 70 \times 60\ mm$. The images of the sphere were constructed in such a way, that both Shape from Silhouette and Shape from Structured Light alone can reconstruct the object completely, whereas for the cuboid a more realistic case was simulated, where the structured light images contain occlusions. The models of these objects were constructed with different parameter values, such as the number of views used and the maximal octree resolution. Figure 3 shows the models built using 360 silhouette and 360 structured light views, with the constant angle of 1° between two views, and the octree resolution 256^3. The tests with the

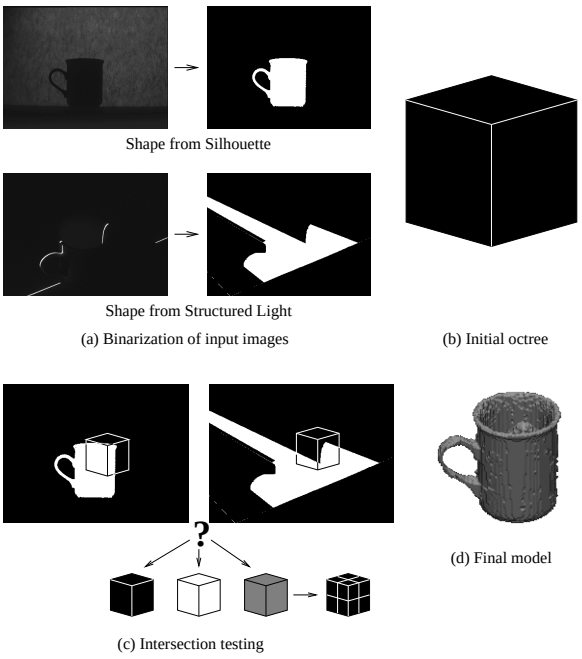


Figure 2: Algorithm overview

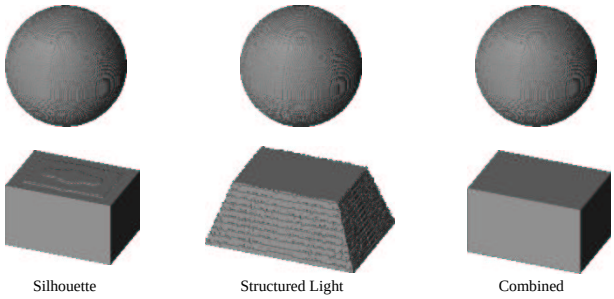


Figure 3: 3D models of synthetic sphere and cuboid

sphere showed that Shape from Silhouette and Shape from Structured Light perform similarly when they have perfect input images — starting from resolution 128^3, both methods were able to create models with the approximation error of 2% or less. Regarding the number of views, 20 views was sufficient for both methods in order to create models with the volume less than 1% different from the models built using 360 views. With the synthetic cuboid, neither of the methods was able to reconstruct the cuboid completely, but the combined method constructed its perfect model starting from the resolution 128^3. However, even using 180 views instead of 360, the volume error of the cuboid was greater than 1% (1.45%), which indicates that flat surfaces are more difficult to model with our method. Table 1 summarizes the results of the models built.

For tests with real objects we used 8 objects: a metal cuboid, a wooden cone, a globe, a coffee cup, two archaeological vessels and two archaeological sherds. The real volume of the

first 3 objects can be computed analytically. For the remaining 5 objects it can be measured by putting the objects in the water, but it has not been done yet at the time of writing this paper, so for these objects we can only compare the bounding cuboid of the model and the object. Figure 4 shows the objects and their models built using 360 views for each of the underlying methods and the octree resolution 256^3. In addition, Figure 5 compares the models of the cup and one of the sherds built using one of the methods only and the combined method, illustrating the necessity of using both methods in order to construct complete models of objects.

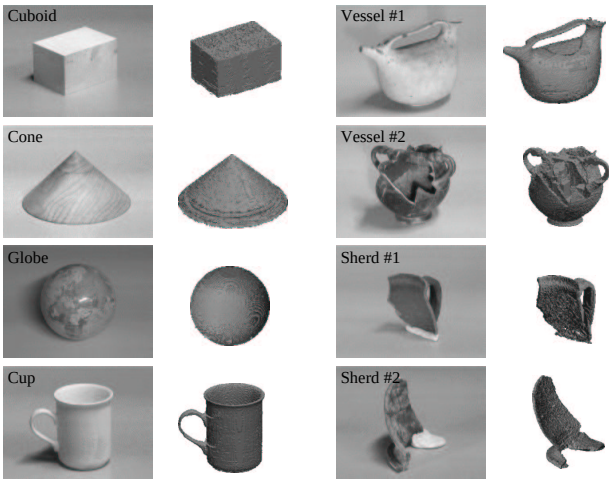


Figure 4: Real objects and their models

Figure 5: Silhouette and structured light based models

The error of the computed volume for real objects was between 3% and 13%, by an order of magnitude larger than the errors with synthetic objects. The main reason turned out to be the threshold based binarization of silhouette images, which interpreted parts of the object as the background, especially close to the turntable surface. That explains why the error was the largest for the cone and the smallest for the globe (see Table 1). The cone has a large base leaning on the turntable, while the globe only touches the turntable in an almost tangential way.

The dimensions and the volume of the objects presented in this section and their 3D models are summarized in Table 1.

object	*octree*	*#views*	*dimensions* (mm)	*volume* (mm^3)	*volume error*
synth. sphere	—	analytic	$400.0 \times 400.0 \times 400.0$	33 510 322	—
	64^3	360+360	$400.0 \times 400.0 \times 400.0$	35 241 984	+5.17%
	128^3	360+360	$400.0 \times 400.0 \times 400.0$	33 786 880	+0.83%
	256^3	360+360	$396.0 \times 396.0 \times 400.0$	33 034 528	-1.42%
	256^3	180+180	$396.0 \times 396.0 \times 400.0$	33 067 552	-1.32%
	256^3	20+20	$400.0 \times 400.0 \times 400.0$	33 230 464	-0.83%
synth. cuboid	—	analytic	$100.0 \times 70.0 \times 60.0$	420 000	—
	64^3	360+360	$100.0 \times 70.0 \times 60.0$	432 000	+2.86%
	128^3	360+360	$100.0 \times 70.0 \times 60.0$	420 000	0.00%
	256^3	360+360	$100.0 \times 70.0 \times 60.0$	420 000	0.00%
	256^3	180+180	$100.0 \times 72.0 \times 60.0$	426 071	+1.45%
	256^3	20+20	$104.0 \times 73.0 \times 60.0$	435 402	+3.67%
real cuboid	—	analytic	$100.0 \times 70.0 \times 60.0$	420 000	—
	256^3	360+360	$101.0 \times 71.0 \times 60.0$	384 678	-8.41%
cone	—	analytic	$156.0 \times 156.0 \times 78.0$	496 950	—
	256^3	360+360	$150.1 \times 149.4 \times 77.5$	435 180	-12.43%
globe	—	analytic	$149.7 \times 149.7 \times 149.7$	1 756 564	—
	256^3	360+360	$149.1 \times 148.2 \times 144.6$	1 717 624	-2.22%
cup	—	analytic	$113.3 \times 80.0 \times 98.9$	N/A	—
	256^3	360+360	$111.6 \times 79.0 \times 98.3$	276 440	N/A
vessel #1	—	analytic	$141.2 \times 84.8 \times 93.7$	N/A	—
	256^3	360+360	$139.2 \times 83.2 \times 91.4$	336 131	N/A
vessel #2	—	analytic	$114.2 \times 114.6 \times 87.4$	N/A	—
	256^3	360+360	$113.0 \times 111.9 \times 86.4$	263 696	N/A
sherd #1	—	analytic	$51.8 \times 67.0 \times 82.2$	N/A	—
	256^3	360+360	$51.0 \times 66.0 \times 79.4$	35 911	N/A
sherd #2	—	analytic	$76.0 \times 107.3 \times 88.5$	N/A	—
	256^3	360+360	$74.9 \times 103.9 \times 86.2$	38 586	N/A

Table 1: Dimensions and volume of objects and their models

6 Conclusion

This paper presented a 3D modeling method based on combination of Shape from Silhouette and laser based Shape from Structured Light, using a turntable to obtain multiple views of an object. The purpose of combining Shape from Silhouette and Shape from Structured Light was to create a method which will use the advantages and overcome the weaknesses of both underlying methods and create complete models of arbitrarily shaped objects.

The experiments with synthetic objects showed that construction of nearly perfect models is possible, limited only by image and model resolution. In the experiments with real objects the results were less accurate, but the algorithm was able to produce complete and visually faithful models for all objects, including sherds and vessels with concave surfaces and a handle. Only the inside of deep objects could not be completely recovered, due to camera and light occlusions.

Overall, our combined modeling approach proved to be useful for automatic creation of models of arbitrarily shaped objects. With respect to its archaeological application it can provide models of any kind of archaeological pottery. The models can also be intersected with arbitrary planes, resulting in profile sections of a sherd or a vessel. Furthermore, the volume of an object can be estimated, including the inside volume of objects such as bowls or cups. However, for high precision measurements of the volume our method did not produce highly accurate results, but it gave a good rough estimate, which is sufficient for most archaeological

applications. Higher accuracy could be achieved by improving the binarization of input images, which showed to be the main reason for relatively large errors for real objects. A possible enhancement to our method would be to take additional color images of an object and perform texture mapping onto the model, which would improve the visual impression of the models built.

References

[1] H. Baker. Three-dimensional modelling. In *Proc. of 5th International Joint Conference on Artificial Intelligence*, pages 649–655, 1977.

[2] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:167–192, March 1988.

[3] P.J. Besl. Active, optical range imaging sensors. *MVA*, 1(2):127–152, 1988.

[4] J. S. De Bonet and P. Viola. Roxels: Responsibility weighted 3D volume reconstruction. In *Proc. of 7th IEEE International Conference on Computer Vision*, pages 418–425, 1999.

[5] N. A. Borghese, G. Ferrigno, G. Baroni, A. Pedotti, S. Ferrari, and R. Savarè. Autoscan: A flexible and portable 3D scanner. *IEEE Computer Graphics and Applications*, 18(3):38–41, 1998.

[6] H. H. Chen and T. S. Huang. A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing*, 43:409–431, 1988.

[7] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 36:100–113, 1983.

[8] J. Davis and X. Chen. A laser range scanner designed for minimum calibration complexity. In *Proc. of 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 91–98, May 2001.

[9] F.W. DePiero and M.M. Trivedi. 3-D computer vision using structured light: Design, calibration, and implementation issues. *Advances in Computers*, 43:243–278, 1996.

[10] R. M. Haralick and L. G. Shapiro. Glossary of computer vision terms. *Pattern Recognition*, 24(1):69–93, 1991.

[11] M. Kampel and R. Sablatnig. Range image registration of rotationally symmetric objects. In N. Brändle, editor, *Proc. of Computer Vision Winter Workshop*, pages 69–77, 1999.

[12] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):197–216, July 2000.

[13] C. Liska and R. Sablatnig. Estimating the next sensor position based on surface characteristics. In *Proc. of 15th. Intl. Conf. on Pattern Recognition, Barcelona, Spain*, pages Vol I: 538–541, 2000.

[14] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):150–158, 1983.

[15] W. Niem. Robust and fast modelling of 3D natural objects from multiple views. In *Image and Video Processing II, Proc. of SPIE*, pages 388–397, 1994.

[16] J. O'Rourke and N. Badler. Decomposition of three-dimensional objects into spheres. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):295–305, 1979.

[17] J. Park, G. N. DeSouza, and A. C. Kak. Dual-beam structured-light scanning for 3-D object modeling. In *Proc. of 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 65–72, May 2001.

[18] M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40:1–29, 1987.

[19] Y. Shirai. *Three-Dimensional Computer Vision*. Springer-Verlag, 1987.

[20] S. K. Srivastava and N. Ahuja. Octree generation from object silhouettes in perspective views. *Computer Vision, Graphics, and Image Processing*, 49:68–84, 1990.

[21] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[22] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.

[23] J. Veenstra and N. Ahuja. Efficient octree generation from silhouettes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 537–542, 1986.

[24] A. Watt. *3D Computer Graphics*. Addison Wesley, 3 edition, December 1999.

[25] R. G. Willson. Tsai camera calibration software. http://www.cs.cmu.edu/˜rgw/TsaiCode.html.

[26] K. Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. In *Proc. of 8th IEEE International Conference on Computer Vision*, pages 217–222, 2001.

Illumination Insensitive Eigenspaces for Mobile Robot Localization *

Matjaž Jogan[1], Horst Wildenauer[2], Aleš Leonardis[1] and Horst Bischof[3]

[1]Faculty of Computer and Information Science, University of Ljubljana

Tržaška 25, 1001 Ljubljana, Slovenia

e-mail: {matjaz.jogan, alesl}@fri.uni-lj.si

[2]Pattern Recognition and Image Processing Group

Institute for Computer Aided Automation, Vienna University of Technology

Favoritenstrasse 9/1832, A-1040 Vienna, Austria

e-mail: wilde@prip.tuwien.ac.at

[3]Institute for Computer Graphics and Vision, Graz University of Technology

Inffeldgasse 16 2. OG, A-8010 Graz, Austria

e-mail: bis@icg.tugraz.ac.at

Abstract

Methods for mobile robot localization that use eigenspaces of panoramic snapshots of the environment are in general sensitive to changes in the illumination of the environment. Therefore, we propose in this paper an approach which achieves a reliable localization under severe illumination conditions by illumination insensitive eigenspaces. The method in question uses gradient filtering of the eigenspaces. The method was tested on images obtained by a mobile robot and, as we show, it outperforms by far the other known methods.

1 Introduction

To enable localization of a mobile robot in an outdoor or indoor environment, a number of methods have been proposed that construct an appearance model of the environment by capturing panoramic views of locations, obtained with an omnidirectional sensor [1,4,10]. The model of appearance is predominantly constructed by compressing the set of visual snapshots captured at

*H. W. was supported by a grant from the Austrian National Fonds zur Förderung der wissenschaftlichen Forschung (P13981INF). M. J. and A. L. acknowledge the support from the Ministry of Education, Science and Sport of Republic of Slovenia (Research Program 506). H. B. was supported by the K plus Competence Center ADVANCED COMPUTER VISION.

different locations using PCA, resulting in the *eigenspace representation*, which has been successfully used in many areas of computer vision [7, 9]. With this approach, images captured during the process of learning get represented as points in a low—dimensional *eigenspace*, which is spanned by the principal components of the data - *eigenimages*. Localization can then be performed by a projection of the momentary panoramic view on the eigenspace, followed by a search for the nearest coefficient of the training images. Panoramic images have the advantage of capturing a wide field of view. Equally oriented images taken at nearby positions are strongly correlated, which allows to build compact models that approximate well the overall appearance of the environment. Furthermore, a dense representation can be obtained even from sparsely acquired images by means of interpolation. The eigenspace method was mainly used in a straightforward way of classifying target appearances by projection without accounting for the possible discrepancies between the learned data and the subsequent images that have to be recognized during the localization phase. This can lead to potential problems when we want the robot to be able to estimate its position in a dynamic environment, with changing configurations of moving objects and persons, and with changing illumination conditions. To cope with occlusions from objects, a robust algorithm for the calculation of the eigenimage coefficients was proposed [5, 4].

While this method can also tolerate some artifacts that appear due to the illumination (e.g. specularities or dark shadows), it results in erroneous localization when dealing with global or smooth illumination changes. Some approaches attempt to alleviate the problem of global illumination by a normalization [6]. In the case of occlusions such an approach can not be applied, since normalization is inherently nonrobust. In panoramic images this problem gets even harder, since they depict 360 degrees of the surrounding, which integrates several local lighting conditions. Such a variety clearly cannot be handled by simple normalization.

In this paper we describe a method for mobile robot localization under varying illumination that achieves illumination invariance of the recognition process by convolving the eigenimages with a bank of linear filters. As a starting point we use the method that was presented and tested on object recognition by Bischof et al. [2], and modify it in order to be applicable for the task of mobile robot localization. As we will demonstrate, we achieve excellent results even in severe illumination conditions.

In section 2 we first briefly review the eigenspace method. Then we show how it is possible to calculate the coefficients of the eigenimage expansion from the filtered eigenimages. This approach is the core of illumination insensitive localization, which is also described in this section. In section 3 we evaluate the method on sets of images obtained by the mobile robot while moving in an indoor environment with changing illumination. We demonstrate, that by applying a filter bank of gradient filters on an eigenspace of panoramic images, we achieve accurate and illumination insensitive localization, compared to the other known approaches. In section 4 we conclude with a discussion.

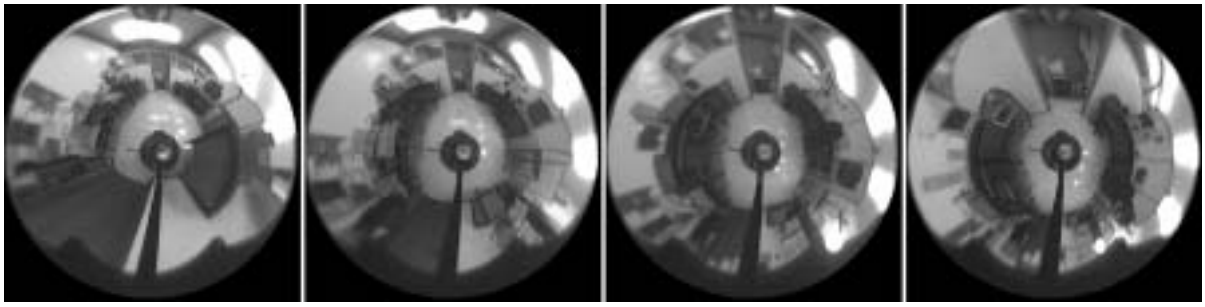


Figure 1: Four images from the training set.

2 Illumination insensitive eigenspace

A straightforward approach to the problem of illumination in appearance-based learning and recognition is to learn the appearance under all of the possible light conditions [6]. However, an object in the environment can produce so many different images that it is not clear how to sample all of them.

It is therefore better to use a recognition method that exploits image features that are invariant to illumination. As we will show, it is possible to greatly alleviate the sensitivity to illumination by convolving the eigenimages with linear filters in order to remove the illumination artifacts.

In this section we first review the standard eigenspace approach and point out the key problems that arise due to its nonrobustness. Then we review the method that achieves illumination insensitive recognition, as it is described in [2].

Eigenspace based recognition To build the eigenspace, we first represent the images from the training set (Fig. 1) as image vectors, from which the mean image is subtracted, $\mathbf{x}_i;\ i = 0 \ldots N-1$, which form an image matrix $X = \left[\begin{array}{cccc} \mathbf{x}_0 & \mathbf{x}_1 & \ldots & \mathbf{x}_{N-1} \end{array} \right],\ X \in \mathbb{R}^{n \times N}$; where n is the number of pixels in the image and N is the number of images. These training images serve as input for the Principal Components Analysis (PCA) algorithm, which results in a set of p *eigenimages* $\mathbf{e}_i,\ i = 1, \ldots, p$, that span a low-dimensional *eigenspace*. Eigenimages are selected on the basis of the variance that they represent in the training set. Every original image $\mathbf{x}_i$ can be transformed and represented with a set of coefficients $q_{ij} = \mathbf{x}_i^T \mathbf{e}_j,\ j = 1, \ldots, p$, which represent a point in the eigenspace. That way, every image is approximated as $\tilde{\mathbf{x}}_i = \sum_{i=1}^{p} q_{ij} \mathbf{e}_i$.

The standard approach to localization is to find the coefficient vector $\mathbf{q}$ of the momentary input image $\mathbf{y}$ by projecting it onto the eigenspace using the dot product $q_i =< \mathbf{y}, \mathbf{e}_i >$, so that $\mathbf{q} = [q_1, \ldots, q_p]^T$ is the point in the eigenspace.

If we want the image $\mathbf{y}$ to be recognized as its most similar counterpart in the training set (or in a representation constructed by means of interpolation, see [7]), the corresponding coefficients have to lie close together in the eigenspace. However, in the case when the input

image is distorted, either due to occlusion, noise or variation in lighting, the coefficient we get by projecting onto the eigenspace can be arbitrarily erroneous.

It was however shown, that one can also calculate the coefficient vector $\mathbf{q}$ by solving a system of k linear equations on $k \geq p$ points $\mathbf{r} = (r_1, \ldots r_k)$

$$y_{r_i} = \sum_{j=1}^{n} q_j e_{jr_i} \;\; 1 \leq i \leq k \tag{1}$$

using a robust equation solver and multiple hypotheses [5].

In [4] it was shown how this method can be used to allow robust localization in presence of occlusions. However it does not solve the problem of illumination.

Illumination insensitivity The method presented in [2], takes the computations of parameters one step further. Since Eq. (1) is linear, it also holds that

$$(f * x)(r) = \sum_{i=1}^{p} q_j (f * e_i)(r) \;\; , \tag{2}$$

where f denotes a filter kernel.

This means that if we convolve both sides with a kernel, the equality still holds. Therefore, we can calculate the coefficients q_i also from the filtered eigenimages, if we filter the input image.

By using a set of linear filters $\mathcal{F}$ we can construct a system of equations

$$(f_s * x)(r) = \sum_{i=1}^{p} q_i (f_s * e_i)(r) \;\; s = 1, \ldots, h \;\; . \tag{3}$$

It is now possible to calculate the coefficients $\mathbf{q}$ either by using k points, or using h filter responses at that single point, or a combination of this two.

It is well known from the literature that gradient-based filters are insensitive to illumination variations. By taking a filter bank of gradient filters in several orientations, we can therefore augment the descriptive power of the representation and achieve illumination invariance in the recognition phase.

Illumination invariant localization of a mobile robot can therefore be performed as follows: once the eigenspace is built, we filter the eigenimages by a bank of filters (Fig. 2). Then, for localization, the momentary input image $\mathbf{y}$ from the panoramic camera has to be filtered with the same filters; only after that we retrieve its coefficient vector $\mathbf{q}$ using the robust equation solver. The calculated coefficients are used to infer the momentary location of the robot.

Note that this approach is not the same as constructing the eigenspace from filtered images, as it allows far more flexibility. In fact, once the model is built, we can select the filters and their number according to the momentary illumination conditions, or even run the algorithm on non–filtered images. Moreover, the eigenspace built from filtered images is a suboptimal representation due to the small correlation ratio between the transformed images [11].

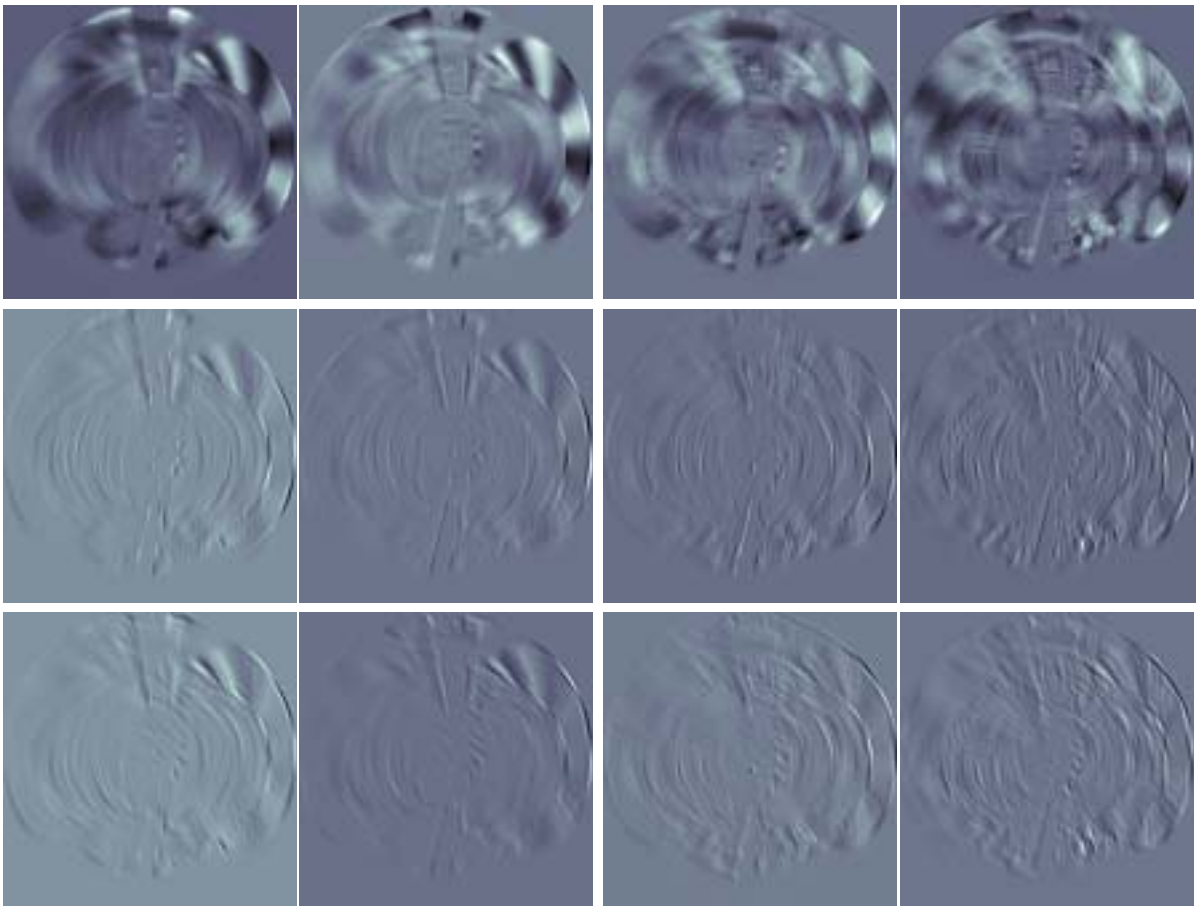


Figure 2: First row shows the first four eigenimages. The consecutive rows show two filter responses for each eigenimage.

3 Robot localization — Experimental Results

We have extensively evaluated the proposed method on sequences of images acquired by a mobile robot using a panoramic camera setup with a hyperbolic mirror. The set of training images was generated by moving the robot on a straight 4 meter long path through the laboratory. Pictures were taken at positions 10cm apart from each other, yielding a total number of 40 images. All images were acquired under constant lighting conditions. These training images were represented in an eigenspace of dimension 10, which serves as our appearance model of the environment. Seven sets of test images each consisting of 40 images were produced by moving the robot on the same path as in the training case. This time the illumination conditions were changed for every set. In addition, in 4 sets, occlusions in the images were caused by people walking through the room. In Figure 3 we depict examples of the training images and test images.

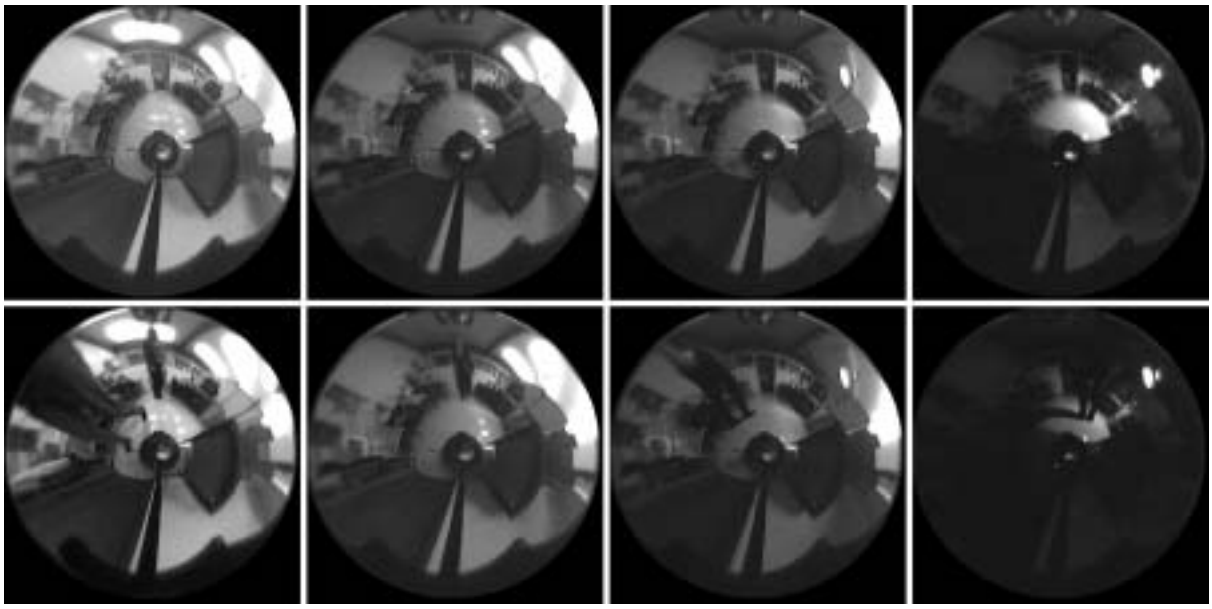


Figure 3: Upper row, from left to right: Training set images and test sets 1 to 3 with changing illumination. Lower row: Test sets 4 to 7 with occlusions.

Choosing the filters To achieve insensitivity against the changes in the illumination we convolve the eigenimages and test images with a bank of linear gradient-based filters. In particular, we have chosen a set of six steerable filters [8]. It is important to note that it is always possible to change the set of linear filters, even when the model is completely built. Fig. 2 depicts the first four eigenimages filtered with the two of the six derivative filters.

Evaluation of the retrieved coefficients To have a systematic performance evaluation we compared the performance of the filtered eigenspace with two other approaches: the standard approach, in which images are projected on the non-filtered eigenspace, and the normalized standard PCA approach, in which the eigenspace is built from training images normalized to unit vector length in order to account for global illumination changes. In the latter approach, the test images also have to be normalized before the projection.

To compensate for the changes in global illumination which represent the additive factor of illumination variability, we applied a distance measure based on the angle between coefficient vectors for the search of the nearest coefficients in the eigenspace. In other words, once we have determined the eigenspace coefficients of an image, we find its corresponding point in the eigenspace by choosing the coefficient vector of the training image with the smallest angle.

In Fig. 4 we compare the angular coefficient error for the test sets with severe illumination changes (set 2 and 3) and the corresponding occluded test sets. The results for all 7 test sets are shown in Table 1. These experiments demonstrate that our approach consistently performs better for all illumination conditions, for the test set without occlusions and for the test sets with occluded images.

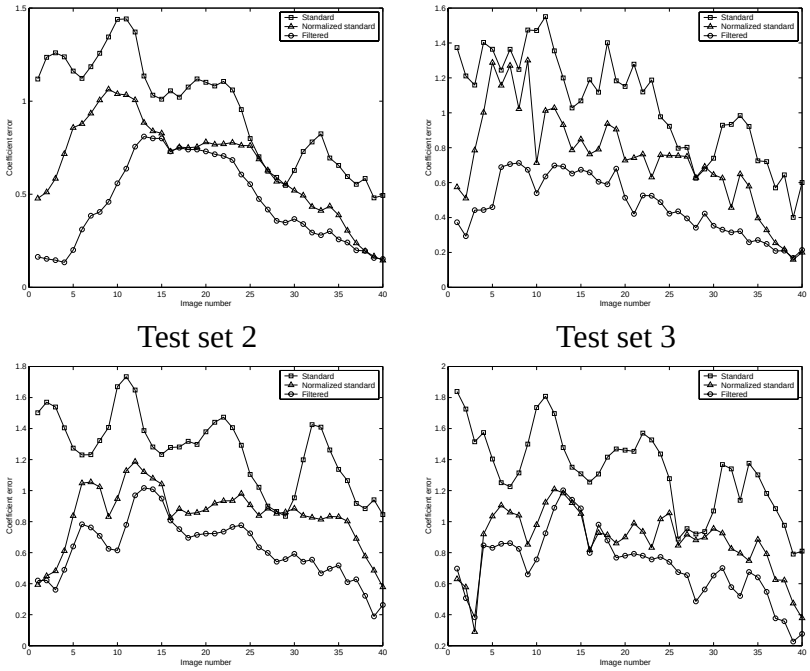


Figure 4: Comparison of the coefficient errors for severe illumination conditions.

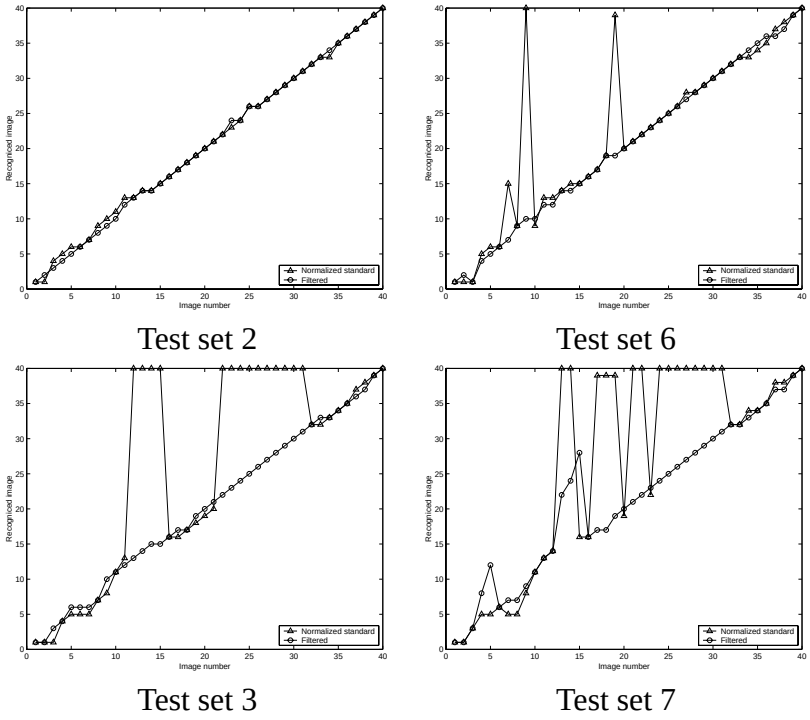


Figure 5: Place recognition for non-occluded and occluded images under identical illumination.

Table 1: Average angular coefficient error.

Set	1	2	3	4	5	6	7
Standard	0.62	0.96	1.26	0.38	0.77	1.05	1.33
Normalized	0.37	0.66	0.84	0.35	0.50	0.73	0.89
Filtered	0.19	0.45	0.63	0.26	0.33	0.47	0.72

Table 2: Average localization error (cm).

Set	1	2	3	4	5	6	7
Standard	7	48.7	74.8	2.5	13.5	57.8	108.0
Normalized	1.5	3.3	65.0	0.8	3.3	19.0	68.3
Filtered	0	1.3	4.0	0.5	1.8	2.3	14.0

Evaluation of the location recognition The graphs in Fig. 5 illustrate the performance of localization of the mobile robot. Every deviation from the diagonal means an error in the recognition of the momentary position. We omitted the results of the standard approach, since its performance was consistently worse than the performance of the depicted methods. For a full comparison of all tested methods on all test sets, see Table 2. It is evident that our method clearly outperforms both the standard and the standard approach with normalization.

Learning and recognition using a coarse representation In view based localization using the eigenspace approach the coefficients of intermediate (untrained) views are usually interpolated from the coefficients representing the trained views, resulting in a parametric manifold representation. To test our illumination invariant localization method on an interpolated representation, we built the eigenspace using every fourth image of the training set. The resulting 11 coefficient vectors were then interpolated to a 10cm grid. We performed tests on localization for the seven sets of forty images in different configurations of illumination and occlusion using a five dimensional eigenspace. Again, our approach was compared with the standard approach with normalization.

In Fig. 6 the angular coefficient error for the test sets with severe illumination changes (set 2 and 3) and the corresponding occluded test sets (set 6 and 7) is depicted. The results for all 7 test sets can be found in Table 3. The graphs in Fig. 7 illustrate the localization performance of the mobile robot. The results of the localization evaluation for all test sets are shown in Table 4. These experiments demonstrate that our approach performs better than the standard approach with normalization even when a parametric manifold representation interpolated from a coarse training set is used.

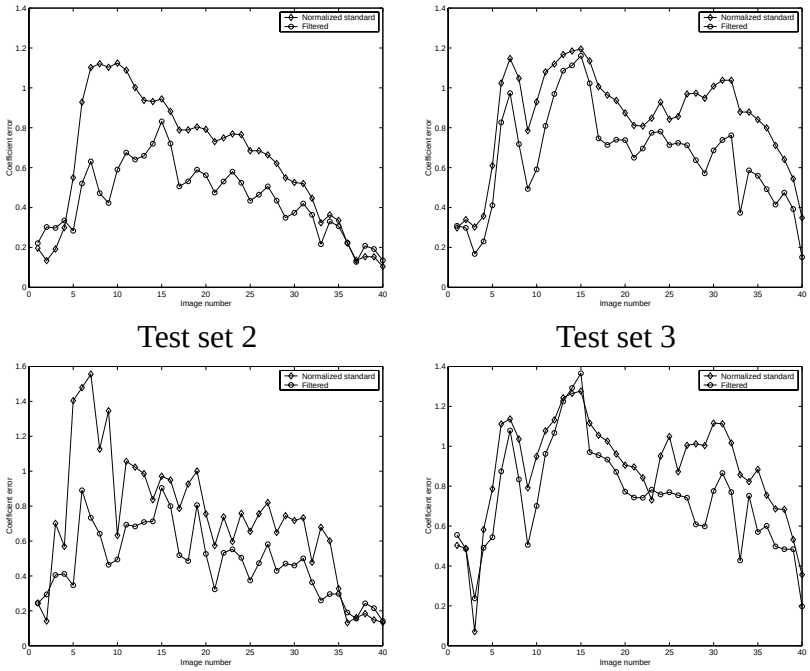


Figure 6: Comparison of coefficient errors for the parametric manifold.

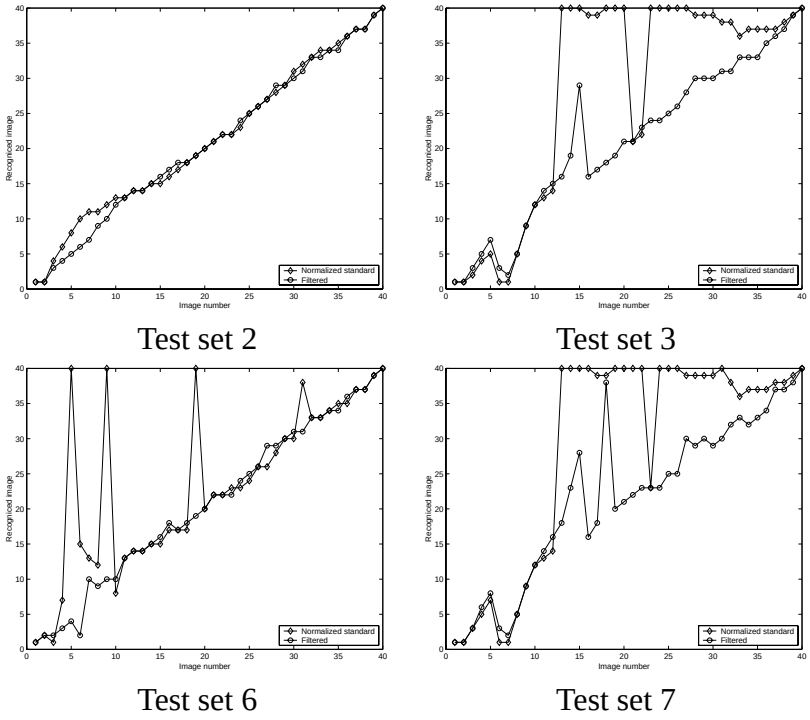


Figure 7: Comparison of view recognition rates for the parametric manifold.

Table 3: Average angular coefficient error - Interpolated Views.

Set	1	2	3	4	5	6	7
Normalized	0.31	0.64	0.86	0.34	0.44	0.73	0.89
Filtered	0.24	0.44	0.65	0.24	0.28	0.48	0.74

Table 4: Average localization error (cm) - Interpolated Views.

Set	1	2	3	4	5	6	7
Normalized	4.8	9.3	83.8	6	6.8	34	89.8
Filtered	3.3	4.8	14.8	2.8	2.8	7.8	23.8

4 Conclusion

In this contribution we described an eigenspace–based method for mobile robot localization under varying illumination, which performs illumination invariant recognition based on filtering of the eigenimages. As our experiments show, the method outperforms the other known methods. Furthermore, it does not require a special model for the purpose, since the invariance achieved is inherent in the recognition method itself. This allows for extreme flexibility; note that we can easily combine the method with the robust approach to the retrieval of parameters that copes with occlusion [2]. Further, since the model is constructed from original images, the correlation properties of panoramic images are being preserved. We can therefore build and use an efficient interpolated representation.

References

[1] H. Aihara, N. Iwasa, N. Yokoya, and H. Takemura. Memory-based self-localisation using omnidirectional images. In Anil K. Jain, Svetha Venkatesh, and Brian C. Lovell, editors, *14th International Conference on Pattern Recognition*, pages 297–299. IEEE Computer Society Press, August 1998.

[2] Horst Bischof, Horst Wildenauer, and Aleš Leonardis. Illumination insensitive eigenspaces. In *Proc. Intl. Conf. Computer Vision ICCV01*, pages I: 233–238. IEEE Computer Society, 2001.

[3] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[4] Matjaž Jogan and Aleš Leonardis. Robust localization using panoramic view-based recognition. In *15th International Conference on Pattern Recognition*, volume 4, pages 136–139. IEEE Computer Society, September 2000.

[5] Aleš Leonardis and Horst Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding - Special Issue on Robust Statistical Techniques in Image Understanding*, 78(1):99–118, 2000.

[6] Hiroshi Murase and Shree K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (12):1219–1227, 1994.

[7] S. K. Nayar, S. A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Trans. on Robotics and Automation*, 12(5):750–758, October 1996.

[8] E. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. In *IEEE Trans. on Image Processing*, pages 1–15, 1996.

[9] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 586–591, 1991.

[10] Niall Winters, José Gaspar, Gerard Lacey, and José Santos-Victor. Omni-directional vision for robot navigation. In *IEEE Workshop on Omnidirectional Vision*, pages 21–28. IEEE Computer Society, 2000.

[11] A. Yilmaz and M. Gokmen. Eigenhill vs. eigenface and eigenedge. *Pattern Recognition*, 34(1):181–184, January 2001.

Retrieving and Using Topological Characteristics from 3D Discrete Images

Pascal Desbarats, Jean-Philippe Domenger

Laboratoire Bordelais de Recherche en Informatique - UMR 5800

Université Bordeaux 1, 351, cours de la Libération, 33405 Talence, France

Fax Number: +33 556 846 669 Phone Number: +33 556 846 091

e-mail: {Desbarats|Domenger}@labri.u-bordeaux.fr

Abstract

In order to characterize surfaces and 3D objects, we usually use topological invariants. In this paper, we show how to retrieve topological informations such as the Euler characteristic or the Betti numbers by using a representation of the segmented image by 3-maps and inter-voxel boundaries. A 3-map is a three dimensional topological map that encodes the topology of a subdivision of an 3D orientable quasi-manifold without boundary. We then use these topological informations to direct a minimization algorithm for the 3-map.

1 Introduction

In the field of image analysis, geometrical informations such as the volume of a region, the curvature of a surface, etc. are widely used. In some case, these informations are insufficient. As stressed by C.N. Lee and A. Rosenfeld [13], topological informations are also very important. These informations being invariant through continuous deformations, they can be used for example to recognize a 3D objet that is flexible and can change of shape (such as the heart in medical imagery). They can also be used in analysis: to characterize a pathology (like a number of holes through the surface between the ventricles of the heart different from the usual) or to determine the number of cavities in seismic images for example.

We have already introduced a model for the representation of 3D discrete images [6, 4] which is an extension to the 3D case of a 2D model developped by J.P. Braquelaire, L. Brun and J.P. Domenger [8, 3, 7]. This 3D model associates a geometrical level based on inter-voxel boundaries and a topological model based on 3-maps. A similar approach has been proposed by Bertrand et al [2] which proposes to build 3-maps from several levels on border maps [1] by using the precode method introduced by Fiorio [10].

Here, we show how to use our model of representation to retrieve topological informations from a 3D segmented discrete image. We recall first some definitions that we use in this paper, then we briefly explain the construction of our model. Finally we use the Euler characteristic

to simplify the 3-map and we show how to compute the Betti numbers for the surfaces of 3D objects.

2 Definitions

We consider in the following 3D image which are parallelepipedic sets of voxels, each voxel being defined by a triplet of integer coordinates and a value. The set of coordinates of image voxels is called the *domain* of the image. The values of voxels are gray levels. The result of the segmentation is a labelled image that can also be seen as a grey level image by associating a gray level to each label. The support of the image is the discrete space $\mathbb{Z}^3$. A region (or volume) of the 3D image is a maximal isovalued 6-connected set of voxels of an image. If we represent a voxel by a unit cube, the boundary of a region can be defined as the set of cube faces shared by a pair of voxels which one belongs to the region and the other one does not. Such boundaries have been for instance described by Rosenfeld [15] and by Françon [11].

This representation leads to a cellular decomposition of the discrete space into cells of dimension 0, 1, 2 and 3. A *3-cell* of coordinates (x_p, y_p, z_p) is the unit cube centered at the integer point p, a *2-cell* is the intersection of two *3-cells* V_{P_1} and V_{P_2} such that p_1 and p_2 are 6-neighbors, a *1-cell* is the intersection of two *3-cells* V_{p_1} and V_{p_2} such that p_1 and p_2 are 18-neighbors but not 6-neighbors, and a *0-cell* is the intersection of two *3-cells* V_{P_1} and V_{P_2} such that p_1 and p_2 are 26-neighbors but not 18-neighbors (see Fig. 1). The discrete space is then decomposed into *0-cells*, open *1-cells*, open *2-cells*, and open *3-cells*.

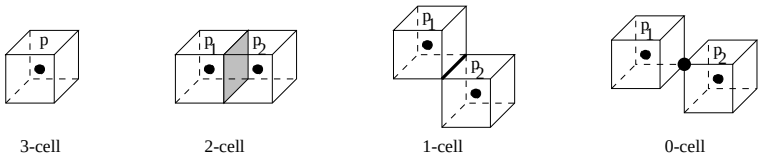


Figure 1: Elements of the cellular decomposition of the 3D discrete space.

According to this terminology a voxel is thus a valued *3-cell*, and the boundary of the image is the set of 2-cells separating two voxels with different values (i.e. belonging to two different regions) or separating a voxel belonging to the domain of the image and a voxel not belonging to the domain of the image. A *2-cell* belonging to the boundary of an object is called a *s-cell*. The set of *s-cells* is called the *s-boundary*. Finally the *s-boundary* is cut into paths homeomorphic to a disc and called *s-patches* [6, 4].

The result of the cutting is a set of *1-cells* paths drawn on the *s-boundary* and corresponding either to the borders of part of surface shared by two different objects (inter-objects cutting) or to the cutting of these parts into topological discs (intra-object cutting). The *1-cells* defining the cutting are called the *l-cells*. Finally the set of *l-cells* induces a graph which the vertices are some *0-cells* called *p-cells* and the edges are the sequences of *l-cells* joining two *p-cells* and called *l-chains*. The *p-cells*, *s-cells*, and *l-cells* are encoded by using an array of same size than the 3D image with seven binary flags by entry (three for encoding the *s-cells*, three for the *l-cells* and one for the *p-cells*) [6]. Up to seven additional flags may be needed to perform

markings involved in geometrical updating. This data structure is called the *boundary image*. Any traversal of the geometrical structure is done by using this boundary image.

The topological representation of 3D segmented images lays on the association of 3-maps with the elements of the geometrical boundary of the segmented image. Each *s-patch* is associated with a face of 3-map and is described by a sequence of edges corresponding to the *l-chains*. The faces of the map are sewed along the edges in order to define the volumes.

3 Representation of the segmented image

Let us first recall the following definitions: a 3-map is a tuple $(\mathbf{D}, \alpha, \sigma, \gamma)$ where $\mathbf{D}$ is a set of elements called *darts* and α, σ and γ permutations defined on $\mathbf{D}$ such that α and σ are involutions without fixed point and γ is such that $\alpha\gamma$ is an involution without fixed point. The permutation α links two opposite darts and each pair of darts $(d, \alpha(d))$ corresponds to an edge of a face border. For each vertex the permutation σ links the two half-edges belonging to a same face. Finally the permutation γ links darts of adjacent faces along their common edges (see for instance [7]).

The correspondence between the geometry and the topology encodings lays on the decomposition of the *s-boundary* into elements of dimension 0 (the *p-cells*), 1 (the *l-chains*), and 2 (the *s-patches*). It allows to define the geometrical equivalent of a dart of the 3-map. We denote by $S_p^{(\overrightarrow{u}, \overrightarrow{v})}$ the *s-cell* containing the *p-cellp* and being parallel to the half-plane $\overrightarrow{u}, \overrightarrow{v}$ (with $\{\overrightarrow{u}, \overrightarrow{v}\} \subset \{-\overrightarrow{x}, -\overrightarrow{y}, -\overrightarrow{z}, \overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}\}$). Let $L_p^{\overrightarrow{d}}$ be the *l-cell* included in $S_p^{(\overrightarrow{u}, \overrightarrow{v})}$ with $\overrightarrow{d} \in \{\overrightarrow{u}, \overrightarrow{v}\}$). Then the geometrical equivalent of a dart is the oriented *s-cell* defined by the tuple $\prec p, \overrightarrow{d}, s \succ$. Such a tuple is called a *head*.

The construction of the 3-map associated with a cut *s-boundary* (i.e. a boundary image) is done by following all the *l-chains* generated by the cutting algorithm. By following *s-cells* along *l-chains* it is possible to find any pair of associated heads. For each head h_i a dart d_i is created, such that if h_i and h_j are the two heads associated by a *l-chain* then $\alpha(d_i) = d_j$ (the involution α is encoded by $\alpha(d) = -d$ and thus we have $d_j = -d_i$). For a dart d the permutation γ is updated by turning counterclockwise around the *l-cell* of the head associated with d . Finally, the construction of σ is done by associating for each node the darts associated with the heads belonging to the same *s-cell*.

The elements of the topological decomposition are implicitly encoded by the 3-map and can be explicited by extracting some 2-maps from the 3-map (a 2-map is a 2D graph defined by a pair of permutation which one is an involution without fix point). The relevant 2-maps are the following ones: $(\mathbf{D}, \sigma, \gamma)$ encodes the vertices of the decomposition, $(\mathbf{D}, \alpha, \gamma)$ encodes its edges, $(\mathbf{D}, \alpha, \sigma)$ encodes its faces, and $(\mathbf{D}, \gamma^{-1}\alpha, \gamma^{-1}\sigma)$ encodes its volumes. Each volume is associated with a label v by a labelling function Λ defined on the set of darts. In other terms $\Lambda(d) = \Lambda(d')$ iff d and d' belong to the same connected component of the volume map $(\mathbf{D}, \gamma^{-1}\alpha, \gamma^{-1}\sigma)$

If there are cavities in the image the 3-map has several connected components. For this reason an inclusion relation is used to associate the infinite volume of each 3-map with the

finite volume in which it is geometrically included (an infinite volume is the external volume of a cavity). The inclusion relation between a finite volume and its infinite volumes is described by two functions. If v_o is the label of a finite volume and v_i is the label of an infinite volume included in v_o then the volume v_o is called the *parent* of v_i and v_i is a called a *child* of v_o.

4 Minimization of the graph using the Euler characteristic

The intra-object cutting, i.e. the cutting which decomposes the *s-boundary* into topological discs in order to be identified with the faces of the 3-map, is not minimal. The Figure 2a represents a torus with its cutting, which is extracted on the Figure 2b. The Figure 2c represents the minimal graph that can be associated with a torus.

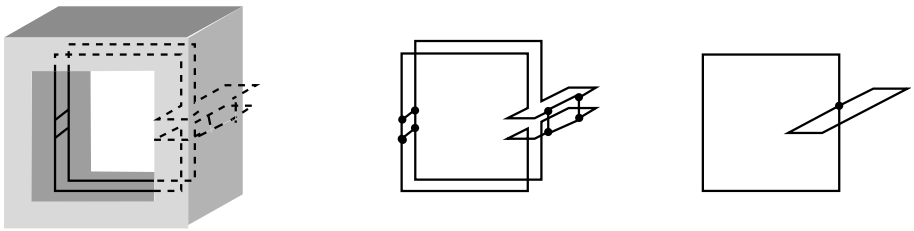


Figure 2: Cutting and minimal graph of a torus

One can see that the cutting defines a graph with 4 faces, 12 edges and 8 vertices, whereas the minimal graph contains only 1 face, 2 edges and 1 vertex. The number of faces, edges and vertex can easily be determined by considering the 3-map. We will denote them #F, #E and #V respectively.

Let us recall a theorem taken from [14]:

Theorem 1 *Let the surface S be given as a plane model and let #F, #E and #V denote the number of faces, edges and vertices in the plane model. Then the sum #V − #E + #F is a constant independent of the manner in which S is divided up to form the plane model. This constant is called the* Euler characteristic *of the surface and is denoted* $\chi(S)$.

We can verify this theorem for the exemple of the torus above.
For the cutting, we have $\chi(S) = \#V - \#E + \#F = 8 - 12 + 4 = 0$ and for the minimal graph $\chi(S) = \#V - \#E + \#F = 1 - 2 + 1 = 0$.

The minimization of the graph will then be done by suppressing faces, edges and vertices while verifying that the Euler characteristic remains constant.

We can now minimize the graph of the Figure 2b. On the Figure 3a, the edge e_0 can be suppressed because it will merge two faces, i.e. $\#F = \#F - 1$ and $\#E = \#E - 1$. The Euler characteristic is preserved.
The same argument is utilised on Figure 3b. The vertices v_1 and v_2 can be suppressed because

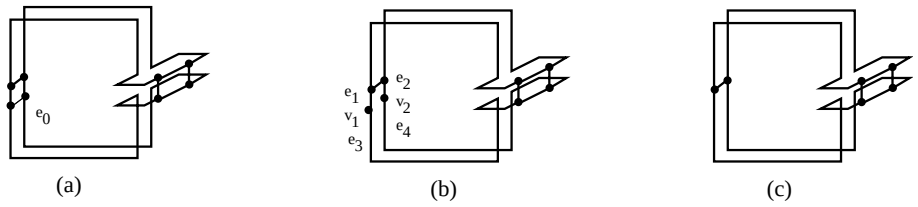


Figure 3: First set of simplification

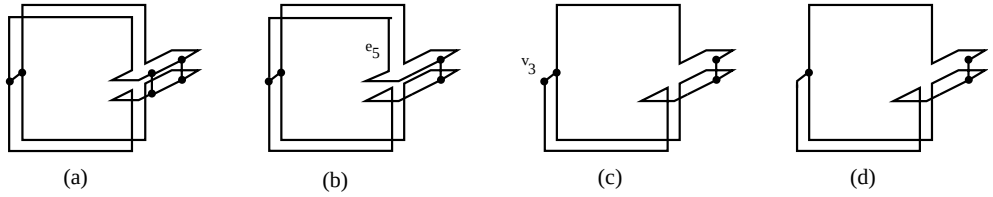


Figure 4: Second set of simplification

it will merge respectivily e_1 and e_3, and e_2 and e_4 ($\#E = \#E - 2$ and $\#V = \#V - 2$). The result is displayed on Figure 3c.

A second set of simplification can then be done.
By applying the same method as above the graph of the Figure 4a is simplified to the graph of the Figure 4a. The edge e_5 can be suppressed because it merges two faces and finally the vertice v_3 can be suppressed because it merges two edges. The result is displayed on the Figure 4c. The processus of simplification ends on that graph because no element can be suppressed without modifying the Euler characteristic of the surface. This graph will be called *minimized graph*. Remark that this graph is different from the minimal graph of the Figure 2c.

All the operations of merge and suppression of elements are done directly on the 3-maps accordingly to the coherence of the model (see [5]).

5 Retrieving the Betti numbers

5.1 Betti numbers for the surface of a 3D region

We have shown in the previous section how to calculate the Euler characteristic for the *s-boundary* of a 3D region. The Euler-Poincaré formula tells us that the Euler characteristic can be expressed as :

$$\chi(S) = \beta_0 - \beta_1 + \beta_2 \tag{1}$$

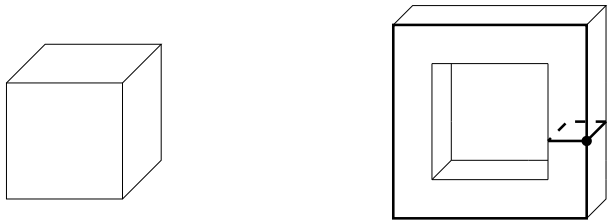
where β_0, β_1 and β_2 are called the Betti numbers for the surface.

These numbers can be calculated through group-theoric techniques [12], but we will rather be interested in their topological significance.
The zeroth Betti number (β_0) is the number of connected components of the surface. The first

Betti number (β_1), also called 1-connectivity (or connectivity number), can be defined as the maximum number of simultaneous cuts that can be made on a surface without disconnecting it. Finally, the second Betti number (β_2) characterizes the orientability of the surface (i.e. it equals 0 if the surface is non-orientable and 1 if it is orientable).

Remark that we only have closed surfaces in our model. In this case, the gender of the surface is half its first betti number.

Let us illustrate these definitions with some examples. For the two examples of Figure 5,



(a) (b)

Figure 5: Betti numbers on surfaces: cube and torus

we have $\beta_0 = 0$ (there is only one connected component) and $\beta_2 = 1$ since the surfaces are orientable. The surface of the cube displayed on Figure 5a cannot be cut without being disconnected whereas the surface of the torus of the Figure 5b can be cut twice. Therefore $\beta_1 = 0$ for the cube and $\beta_1 = 2$ for the torus. The gender of the surface of the cube is 0 and 1 for the surface of the torus.

We only deal in our model with orientable surfaces, i.e. we will always have $\beta_2 = 1$ for our surfaces. We will now explain how to calculate β_1 for a single 3D region.

In this case there is only one connected component of surface, i.e. $\beta_0 = 1$. We have explained in the previous section that our decomposition does not represent a minimal cutting of the surface into topological discs. Therefore we cannot have directly β_1.
But we can easily calculate the Euler characteristic for the surface and then obtain the first Betti number from the Euler-Poincarré formula (equation 1) :

$$\beta_1 = \beta_0 + \beta_2 - \chi(S) \tag{2}$$

The intra-object cutting of the surface of a cube and a torus (simplified using the method described in the previous section) is displayed on the figure 6.

For the cube, we have 2 faces, 1 edge and 1 node. Thus, we have $\chi(S) = 2 - 1 + 1 = 2$. Therefore, using the equation 2, we have $\beta_1 = \beta_0 + \beta_2 - \chi(S)$ with $\beta_0 = 1$ and $\beta_2 = 1$. Then $\beta_1 = 1 + 1 - 2 = 0$. This verifies the result obtained higher and that the gender of the surface of a cube is $\frac{0}{2} = 0$.

For the torus displayed on Figure 6b, the decomposition is composed of 2 faces, 4 edges and 2 nodes. The Euler characteristic for this surface is then $\chi(S) = 2 - 4 + 2 = 0$. Therefore $\beta_1 = \beta_0 + \beta_2 - \chi(S) = 1 + 1 - 0 = 2$. We verify that the surface of a torus can be cut twice without disconnecting it and that the gender of this surface is 1.

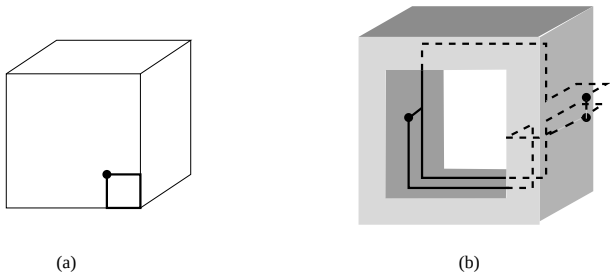


Figure 6: Betti numbers on surfaces : Cuttings of cube and torus s-bound

5.2 Betti numbers for aggregates

The model of 3-maps that we use allows us to describe the topology of aggregates, i.e. of face-adjacent 6-connected 3D regions. An exemple of aggregate is given on the Figure 7.
In this section, we will compute the Betti numbers for the outer surface of an aggregate and for each objet of this aggregate.

Each region of the aggregate has an orientable surface, therefore the second number of Betti for each region and for the aggregate is equal to 1.
We will have as many connected componant of surface as regions composing the aggregate, therefore the zeroth Betti number for the aggregate will be equal to the number of regions in this aggregate (and equals 1 for each single region).
In [9], G. Damiand indicates that in the case of such aggregates, the gender of the aggregate is the sum of the genders of the regions. The gender being half the first Betti number, the first Betti number for the aggregate will be the sum of the first Betti numbers of the objects.

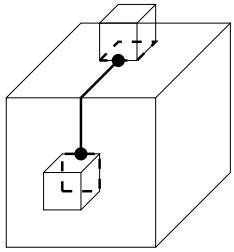


Figure 7: An aggregate composed of three cubes

The Figure 7 represents an aggregate composed of three cubes and displays the minimized graph of the cutting. On this graph we have for the surface of the aggregate $\#F = 5$, $\#E = 3$ et $\#V = 2$. The Euler characteristic for this surface is then $\chi(S) = 2 - 3 + 5 = 4$. We have also directly $\beta_0 = 3$ and $\beta_2 = 1$. Therefore, we can compute β_1 using the equation (2):
$\beta_1 = \beta_0 + \beta_2 - \chi(S) = 3 + 1 - 4 = 0$. We verify that the outer surface of this aggregate is homeomorphic to a sphere.
Let us now consider the central cube. We have for its surface $\#F = 3$, $\#E = 3$ et $\#V = 2$, therefore $\chi(S) = 2-3+3 = 2$. We also have $\beta_0 = 1$ and $\beta_2 = 1$. Therefore $\beta_1 = 1+1-2 = 0$.

If we consider finally the other two cubes, we have for each one $\#F = 2$, $\#E = 1$ et $\#V = 1$, therefore $\chi(S) = 1 - 1 + 2 = 2$. Furthermore $\beta_0 = 1$ and $\beta_2 = 1$, then $\beta_1 = 1 + 1 - 2 = 0$. This verifies that the first Betti number for the aggregate is the sum of the first Betti numbers for each region.

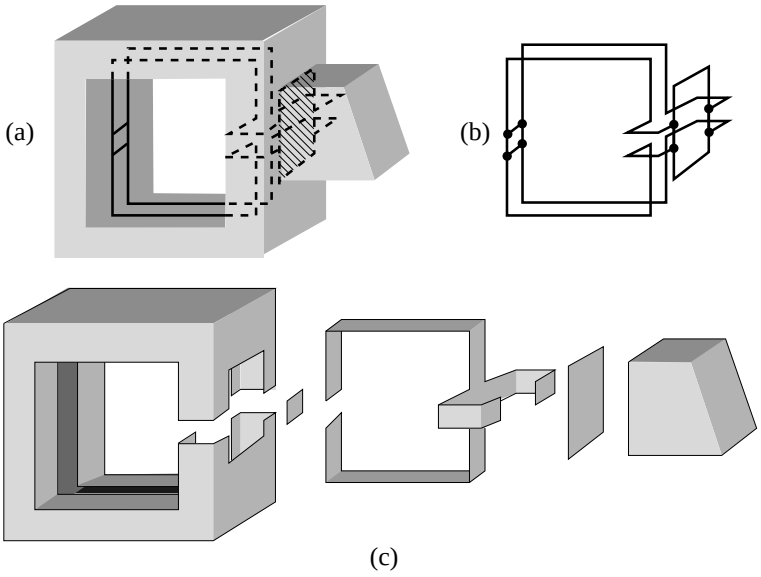


Figure 8: An aggregate composed of a torus and a cube

Let us finally consider the example of the Figure 8. We have on the graph for the aggregate $\#F = 5$ (see Figure 8c), $\#E = 12$ and $\#V = 8$. Then, $\chi(S) = 8 - 12 + 5 = 1$.
As $\beta_0 = 2$ (two objets in the aggregate) and $\beta_2 = 1$ (the volume of the aggregate is orientable), $\beta_1 = 2 + 1 - 1 = 2$. The surface of the aggregate is homeomorphic to a torus.
For the torus, $\#F = 4$, $\#E = 12$, $\#V = 8$ and $\beta_0 = 1$, $\beta_2 = 1$. Therefore $\beta_1 = 1 + 1 - 0 = 2$.
For the cube, $\#F = 2$, $\#E = 4$, $\#V = 4$ and $\beta_0 = 1$, $\beta_2 = 1$. Therefore $\beta_1 = 1 + 1 - 2 = 0$.
Here again, the first Betti number for the aggregate is the sum of the first Betti numbers for each region.

6 Conclusion

This work proves the interest of having a representation model with a geometrical level and a topological one. The direct decomposition of the surface by the cutting algorithm allows us to build the 3-maps and to have directly the number of faces, of edges and of vertices of the underlying graph. Thus, computing the Euler characteristic of the surface of a region or of an aggregate is very easy and fast.

We have also shown how to compute the Betti numbers for the surfaces. These numbers are essential to the analysis of 3D objets as we mentionned in the introduction.

Our approach is based on the decomposition of the surfaces of the 3D regions but the next step may be to compute the Betti numbers for the volumes. For 3D regions [16], the Euler characteristic is defined by

$$\chi(V) = \#V - \#E + \#F - \#R$$

where R is the number of 3D regions.
The Euler-Poincaré formula is then:

$$\chi(V) = \beta_0 - \beta_1 + \beta_2 - \beta_3$$

with β_0 being the number of 3D connected components, β_1 being the maximum number of surfaces that can traverse the volume without disconnecting it, β_2 being the number of cavities of the 3D region and β_3 its orientability. Note that we directly have β_0 and β_3 and that β_2 is computable from the inclusion list that we have defined section 3.

References

[1] Y. Bertrand and Ch. Fiorio anf Y. Pennaneach. Border map : a topological representation for nd image analysis. In G. Bertrand, M. Crouprie, and L. Perroton, editors, *Proceedings of DGCI'1999, Lecture Notes in Computer Science*, volume 1568, pages 242–257, 1999.

[2] Y. Bertrand, G.Damiand, and Ch. Fiorio. Topological encoding of 3d segmented images. In G. Borgefors, I. Nyström, and G. Sanniti di Baja, editors, *Proceedings of DGCI'2000, Lecture Notes in Computer Science*, volume 1953, pages 311–324. SV, 2000. To appear in Lecture Notes in Computer Science.

[3] J.P. Braquelaire and L. Brun. Image segmentation with topological maps and inter-pixel representation. *Journal on Visual Communication and Image Representation*, 9(1):62–79, 1998.

[4] JP. Braquelaire, P. Desbarats, and J.P. Domenger. Representation of 3d segmented image with intervoxel boundaries and combinatorial maps. Technical report, University of Bordeaux 1 – Labri, 2000. Research report – submitted.

[5] J.P. Braquelaire, P. Desbarats, and J.P. Domenger. 3d split and merge with 3d-maps. In *Proc. of the 3rd IAPR-TC-15 Workshop on Graph-based Representation*, pages 32–43. CUEN, 2001. ISBN 887146579-2.

[6] J.P. Braquelaire, P. Desbarats, J.P. Domenger, and C.A. Wüthrich. A topological structuring for aggregates of 3D discrete objects. In *Proc. of the 2nd IAPR-TC-15 Workshop on Graph-based Representation*, pages 193–202. Österreichische Computer Gesellschaft, 1999. ISBN 3-85804-126-2.

[7] J.P. Braquelaire and J.P. Domenger. Representation of segmented image with discrete geometric maps. *Image and Vision Computing*, 17:715–735, 1999.

[8] L. Brun. *Segmentation d'images couleur à base topologique*. PhD thesis, Université Bordeaux 1, december 1996.

[9] G. Damiand. *Définition et étude d'un modèle topologique minimal de reprsentation d'images 2D et 3D*. PhD thesis, Université Montpellier II, December 2001.

[10] C. Fiorio. *Approche inter-pixel en analyse d'image: une topologie et des algorithmes de segmentation*. PhD thesis, Université Montpellier 2, 1995.

[11] J. Françon. Discrete combinatorial surfaces. *Graphical Models and Image Processing*, 57(1):20–26, 1995.

[12] L. Christine Kinsey. *Topology of Surfaces*. Springer Verlag, 1993.

[13] C.N. Lee and A. Rosenfeld. Simple connectivity is not locally computable for connected 3d images. *Computer Vision, Graphics, and Image Processing*, 51:87–95, 1990.

[14] Martti Mntyl. *An Introduction to Solid Modeling*. Computer Science Press, 1988.

[15] A. Rosenfeld, T.Yung Kong, and A.Y. Wu. Digital surfaces. *Graphical Models and Image Processing*, 53(4):305–312, 1991.

[16] Jarek Rossignac. *Tutorial: Representations, Design and Visualization of Solids and of Geometric Structures*. EG93 TN6 Eurographics, 1993. ISSN 1017-4656.

Stable Matching Based on Disparity Components

Jana Kostková and Radim Šára
Center for Machine Perception, Faculty of Electrical Engineering
Czech Technical University, Technicka 2, Prague, Czech Republic
tel: +420 2 2435 7637, fax: +420 2 2435 7385
{kostkova,sara}@cmp.felk.cvut.cz

Abstract

In this paper, we demonstrate how disparity maps could be improved by applying adaptive windows (varying in their shape). In our approach, windows adapt to high-correlation structures in disparity space, which we call *disparity components*. The method is straightforward and it is applicable to any stereo algorithm. This helps us in solving standard stereo problems such as: (1) to enforce the inter-line continuity constraint, (2) to improve disparity map accuracy.

To show the proposed method properties, we measure the improvement of stereo matching failure due to adaptive windows in a rigorous ground-truth evaluation experiment. We demonstrate that adaptive windows account for $15\times$ improvement in false positive rate and $5\times$ improvement in mismatch rate. All other measured types of error improve only marginally. Our results on standard stereo pairs are also presented.

1 Introduction

The stereo matching problem has been studied from the early sixties of the last century. However, there are still many open problems remaining. In this paper, we demonstrate that selection of correct windows for computing the similarity values is crucial for the accuracy of disparity maps. We propose applying adaptive windows. The windows adapt to high-correlation structures in disparity space. They are traced out as connected disparity components of match candidates with similar disparity. This approach allows us to solve the following problems:

- P1: to better enforce the inter-line continuity constraint, and thus improve the mismatch error rate.
- P2: to obtain accurate disparity maps: to detect small objects and occluding boundaries correctly, to identify half-occluded regions, and to capture fine variations in disparity.

We demonstrate that these problems could be solved by a straightforward method resulting in a significant improvement of disparity map. Also, due to the generality of our method, it is applicable to all area-based standard stereo matching approaches.

In [1], a method based on disparity components has also been proposed. Their algorithm finds correspondences based on the size of sets of points with the same disparity called disparity components.

Kanade and Okutomi [4] have already used the term 'adaptive window'. They used rectangular windows, which adapted only by increasing their size. This process was performed iteratively in all four directions, until uncertainty (computed over the window) converged.

In [2], the final pixel disparities are computed by tracing the highest correlation matches. Firstly, the local maximum correlation pairs are selected as the seeds. Then, the disparities for each pixel are found by tracing high correlation pairs neighbouring with the seeds.

In our approach, the windows are generated by components in disparity space. The algorithm allows small variations of disparity within one component (to capture the scene shape better) and the match selection is based on recomputed similarity values, not on disparity component size alone. This allows us to match correctly problem-causing objects, which are, for example: small, curved, or very slanted ones. Due to the generality of the algorithm, it can be applied to an arbitrary stereo algorithm, while the improvement of the resulting disparity map should be significant.

We demonstrate the improvement of the disparity map on a comparison of two simple methods. Both are based on the same stable matching algorithm. One includes the adaptive window approach, while the other does not. The results are shown on a ground-truth experiment, as well as on real scenes.

The next section describes the disparity component matching. The experimental results are shown in Sec. 3. In Sec. 4 we discuss properties of the proposed method and outline future work. Conclusions are given in Sec. 5.

2 Disparity Components

The disparity components algorithm consists of four steps:

1. preliminary selection of matching candidates (called aggregation in [9]),
2. tracing disparity components,
3. re-weighting the correlation values,
4. computing final disparity map.

In the first step, a pre-selection algorithm selects *match candidates*. The pre-selection approach can be based on global energy minimizations [3, 6, 5] or local correlation methods [4, 10]. The only condition is that multi-value results are expected. In our approach, we have applied the correlation method with a full disparity range matching table. The correlations can be evaluated on small rectangular fixed-size image windows in order to improve their discriminability.

In the second step, the disparity components are created from match candidates resulting from the first step. Continuity within a disparity component is defined in disparity space by

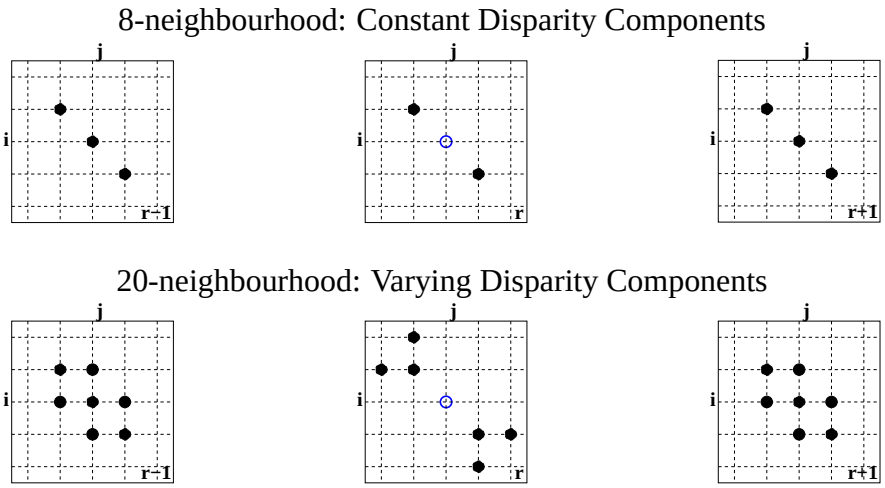


Figure 1: Neighbourhoods of point on row r on position (i, j) in matching table (empty circle): left column is matching table for row $r - 1$, center for row r and right for row $r + 1$.

a neighbourhood relation. If we allow just points of the same disparity in one disparity component, the components would be 8-connected structures. The 8-neighbourhood is shown in Fig. 1(a) - (c). Each input image row has its own matching table, the matching table for the row r is shown in the middle sub-figure. In the matching table, the columns from the left image are labeled by i, from the right image by j. The definition of 8-connected components corresponds to Boykov's approach [1]. On the one hand, it is simple, but on the other hand, it is limited to scenes with objects of constant disparity (which is a strong restriction).

In our approach, small variations in disparity within a component are allowed: the points in the same component have 'similar disparities', i.e. the difference of neighbouring pixel disparities must be smaller or equal to one. This definition results in 20-neighbourhood, which is shown in Fig. 1(d) - (f). For each pre-selected match candidate, the unique disparity component is identified. The component defines unambiguously two non-rectangular windows for this match in both the left and right images.

In the third step, these windows are used to compute *a new correlation value* for each pre-selected match. These windows make it possible to deal with varying disparity. This is where we substantially differ from the work of others.

Each time only a small fixed-size subset of the component is taken into account in order to obtain correlations comparable by statistical procedures. If two correlation values are to be compared, we need to make sure their statistical properties are similar. This is the reason why sample window size must be equal for all pixels. High correlation computed over a large window and high correlation computed over a small window are incomparable because they are likely to have very different confidence intervals. On the one hand, the enlarging the window increases correlation discriminability, on the other hand, from our experience, it follows that there is no need for large correlation windows in area-based matching. The match correlation is re-computed only if the corresponding disparity component is large enough, otherwise it is removed (to suppress the mismatches caused by noise or low texture areas).

In the fourth step, the final disparity map is computed. Any stereo matching algorithm can be used. The method is applied to the set of re-weighted match candidates and it results in a single-valued disparity map.

3 Experimental Results

In this section we demonstrate the improvement of disparity map as a result of the application of adaptive windows. We compare results from a simple stereo algorithm and results given by the same stereo algorithm with applying the disparity components approach. Firstly, we describe this algorithm. Then, we show differences on graphs where various matching errors are monitored. This is how the behaviour of algorithms can be studied in greater detail. Finally, we compare results on real scene images.

3.1 FX-Stable Matching

As the reference stereo matching algorithm, we have chosen FX-stable matching [10]. Stable matching has been selected for its key features: completeness (it explains all points in disparity space either matched or occluded), low false positive rate and mismatch rate [10], and the ability to compute multi-value disparity maps.

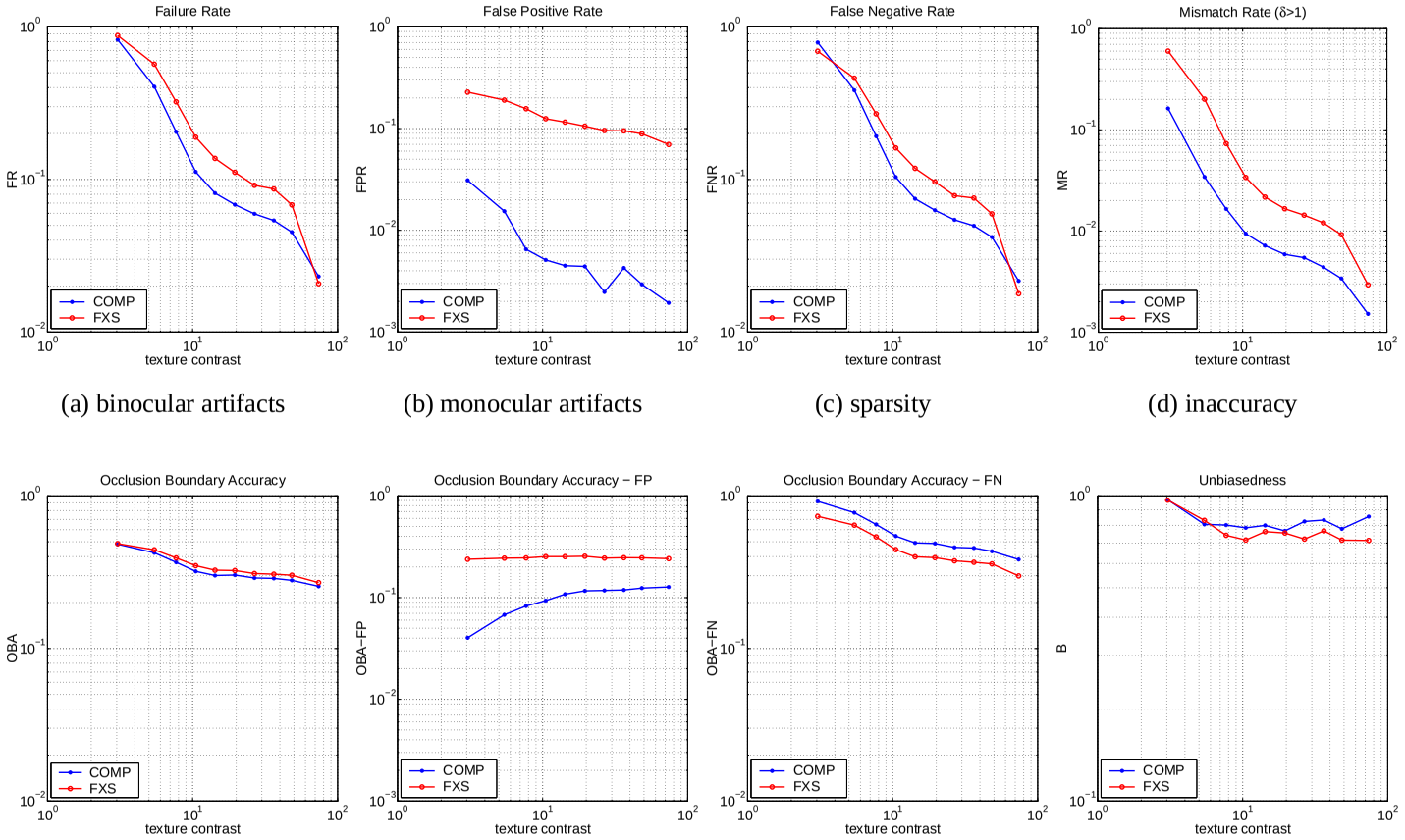
The matching candidates, the first step, are found by the stable matching algorithm [10] with finite inhibition zone depth. We used FX-zone and the inhibition zone depth was set to 10 pixels.

3.2 Quantitative Comparison Based on Ground Truth

The methodology is based on ground-truth (shown in Fig. 3 left) and focuses on specific failure mechanisms (discussed below). This approach enables us to study various algorithms in detail in order to discover their specific weaknesses. The algorithms are tested under varying signal-to-noise ratio.

Types of Error In our experiment we distinguished the following eight types of errors: *Failure Rate (FR)* measures the overall quality. *False Positive Rate (FPR)* measures the half-occluded artifacts. *Occlusion Boundary Accuracy (OBA)* measures the occlusion boundary detection quality. *Mismatch Rate (MR)* measures the accuracy of matching. *False Negatives Rate (FNR)* measures the disparity map sparsity. *Occlusion Boundary False Positive Rate (OBFPR)* measures the shift of a misdetected occlusion boundary into the half-occluded region. *Occlusion Boundary False Negative Rate (OBFNR)* measures the shift of a misdetected occlusion boundary into the object. *Unbiasedness (B)* measures bias towards large objects. Precise definitions are given in [10].

Evaluation Results of both algorithms are shown in plots in Fig. 2. Texture contrast (on the horizontal axis in all plots) is directly related to signal-to-noise ratio. Respective error rates are shown on the vertical axes. Note that both axes have logarithmic scale.



(a) binocular artifacts (b) monocular artifacts (c) sparsity (d) inaccuracy

(e) occlusion artifacts (f) occlusion FP (g) occlusion FN (h) unbiasedness

Figure 2: Different types of matching error evaluated on ground-truth test data for both algorithms.

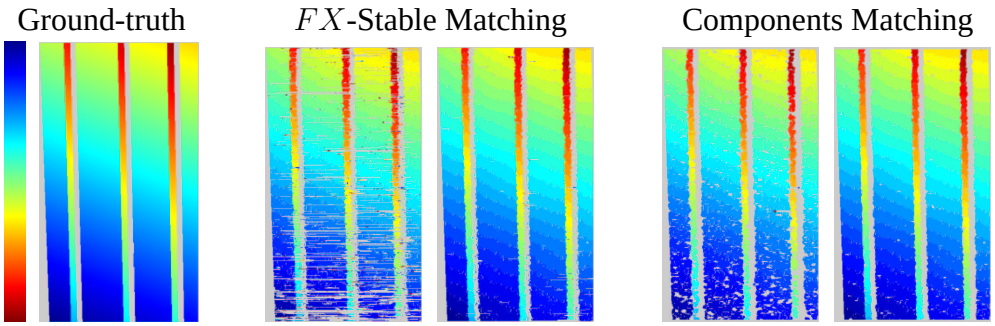


Figure 3: Disparity maps from the tested algorithms. The bar on the left shows disparity map color coding: low disparities are blue, high disparities are red. The left disparity maps in each column give results for the intermediate image contrast of 10.5 and the right ones for the maximum contrast of 74.0.

The improvement of results can be seen in all graphs. The dramatic improvement (about $15\times$) is in false positive rate (Fig. 2(b)), which means, the components algorithm is able to detect half-occluded regions more correctly. The second significant improvement (about $5\times$) is in mismatch rate (Fig. 2(d)).

The overall quality of the occlusion boundary detection (Fig. 2(e)) is comparable for both methods. However, they differ upon detailed examination: false positives (Fig. 2(f)) have decreased by the components approach (also at boundaries the detection of half-occluded regions is improved), but false negatives (Fig. 2(g)) have increased. The false negatives are an artifact of our implementation of the third step. Correlations are re-computed using centered windows, i.e. the re-computed match is in the centre of the window (note that windows are non-rectangular). Therefore, the re-computing subsets of corresponding components cannot have the required size for matches at objects boundaries. Thus, such matches are suppressed.

Disparity maps computed by both algorithms are shown in Fig. 3 in order to study their improvement due to the application of the components approach in detail. For each method, the left disparity maps give results for the intermediate image contrast of 10.5 and the right ones for the maximum contrast of 74.0.

3.3 Qualitative Comparison on Real Scenes

We also show results on real scenes images to obtain a (visual) comparison with other stereo approaches, see Fig 4. The images were selected to describe the algorithms behaviour and its properties. Last but not least they can demonstrate how the defined criteria have been met.

In all disparity maps, the significant improvement in mismatch rate can be observed. This is a result of incorporating the inter-line continuity constraint into the standard approach (the first problem P1). The half-occluded regions contains (almost) no matches, consequently, their identification together with occluding boundaries detection has improved as well (the second problem P2).

The Lab scene has been created at the University of Tsukuba [8] and has become one of the images most often used in the stereo research community. Results are shown in Figs. 4(a), 4(b). The considerable suppression of mismatches can be seen under the table (right bottom part) and

in the right top part of the image. Both these regions are of low-texture in the original image, which is why no correspondences should be found there. On the lamp, the mismatches caused by specularities have also been eliminated. In the right top part, there is a repetitive structure in the original image. The mismatch rate has been improved, but a few mismatches have remained. The detection of the thin segments at the lamp's handle is comparable in both approaches. This is due to the application of centered windows for re-computing correlations in the components algorithm, as discussed above. Applying the non-centered windows could improve the results.

The Pentagon scene also belongs to well-known stereo image pairs. As the image consists of repetitive structures, it is difficult to an obtain accurate disparity map. Our results are shown in Figs. 4(c), 4(d). The mismatches inside the Pentagon (visible in both left and right parts as dark blue coloured points) have been suppressed. On the right part, there are none of them left, but on the left part, a few of them remained (due to shadows in the original image). Trees, which are outside of the Pentagon, are detected better. The mismatches in the car park, in the left bottom part of the image, have also been eliminated.

The last real scene is the Birch scene (Figs. 4(e), 4(f)). In this image the ordering constraint is not fulfilled, which enables us to demonstrate the ability of components matching algorithm to cope with such data. Even though the disparity map is sparser, it has few errors and includes all important objects and structures. The thin trees have been detected correctly. The false positives almost disappeared, which can be seen in particular on the trees.

4 Discussion

The components matching algorithm, we have introduced in this paper, is based on centered (non-rectangular) adaptive windows. The centering is responsible for suppression of matches near occlusion boundaries. This results in widening the half-occluded regions. In our future work non-centered windows are to be applied to cover the components at boundaries.

The pre-selection of match candidates is crucial for the overall success of our approach. In our future research we will focus on the selection of a method which optimizes this step. The promising possibility seems to be applying the Graph Cuts approach [5].

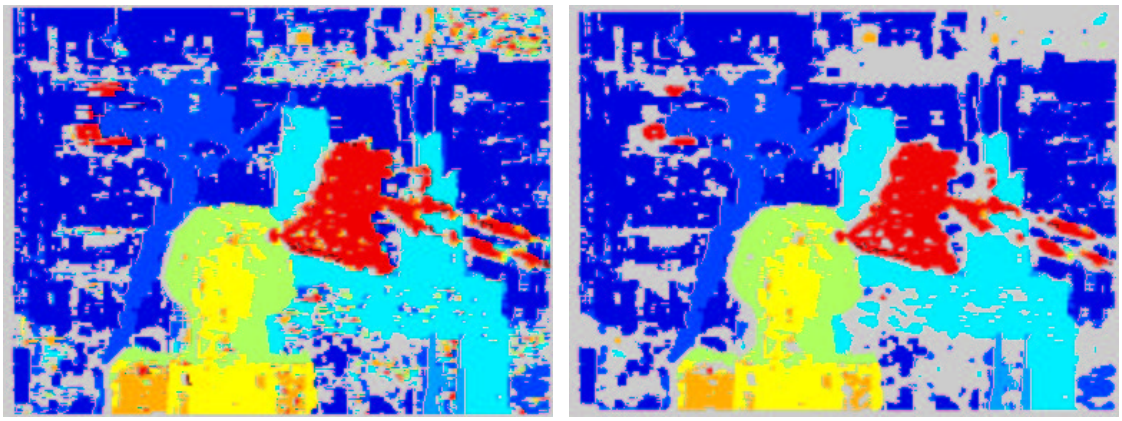
In the last step of our method, a matching able to detect small objects is required. The best results (in this sense) are produced by the FX-dominant matching [10] but the disparity map is very sparse. This is why the FX-stable matching was selected. In the future, we plan to use Confidently Stable matching algorithm [7] instead of FX-stable approach because its error rate is extremely low, while the density of the disparity map remains high.

5 Conclusions

In this paper, we have shown the improved results of binocular matching failure rates by applying adaptive windows. The designed method is very straightforward. Our work differs from others in: (1) the way windows are adapted to high-correlation structures in disparity space, (2) the way disparity components are used to re-compute match correlation, (3) the rigorous evaluation of results compared to an approach without adaptive windows.

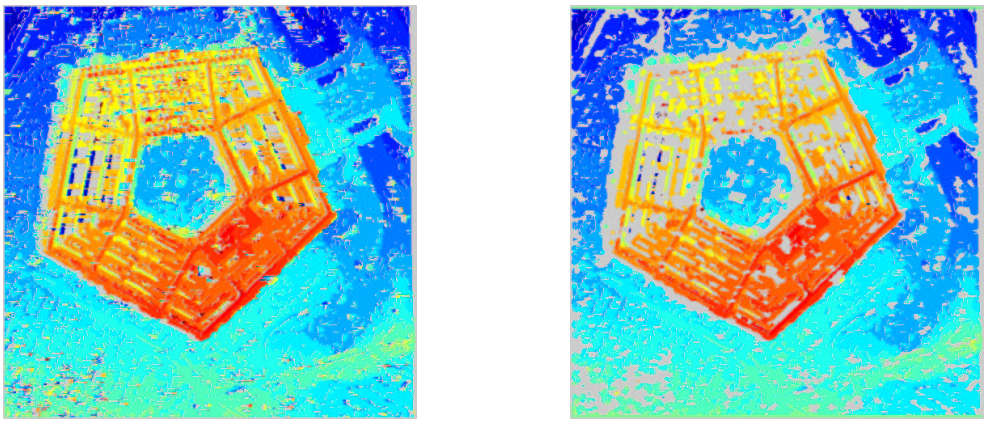
In a quantitative evaluation with ground truth, we have demonstrated the improvement of disparity maps due to adaptive windows. The highest (about $15\times$) improvement was in the false

Lab scene



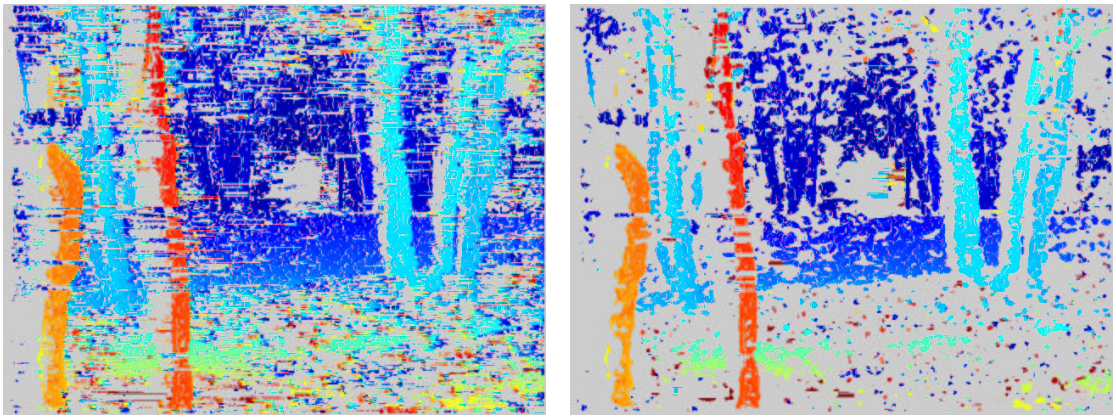
(a) (b)

Pentagon



(c) (d)

Birch



(e) (f)

Figure 4: Resulting disparity maps for different real scenes. Left column is performed by FX-stable matching algorithm, right column by Components matching. Colour coding has been described in Fig. 3.

positive rate, which showed that the identification of half-occluded regions and detection of occlusion boundaries have been greatly improved. The second largest (about $5\times$) improvement was in the mismatch rate, which demonstrated that including the inter-line continuity constraint significantly decreased the mismatches. All other error types have been improved as well, but only marginally. We have also performed qualitative experiments on real scenes, which confirmed the results from the quantitative evaluation.

Acknowledgements

This research was supported by the Grant Agency of the Czech Republic under project GACR 102/01/1371 and by the Czech Ministry of Education under project MSM 212300013.

The Lab scene images are the courtesy of University of Tsukuba, and the Pentagon images and the Birch images are the courtesy of Carnegie Mellon University.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Disparity component matching for visual correspondence. In *Proc Conf on Computer Vision and Pattern Recognition*, pages 470–475, 1997.

[2] Q. Chen and G. Medioni. A volumetric stereo matching method: Application to image-based modeling. In *Proc Conf on Computer Vision and Pattern Recognition*, pages 29–34, 1999.

[3] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proc European Conf on Computer Vision*, 1998.

[4] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on PAMI*, 16(9):920–932, September 1994.

[5] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc Int Conf on Computer Vision*, July 2001.

[6] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc Int Conf on Computer Vision*, 1998.

[7] R. Šára. Finding the largest unambiguous component of stereo matching. In *Proc European Conf on Computer Vision*, Copenhagen, Denmark, May/June 2002. To appear.

[8] K. Satoh and Y. Ohta. Occlusion detectable stereo using a camera matrix. In *Proc Asian Conf on Computer Vision*, pages 331–335, 1995.

[9] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc of Workshop on Stereo and Multi-Baseline Vision*, pages 131–140, Kauai, Hawaii, 2001.

[10] R. Šára. Sigma-delta stable matching for computational stereopsis. Research Report CTU–CMP–2001–25, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, September 2001.

Matching Hierarchies of Segmentations *

R. Glantz[+], M. Pelillo[+], and W. G. Kropatsch[#]

[+] Dipartimento di Informatica

Università Ca' Foscari di Venezia

Via Torino 155, 30172 Mestre (VE), Italy

e-mail:{glantz,pelillo}@dsi.unive.it

[#] Pattern Recognition and Image Processing Group 183/2

Institute for Computer Aided Automation, Vienna University of Technology

Favoritenstr. 9, A-1040 Vienna, Austria

e-mail:krw@prip.tuwien.ac.at

Abstract

We propose to match two hierarchies of segmentations by many-to-many mappings between the regions of the two hierarchies. The mappings preserve the order of the regions (w.r.t. set inclusion) in both hierarchies. The matching involves weights for the significance of individual regions within a hierarchy and similarity measures for the comparison of regions from different hierarchies. Irregular pyramids, in which each level consists of an attributed plane graph and an attributed dual graph are well suited to represent the hierarchies and to provide the information for computing the weights and the similarity measures.

keywords: many-to-many matching, segmentation, pyramid, graphs.

*This work is supported by the Austrian Science Foundation (FWF) under grant P14445-MAT and by MURST under grant MM09308497.

1 Introduction

Hierarchies of segmentations can be obtained from an image by a sequence of criteria for merging neighboring regions. When criterion n cannot be applied anymore, the n-th segmentation is attained. Consider two hierarchies $\mathcal{H}(I_1)$ and $\mathcal{H}(I_2)$ of segmentations with respect to the images I_1 and I_2, respectively. We assume that the hierarchies have been constructed according to the same sequence of criteria. In this paper the *structural* similarity of I_1 and I_2 is grasped by a hierarchy-preserving many-to-many mapping between the regions of $\mathcal{H}(I_1)$ and $\mathcal{H}(I_2)$.

If we assume that the highest level of $\mathcal{H}(I_j)$ ($j = 1, 2$) contains only one region, i.e. the whole image, the partial order of the regions in each hierarchy may be described by a rooted tree, the vertices of which represent the regions (the root represents the whole image), and the edges of which represent set inclusion. Thus, we may focus on many-to-many mappings between two rooted trees that preserve the orders imposed by the rooted trees.

We use a tree matching algorithm that is based on a maximum clique formulation in a derived association graph [7]. Alterations of the region properties are taken into account by a similarity measure between regions and structural alterations are balanced by means of weights that indicate the relevance of the regions for the hierarchy.

The paper is organized as follows: In Sec. 2 we present a graph-based concept for calculating and representing nested morphological segmentation. The tree matching algorithm is explained in Sec. 3. Sec. 4 is devoted to the weights and the similarity measures for matching nested morphological segmentations. Experimental results are presented in Sec. 5.

2 Nested Morphological Segmentation

Morphological segmentation methods rely on the intuitive idea of flooding a topographic surface in order to find the watersheds and to determine the catchment basins [5]. The idea of flooding is also used to derive hierarchies of catchment basins [6]. We will first sketch how to derive the watersheds and the catchment basins by dual graph contraction [4]. Then we will construct the hierarchy of the catchment basins.

Let the topographic surface be defined by the modulus of the gradient image as in [6]. We represent the topographic surface by a dual pair $(\overline{G_0}, G_0)$ of graphs, G_0 being plane. The vertices and edges of G_0 represent the pixels and the 4-neighborhood of the pixels, respectively. For each vertex v of G_0 let $alt(v)$ denote the altitude (modulus of the gradient) at v. The vertices and the edges of the graphs $G_0 = (V_0, E_0)$ and $\overline{G_0} = (\overline{V_0}, \overline{E_0})$ are equipped with attribute values $att(\cdot)$ as follows [3] (Figure 1):

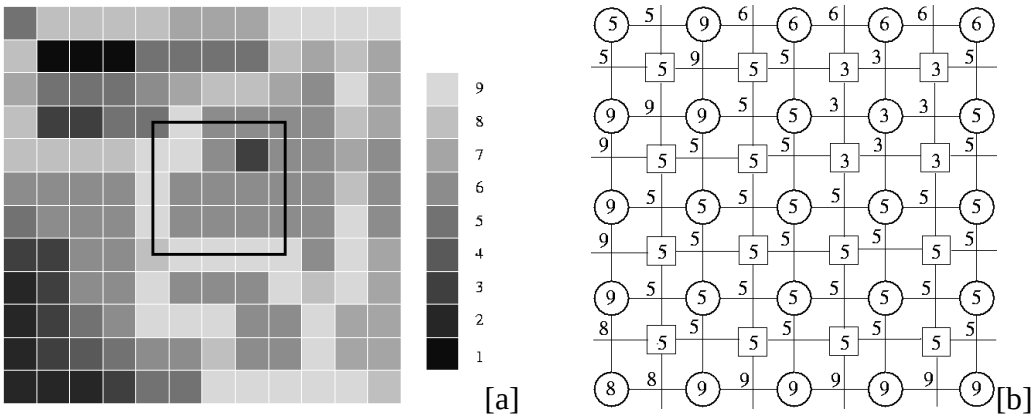


Figure 1: (a) Representation of a topographic surface by means of pixels whose gray values indicate the altitude. (b) Initial pair $(\overline{G_0}, G_0)$ of attributed dual graphs restricted to the square in (a). The circular and square vertices belong to G_0 and $\overline{G_0}$, respectively.

- $att(v) := alt(v) \quad \forall v \in V_0$,
- $att(e) := min\{att(v) \mid v \text{ is end vertex of } e\} \quad \forall e \in E_0$,
- $att(\overline{e}) := att(e)$ for all pairs of dual edges $(e, \overline{e}) \in E_0 \times \overline{E_0}$,
- $att(\overline{v_0}) := min\{att(\overline{e}) \mid \overline{e} \text{ has } \overline{v_0} \text{ as an end vertex}\} \quad \forall \overline{v_0} \in \overline{V_0}$.

A sequence of monotonic dual graph contractions [3] transforms the dual pair $(\overline{G_0}, G_0)$ into the dual pair $(\overline{G_{2n}}, G_{2n})$. The dual pair $(\overline{G_{2n}}, G_{2n})$ obtained from Figure 1a is depicted in Figure 2a. The vertices and the edges of $\overline{G_{2n}}$ represent the catchment basins and the neighborhood relations of the catchment basins, respectively. In accordance with [3] the contraction of G is done in a way which ensures that G_{2n} may be embedded on G_0.

Coarser segmentations are derived from the catchment basins by unifying the basins. The unification of neighboring basins b_1 and b_2 is achieved by contracting the edge in $\overline{G_{2n}}$ that connects the vertices represented by b_1 and b_2. As pointed out in [6], a variety of criteria can be used for the choice of the basins to be unified first. The criteria are usually formulated by means of the basin sizes, their depths or the minimal altitude on the common border of the basins. In [3] it is proven that the altitude of the deepest point in basin b is given by the attribute of the vertex representing b in $\overline{G_{2n}}$. It is also shown that the attribute of each edge $\overline{e}$ with end vertices representing b_1 and b_2 indicates the minimal altitude along that part of the border line between b_1 and b_2 which is represented by e (e and $\overline{e}$ being a dual pair of edges).

Contracting the edges of $\overline{G_{2n}}$ according to increasing values of $att(e_{2n})$ (e_{2n} edge of $\overline{G_{2n}}$) yields a hierarchy of regions as the one depicted in Figure 2b, where unification of R_i and R_j is denoted by $R_i + R_j$.

The graph $\overline{G_{2n}}$ is contracted in subsequent parallel steps, until there exists but one vertex. The hierarchy of the regions obtained forms a so called *irregular pyramid* [4]

$$(\overline{G_{2n}}, G_{2n}), (\overline{G_{2n+1}}, G_{2n+1}), \ldots, (\overline{G_{2n+2m}}, G_{2n+2m}). \tag{1}$$

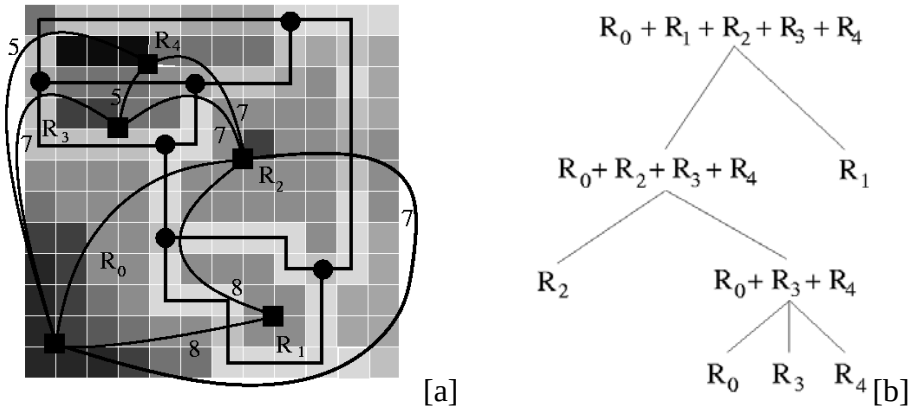


Figure 2: (a) The dual pair $(\overline{G_{2n}}, G_{2n})$. The circular vertices belong to G_{2n} and the square vertices belong to $\overline{G_{2n}}$. (b) The hierarchy of the regions from the pyramid on top of $(\overline{G_{2n}}, G_{2n})$.

The vertices of $\overline{G_{2n+2i}}$ represent the regions of the nested morphological segmentation. The order of these regions with respect to set inclusion defines the hierarchy of the regions.

3 Many-to-many Matching of Attributed Trees

To match hierarchies of segmentations we used a framework recently introduced in [8], which expands on previous work developed in [7]. The basic idea behind this approach is to cast the tree matching problem as an equivalent maximum weight clique problem. This is in turn mapped onto an equivalent quadratic program which is then (approximately) solved by simple dynamics arising in evolutionary game theory and related fields.

Formally, an *attributed tree* is a triple $T = (V, E, \alpha)$, where (V, E) is the "underlying" rooted tree and $\alpha : V \to \mathcal{A}$ is a function which assigns an attribute vector $\alpha(u)$ to each node $u \in V$. Two nodes $u, v \in V$ are said to be *adjacent* (denoted $u \sim v$) if they are connected by an edge. We shall also consider a function $\delta : \mathcal{A} \to \mathbb{R}_+$ which assigns to each set of attributes (and therefore to each node in the tree) a real positive number. This will be interpreted as the negligibility of the corresponding node in the tree. Specifically, a node will be declared "negligible" if the value of the function δ corresponding to its attributes is smaller than a fixed threshold ϵ. Clusters of nodes that contain only one non-negligible node (w.r.t. ϵ) are called ϵ-*clusters*. For a formal definition see [7]. We associate an ϵ-cluster of negligible nodes in the first subtree to an ϵ-cluster of negligible nodes in the second tree, thereby defining a many-to-many mapping from the first to the second tree.

A relation $M \subseteq V_1 \times V_2$ is called a *subtree ϵ-morphism* if it preserves the hierarchies of the ϵ-clusters in each of the trees. A formal definition is given in [8].

Clearly, in realistic applications, it would be desirable to find a subtree ϵ-morphism which pairs nodes having "similar" attributes. To this end, let σ be any similarity measure on the attribute space, i.e. any (symmetric) function which assigns a positive number to any pair of

attribute vectors.

If M is a subtree ϵ-morphism between two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$, the overall similarity between the matched structures can be defined as follows:

$$S(M) = \sum_{(u,w)\in M} \sigma(\alpha_1(u), \alpha_2(w))$$

The ϵ-morphism M is called a *maximal similarity subtree ϵ-morphism* if we cannot add further matchings to M, while retaining the morphism property. It is called a *maximum similarity subtree ϵ-morphism* if $S(M)$ is the largest among all ϵ-morphisms between T_1 and T_2.

The weighted ϵ-tree association graph (ϵ-TAG) of two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ is the graph $G_\epsilon = (V, E, \omega)$ where $V = V_1 \times V_2$ such that for any two nodes (u, w) and (v, z) in V the level of u in the hierarchy of T_1 equals the level of v in the hierarchy of T_2 and the same applies to the vertices w and z. Again, the levels are the levels of the corresponding clusters [7]. The following result establishes a one-to-one correspondence between the attributed tree morphism problem and the maximum weight clique problem.

Proposition 3.1 *Any maximal (maximum) similarity subtree ϵ-morphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted ϵ-TAG, and vice versa.*

Once the tree morphism problem has been formulated as a maximum weight clique problem, any clique finding algorithm can be employed to solve it (see [1] for a recent review). In the work reported in this paper, we used an approach recently introduced in [7, 2], which is summarized below.

3.1 Matching via game dynamics

Let $G = (V, E, \omega)$ be an arbitrary weighted graph of order n, and let S_n denote the standard simplex of $\mathbb{R}^n$:

$$S_n = \{\, \mathbf{x} \in \mathbb{R}^n \;:\; \mathbf{e}'\mathbf{x} = 1 \text{ and } x_i \geq 0,\ i = 1 \ldots n \,\}$$

where $\mathbf{e}$ is the vector whose components equal 1, and a prime denotes transposition. Given a subset of vertices C of G, we will denote by $\mathbf{x}^c$ its *characteristic vector* which is the point in S_n defined as

$$x_i^c = \begin{cases} \omega(u_i)/\Omega(C), & \text{if } u_i \in C \\ 0, & \text{otherwise} \end{cases}$$

where $\Omega(C) = \sum_{u_j \in C} \omega(u_j)$ is the total weight on C.

Now, consider the following quadratic function

$$f_G(\mathbf{x}) = \mathbf{x}'(\gamma \mathbf{e}\mathbf{e}' - A_G)\mathbf{x} \tag{2}$$

where $A_G = (a_{ij})$ is the $n \times n$ symmetric matrix defined as follows:

$$a_{ij} = \begin{cases} \frac{1}{2\omega(u_i)} & \text{if } i = j \ , \\ 0 & \text{if } i \neq j \text{ and } u_i \sim u_j \ , \\ \frac{1}{2\omega(u_i)} + \frac{1}{2\omega(u_j)} & \text{otherwise} \end{cases} \tag{3}$$

and $\gamma = \max a_{ij}$. The following result allows us to formulate the maximum weight clique problem as a quadratic program, thereby switching from the discrete to the continuous domain (see [2] for proof).

Proposition 3.2 *Let C be a subset of vertices of a weighted graph $G = (V, E, \omega)$, and let A_G be defined as in (3). Then, C is a maximum (maximal) weight clique of G if and only if $\mathbf{x}^C(\mathbf{w})$ is a global (local) maximizer of f_G in S_n. Moreover, all local (and hence global) maximizers of f_G on S_n are strict.*

We now turn our attention to a class of simple dynamical systems that we use for solving our quadratic optimization problem. Let W be a non-negative real-valued $n \times n$ matrix, and consider the following dynamical system:

$$\dot{x}_i(t) = x_i(t) \left[(W\mathbf{x}(t))_i - \mathbf{x}(t)' W \mathbf{x}(t) \right], \quad i = 1 \ldots n \tag{4}$$

where a dot signifies derivative w.r.t. time t, and its discrete-time counterpart

$$x_i(t+1) = x_i(t) \frac{(W\mathbf{x}(t))_i}{\mathbf{x}(t)' W \mathbf{x}(t)}, \quad i = 1 \ldots n \ . \tag{5}$$

It is readily seen that the simplex S_n is invariant under these dynamics, which means that every trajectory starting in S_n will remain in S_n for all future times. Both (4) and (5) are called *replicator equations* in evolutionary game theory, since they are used to model evolution over time of relative frequencies of interacting, self-replicating agents [9].

If $W = W'$ then the function $\mathbf{x}(t)' W \mathbf{x}(t)$ is strictly increasing with increasing t along any non-stationary trajectory $\mathbf{x}(t)$ under both continuous-time (4) and discrete-time (5) replicator dynamics. Furthermore, any such trajectory converges to a stationary point. Finally, a vector $\mathbf{x} \in S_n$ is asymptotically stable under (4) and (5) if and only if $\mathbf{x}$ is a strict local maximizer of $\mathbf{x}' W \mathbf{x}$ on S_n.

The previous result is known in mathematical biology as the fundamental theorem of natural selection [9] and, in its original form, traces back to R. A. Fisher. Motivated by this result, we use (as in [7, 8]) replicator equations as a simple heuristic for solving our attributed tree matching problem. Let $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ be two attributed trees, and let $G = (V, E, \omega)$ be the corresponding association graph. By letting

$$W = \gamma \mathbf{e}\mathbf{e}' - A_G \tag{6}$$

we know that the replicator dynamical systems (4) and (5), starting from an arbitrary initial state, which is usually taken to be the simplex barycenter, will iteratively maximize the function $\mathbf{x}' W \mathbf{x}$ over the simplex and will eventually converge to a strict local optimizer which will then correspond to the characteristic vector of a maximal weight clique in the association graph. This will in turn induce a maximal similarity subtree ϵ-morphism between T_1 and T_2.

4 Weights and Similarity Measures

Matching nested morphological segmentations in a robust way we have to take into account that there are catchment basins which are sensitive to changes of the topography and others that are more stable. The same distinction makes sense for regions obtained by unifying catchment basins. In the following we will define *weights* for regions that reflect the reliability of the regions for the matching. Due to the one-to-one correspondence between the regions of the hierarchical segmentation and the vertices in all $\overline{G_{2n+2i}}$, we may identify the regions with the vertices. The minimal attribute $Att^{min}(r)$ of all edges incident to region/vertex r, i.e.

$$Att^{min}(r) := min\{att(e) \mid r \in \overline{\iota}(e)\} \tag{7}$$

indicates the next higher level of the flood that unifies r with a neighboring region. The maximal attribute Att_{max} of all edges $e \subset r$ (e and r both are subsets of $\mathbb{R}^2$), i.e. $Att_{max}(r) := max\{att(e) \mid e \subset r\}$ indicates the lowest level of the flood at which all sons of r were merged. Let $size(r)$ denote the size of region r, i.e. the number of pixels in r. We define the *weight* of a region r to be

$$weight(r) = (Att^{min}(r) - Att_{max}(r)) * size(r). \tag{8}$$

The *similarity measure* will depend on the application. It can be derived from topological measurements (genus of the regions), geometric measurements of the regions (area, shape) or of the boundaries (perimeter, curvature), or the colors (gray values) of the regions.

5 Experimental Results

To check the algorithms we generated the test images depicted in Fig. 3a-c. The images are composed such that there are different pairs of neighboring regions with the same contrast. Thus, the unification of neighboring basins as defined in Sec. 2 is not unique. In these cases the choice is made by using a random generator. However, each ambiguous unification yields a region of zero weight (Sec. 4) and after contracting the negligible edges (with respect to $\epsilon = 0$) the hierarchies of the images in Fig. 3a-c should be pairwise isomorphic (trees) again. Indeed, we obtained perfect matches between the contracted hierarchies.

We also performed tests on real images. The hierarchies computed from the subimages *l-eye*, *r-eye*, *mouth*, and *nose* in Fig. 3d had 27, 31, 37, and 41 vertices, respectively. Since there is no preferred value for ϵ, we covered a wide range by choosing ϵ such that the number of clusters in the hierarchy of *l-eye* amounted to 24 (all regions with weight greater than 0), 20, 15, and 10, respectively. The corresponding values for ϵ are between 0 and 250. We did not want unreliable regions to contribute to the weights of the cliques in the ϵ-TAG. Hence, we set the weight of a vertex (u, v) in the ϵ-TAG to zero whenever the weight of u or v was smaller or equal to ϵ. In general, the number of clusters for the same ϵ is different in hierarchies from different images. Thus, we have to compensate for the different numbers of clusters if

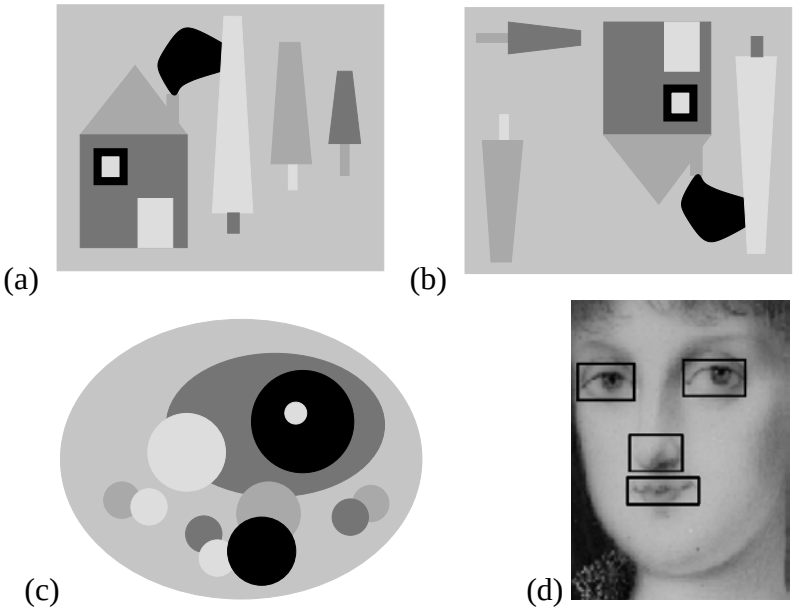


Figure 3: (a-c) Perfectly matched images. (d) The four images *l-eye*, *r-eye*, *mouth*, *nose*.

quantifying the quality of the matches. We calculated the normalized distance [1]

$$dist_\epsilon(T_1, T_2) := 1 - \frac{W_\epsilon(C_{12})}{M\ max(n_\epsilon(T_1), n_\epsilon(T_2))} \quad , \text{ where} \tag{9}$$

- T_1 and T_2 are the attributed trees of the subimages,
- $W_\epsilon(C_{12})$ is the weight of the maximal weight clique C_{12} in the ϵ-TAG of T_1 and T_2,
- $n_\epsilon(T_1)$ denotes the number of ϵ-clusters in T_1, and
- M denotes the upper bound of the similarity function σ.

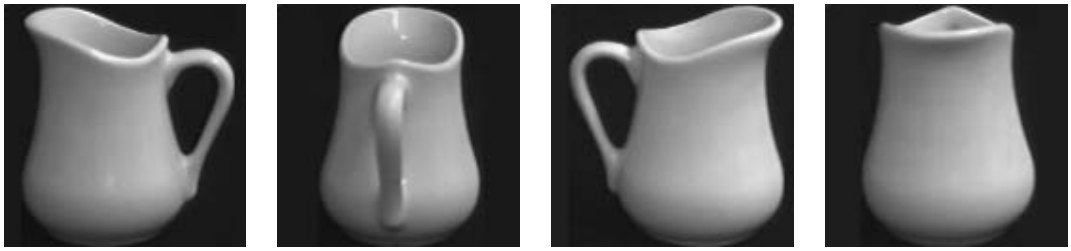
The similarity function is a linear function on the mean gray levels (normalized to $[0, 1]$). Tab. 5 shows the results for $\epsilon = 0$. As for all other ϵ-values tested, the two eyes are most similar, followed by the pair *l-eye* and *mouth*.

Analogous experiments were performed with the images in Fig. 4. For ϵ-values between 3000 and 5000 (see Tab. 5) the two images *pot-0* and *pot-180* have been the most similar ones. Note that the light intensities of the two images are distributed differently and that our method does not make use of shapes.

[1] without proof that the metric axioms are fulfilled.

Table 1: Normalized distances of graphs from subimages of Fig. 3d for $\epsilon = 0$.

$\epsilon = 0$	**l-eye**	**r-eye**	**mouth**	**nose**
l-eye	0.00	0.27	0.42	0.58
r-eye	0.27	0.00	0.45	0.52
mouth	0.42	0.45	0.00	0.51
nose	0.58	0.52	0.51	0.00



(a) pot-0 (b) pot-90 (c) pot-180 (d) pot-270

Figure 4: Images of a pot from the COIL-database.

Table 2: Normalized distances of graphs from pots in Fig. 4 for $\epsilon = 4000$.

$\epsilon = 4000$	**pot-0**	**pot-90**	**pot-180**	**pot-270**
pot-0	0.00	0.32	0.21	0.57
pot-90	0.32	0.00	0.44	0.53
pot-180	0.21	0.44	0.00	0.61
pot-270	0.57	0.53	0.61	0.00

6 Conclusions and Outlook

We proposed a combination of hierarchical segmentation followed by a many-to-many matching of the regions. This combination is well suited to detect structural similarities between images. Robustness is achieved through a weight function and a similarity function for the regions. Our method is invariant to geometrical transformations of homogeneous regions as long as the topological relations between the regions are unchanged. First experiments on real images showed that the matching results correspond to human intuition. In the future we will extend the concept such that the calculation of the hierarchy, as well as the weight and the similarity function may depend on the shape of the regions.

References

[1] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization (Suppl. Vol. A)*, pages 1–74. Kluwer, Boston, MA, 1999.

[2] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator dynamics. *IEEE Trans. Neural Networks*, 11(6):1228–1241, 2000.

[3] R. Glantz and W. G. Kropatsch. Plane embedding of dually contracted graphs. In *Discrete Geometry for Computer Imagery, DGCI'2000*, volume 1953 of *Lecture Notes in Computer Science*, pages 348–357. Springer, 2000.

[4] W. G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Proc.*, 142(6):366 – 374, 1995.

[5] A. Meijsler and J. Roerdink. A Disjoint Set Algorithm for the Watershed Transform. In *Proc. of EUSIPCO'98, IX European Signal Processing Conference*, pages 1665 – 1668, Rhodes, Greece, 1998.

[6] F. Meyer. Graph based morphological segmentation. In Walter G. Kropatsch and Jean-Michel Jolion, editors, *2nd IAPR-TC-15 Workshop on Graph-based Representation*, pages 51–60. OCG-Schriftenreihe, Band 126, Österreichische Computer Gesellschaft, 1999.

[7] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching Hierarchical Structures Using Association Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, 1999.

[8] M. Pelillo, K. Siddiqi, and S. W. Zucker. Many-to-many matching of attributed trees using association graphs and game dynamics. In Carlo Arcelli, Luigi P. Cordella, and Gabriella Sanniti di Baja, editors, *Visual Form 2001*, volume 2059 of *Lecture Notes in Computer Science*, pages 583–593. Springer, 2001.

[9] J. W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, MA, 1995.

Improved Directional Distance Filters

Rastislav Lukac

Department of Electronics and Multimedia Communications,

Technical University of Košice

Park Komenského 13, 041 20 Košice, Slovak Republic

Tel.: +421 55 602 2863, Fax.: +421 55 632 3989

e-mail: lukacr@tuke.sk

Abstract

This paper focuses on a new vector approach for the noisy color images. The proposed method is derived from very interesting class of directional distance filters that combine both the sum of vector distances and the sum of vector angles between input multichannel (vector-valued) samples. Thus, these filters are characterized by the simultaneous noise attenuation and the color chromaticity preservation.

The novelty of the proposed method lies in the considering the sample importance of the input set determined by a filter window, where the largest influence to an estimate provides the central sample. For that reason, the odd integer weight associated with the central sample is included to an account of the sum of vector distances and vector angles. Besides the practical purposes related to significantly improved performance of directional distance filters, another attractive property of the proposed method consists in their theoretical analysis. Clearly, by the simple varying the weight of the central sample, it is possible to achieve special cases of the proposed method such as identity filter, directional distance filter, vector median filter, basic vector directional filter.

In this paper, there will be shown that in term of objective criteria, the proposed improved directional distance filters can provide excellent results related to the color chromaticity preservation near to a threshold of human eyes senselessness.

1 Introduction

In the multichannel image processing [2], [5], [8], [13], [16], [18], the vector approaches represent an attractive and interesting kind of the processing, since there is respected a natural inherent correlation between the color channels. Thus, each image sample is processed as a three-dimensional vector (a number of color channels is equal to three). According to an ordering criterion, where a sum of vector distances or a sum of vector angles to all samples in a filter window can be considered, we differentiate the vector median-based filters or vector directional filters. The vectors' direction signifies the chromaticity of vector samples, while

their magnitude is a measure of their brightness. To combine both kinds of distances, the directional distance filters [3] were developed and thus, these filters are characterized by the simultaneous noise attenuation and color chromaticity preservation. In this case, a filter output represents the sample associated with a minimal sum of vector distances and a minimal sum of vector angles, both to all samples in a filter window.

The aim of this paper is to show that the performance of directional distance filters can be improved significantly by simple introducing the central weight. Thus, the proposed method represents the generalization of many filtering classes, since vector median filter, basic directional filter, distance directional filter and many more are included in the proposed definition as special cases.

This paper is organized as follows. In the next section, the mathematical preliminaries and the definitions of directional distance filters are presented. Section 3 focuses on a new method, namely improved directional distance filters with the weight of central sample are defined and described in detail. In Section 4, the vector definition of the impulse noise for color images is described. Three objective criteria are defined, including well-known mean absolute error, mean square error and criterion for the color chromaticity preservation. The properties of the improved directional distance filters are concluded in Section 5.

2 Directional Distance Filters

Let $y(x): Z^l \to Z^m$ represent a multichannel image, where l is a image dimension and m characterises a number of channels. If $m \geq 2$, it is the case of m-channel image processing. In the case of standard color images $l = 2$ and $m = 3$. Let $W = \{\mathbf{x}_i \in Z^l \; ; \; i = 1, 2..., N\}$ represent a filter window of a finite size N, where $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ is a set of noised samples. Note, that the position of filter window is determined by the central sample $\mathbf{x}_{(N+1)/2}$.

Each input vector $\mathbf{x}_i$ can be associated with the vector distance L_i and vector angle α_i, both respect all samples in a filter window. Mathematically, the vector distance L_i is defined by [1]

$$L_i = \sum_{j=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \quad \text{for } i = 1, 2, ..., N \tag{1}$$

where γ represents an appropriate norm, $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{im})$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, ..., x_{jm})$ are m-dimensional vectors.

Likewise, the angle distance α_i [12] is expressed as

$$\alpha_i = \sum_{j=1}^{N} \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for } i = 1, 2, ..., N \tag{2}$$

where

$$\mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1}\left(\frac{\mathbf{x}_i . \mathbf{x}_j^T}{|\mathbf{x}_i| . |\mathbf{x}_j|} \right) \tag{3}$$

represents the angle between m-dimensional vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

Now, let each input sample $\mathbf{x}_i$ be associated with Ω_i, i.e. product of L_i and α_i, defined by [3]

$$\Omega_i = L_i . \alpha_i \quad \text{for } i = 1, 2, ..., N$$

$$\Omega_i = \left(\sum_{j=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \right) . \left(\sum_{j=1}^{N} \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \right) \quad \text{for } i = 1, 2, ..., N \tag{4}$$

Thus, the vector distances and vector angles are included in (4). In order to control the influence of L_i and α_i on Ω_i, the power parameter p can be introduced. Then [3]

$$\Omega_i = L_i^{1-p}.\alpha_i^p \quad \text{for } i = 1,2,...,N$$

$$\Omega_i = \left(\sum_{j=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \right)^{1-p} . \left(\sum_{j=1}^{N} \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \right)^{p} \quad \text{for } i = 1,2,...,N \tag{5}$$

The ordering of noisy samples $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ represents the base of the order-statistic theory [7], [9], usually used in the case of the impulse noise corruption. In the case of color images, i.e. vector-valued image signals, the direct extension of the robust order-statistic theory is impossible [11], [14] and observed samples are ordered according to the distance function, where both magnitude [1], [4], [8] and direction [5], [6], [12], [16] of multichannel samples can be considered. In general, vectors' magnitude takes a measure of their brightness, whereas the direction of vector samples wreaks their chromaticity [15].

If the vector distances $L_1, L_2, ..., L_N$ between the input samples in the vector space serve as ordering criterion, it is the case of magnitude processing whose typical representatives are vector median-based filters [1], [4]. In this case, the ordering of $L_1, L_2, ..., L_N$ expressed as

$$L_{(1)} \le L_{(2)} \le ... \le L_{(N)} \tag{6}$$

and it means that the same ordering is implied to the input set $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ which results in ordered input sequence

$$\mathbf{x}^{(1)} \le \mathbf{x}^{(2)} \le ... \le \mathbf{x}^{(r)} \le ... \le \mathbf{x}^{(N)} \tag{7}$$

The output of vector median filter (VMF) is a sample $\mathbf{x}^{(1)}$, i.e. a sample with minimal vector distance $L_{(1)}$ to all samples in a filter window.

If the vectors' direction in the vector space between the input samples serves as ordering criteria then it is the case of directional processing and vector directional filters. Then an ordered input sequence (7) is achieved according to ordered vector angles expressed as [12]

$$\alpha_{(1)} \le \alpha_{(2)} \le ... \le \alpha_{(r)} \le ... \le \alpha_{(N)} \tag{8}$$

and a sample $\mathbf{x}^{(1)}$ associated with minimal angle distance $\alpha_{(1)}$, i.e. a sample that minimises the sum of angles with other vectors, represents an output of the basic vector directional filter (BVDF) [12], [15], [16]. Since, the vector directional filters (VDF) pass to a filter output a sample from the set ordered according to a sum of vector angles, these filters preserve the color chromaticity rather than suppress the noise.

If the ordering criterion is based on both vector distances and vector angles, then it is the case of distance directional filters (DDF's) [3] that improve the smoothing property of vector directional filters and the color chromaticity preservation of vector median-based filters, simultaneously. In addition, above mentioned filters are included as special cases of (DDF's).

Mathematically, DDF's are outputting the sample $\mathbf{x}^{(1)}$, given by (7), associated with a minimal value from $\Omega_1, \Omega_2, ..., \Omega_N$ expressed as

$$\Omega_{(1)} \le \Omega_{(2)} \le ... \le \Omega_{(N)} \tag{9}$$

Although the minimisation of products $L_i.\alpha_i$, for $i = 1,2,...,N$ does not necessarily imply a minimum for either of L_i and α_i, it results in very small values for both of them [3]. For that reason, the product minimisation will select as the output vector-valued sample the one that results in a very small sum of vector distances (1) and a very small sum of vector angles (3), simultaneously.

3 Proposed Improvement

Let $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ be an input set determined by a filter window and N represent a window size. Let us assume that $w_1, w_2, ..., w_N$ defined by [4], [7],

$$w_j = \begin{cases} N - 2k + 2 & \text{for } j = (N+1)/2 \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

represent a set of nonnegative integer weights so that each weight w_j, for $j = 1,2,...,N$, is associated with the input sample $\mathbf{x}_j$. Clearly, only the central weight $w_{(N+1)/2}$ associated with the central sample $\mathbf{x}_{(N+1)/2}$ can be alternated, whereas other weights associated with the neighbouring samples are retained to be equal to one. Note that $k = 1,2,...,(N+1)/2$ is a smoothing parameter. Then, it is possible to express the weighted vector distance J_i, [4], [8]

$$J_i = \sum_{j=1}^{N} w_j \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \quad \text{for } i = 1,2,...,N \tag{11}$$

and the weighted angle distance β_i, [5], [6],

$$\beta_i = \sum_{j=1}^{N} w_j \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for } i = 1,2,...,N \tag{12}$$

either of them associated with the input sample $\mathbf{x}_i$.
The combined weighted distance is expressed as

$$\Psi_i = J_i . \beta_i \quad \text{for } i = 1,2,...,N$$

$$\Psi_i = \left(\sum_{j=1}^{N} w_j \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \right) . \left(\sum_{j=1}^{N} w_j \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \right) \quad \text{for } i = 1,2,...,N \tag{13}$$

Similarly to equation (5), there is possible to introduce the power parameter p and thus,

$$\Psi_i = J_i^{1-p} . \beta_i^{p} \quad \text{for } i = 1,2,...,N$$

$$\Psi_i = \left(\sum_{j=1}^{N} w_j \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^{\gamma} \right)^{1-p} . \left(\sum_{j=1}^{N} w_j \mathrm{A}(\mathbf{x}_i, \mathbf{x}_j) \right)^{p} \quad \text{for } i = 1,2,...,N \tag{14}$$

The output of the proposed method can be expressed as

$$\mathbf{y}_k = \mathbf{x}^{(1)} \tag{15}$$

where $\mathbf{x}^{(1)}$ (7) is the ordered vector-valued sample associated with minimal weighted combined distance $\Psi_{(1)}$ according to

$$\Psi_{(1)} \le \Psi_{(2)} \le ... \le \Psi_{(N)} \tag{16}$$

The importance and the estimation accuracy of the proposed method lies in the incorporating of the temporal-order information or a sample importance (expressed by the central weight $w_{(N+1)/2} = N - 2k + 2$ associated with the central sample $\mathbf{x}_{(N+1)/2}$) of the input set.
It is clear that the proposed method represents a generalisation of the VMF, BVDF, DDF and many more. Consider definition (13) or definition (14) for $p = 0.5$. If the smoothing parameter k is equal to its minimal value, i.e. $k = 1$, then the proposed method is equivalent to an identity filter and no smoothing will be provided. In the case of maximal value of parameter k, i.e. $k = (N+1)/2$, the maximal amount of the smoothing will be performed, however, a filter can provide too much smoothing that will result in a blurring. Clearly, the amount of smoothing increases with the increased parameter k. It means that the filter

preserves the signal details for small value of k and well suppress the noise for its larger value. Varying the filter parameter between its minimal and maximal values, it is possible to achieve the best balance between the noise suppression and signal details preservation.
Of course, another filter classes such as VMF, center-weighted VMF [8], BVDF and center-weighted VDF [6] can be expressed through the setting of the smoothing parameter k and power parameter p. All included filter generalisations are presented in Table 1.

Table 1 Generalisation of the filter classes

Filter class	k	p
Identity filter	1	<0,1>
VMF	(N+1)/2	0
Center-weighted VMF	{1,3,...,(N+1)/2}	0
BVDF	(N+1)/2	1
Center-weighted BVDF	{1,3,...,(N+1)/2}	1
DDF	(N+1)/2	0.5
Center-Weighted DDF	{1,3,...,(N+1)/2}	0.5

In Table 1 is shown that the proposed method is equivalent to an identity filter for arbitrary possible value of p (parameter p should be from interval $\langle 0,1 \rangle$) and smoothing parameter $k = 1$. If smoothing parameter has the maximal possible value. i.e. $k = (N+1)/2$, it is possible to express VMF (for $p = 0$), BVDF ($p = 1$) and DDF ($p = 0.5$). For the case $k = 1,3,...,(N+1)/2$, the output of the proposed method is equivalent to center-weighted VMF (for $p = 0$), center-weighted VDF ($p = 1$) and the proposed center-weighted DDF ($p = 0.5$). The last theorem is correct, since above center weighted filters include VMF, BVDF and DDF as the special cases, separately.

4 Experimental Results

As the test image was used well-known color image Lena (Figure 1a). The noise corruption (Figure 1b) was simulated by the impulse noise (Fig.1b) that is defined by [6], [8]

$$\mathbf{x}_{i,j} = \begin{cases} \boldsymbol{\upsilon} & \text{with probability } p_{\boldsymbol{\upsilon}} \\ \mathbf{o}_{i,j} & \text{with probability } 1 - p_{\boldsymbol{\upsilon}} \end{cases} \tag{17}$$

where i, j characterize sample position, $\mathbf{o}_{i,j}$ is the sample from the original image, $\mathbf{x}_{i,j}$ represents the sample from the noisy image, $p_{\boldsymbol{\upsilon}}$ is a corruption probability and $\boldsymbol{\upsilon} = (\upsilon_R, \upsilon_G, \upsilon_B)$ is a noise vector of intensity random values. Since, single components of $\boldsymbol{\upsilon}$ are generated independently, the gray impulse, i.e. an equivalence of all components of $\boldsymbol{\upsilon}$ ($\upsilon_R = \upsilon_G = \upsilon_B$), can occur in the special case, only.
As a measure of the noise corruption and the filter performance, too, three objective criteria [10], [13], [17], namely mean absolute error (MAE), mean square error (MSE) and color difference (CD), are used. In general, MAE is a mirror of the signal-details preservation, MSE evaluates the noise suppression well and CD is a measure of the color chromaticity preservation. Thus, the quality of the processed image sequences is quantified with a high accuracy related to the signal dimensionality.

(a) (b) (c)

(d) (e) (f)

Figure 1 Achieved Results
(a) Original image (b) Impulse noise ($p_v = 0.1$) (c) Output of VMF
(d) Output of marginal median (e) Output of BVDF (f) Output of DDF

Mathematically, the definitions of MAE and MSE for monochromatic images are given by

$$MAE = \frac{1}{KL}\sum_{i=1}^{K}\sum_{j=1}^{L}\left|o_{i,j} - x_{i,j}\right| \tag{18}$$

$$MSE = \frac{1}{KL}\sum_{i=1}^{K}\sum_{j=1}^{L}\left(o_{i,j} - x_{i,j}\right)^2 \tag{19}$$

where $\{o_{i,j}\}$ is the original image, $\{x_{i,j}\}$ is the filtered (noisy) image, i, j are indices of sample position and K, L characterize an image size. Note that in the case of color images, MAE and MSE criteria are understood as a mean over color channels.
Finally, the measure of color distortion or color chromaticity preservation is evaluated by CD that requires transformation from RGB to Luv color space [13]. For a color image, the CD is expressed as

$$\Delta E_{Luv} = \sqrt{(\Delta L)^2 + (\Delta u)^2 + (\Delta v)^2} \tag{20}$$

where $\Delta L, \Delta u$ and Δv represent the difference between original and noisy images in L, u and v color channels. The overall value of CD is a mean value over all frames. Unlike MAE and MSE, in the case of CD was established the threshold value around 2.9 that characterizes the senselessness of human eyes to color distortion.

(a) (b) (c)

(d) (e) (f)

Figure 2 Achieved Results

(a) Output of center-weighted VMF ($k = 3$) (b) Output of center-weighted VMF ($k = 4$)
(c) Output of center-weighted VDF ($k = 3$) (d) Output of center-weighted VDF ($k = 4$)
(e) Output of center-weighted DDF ($k = 3$) (f) Output of center-weighted DDF ($k = 4$)

Table 2 Performance of the methods

Filter class	MAE	MSE	CD
Identity filter	7.312	832.0	32.717
Marginal median	3.703	56.8	17.777
Center-weighted VMF (k = 2)	3.667	369.4	16.425
Center-weighted VMF (k = 3)	1.438	55.6	5.957
Center-weighted VMF (k = 4)	1.995	32.4	8.226
VMF (Center-weighted VMF for k = 5)	3.687	56.5	15.396
Center-weighted VDF (k = 2)	3.228	304.7	12.430
Center-weighted VDF (k = 3)	1.632	62.5	5.617
Center-weighted VDF (k = 4)	2.393	42.9	8.817
BVDF (Center-weighted VDF for k = 5)	4.099	67.6	15.343
Center-weighted DDF (k = 2)	3.509	351.9	14.965
Center-weighted DDF (k = 3)	1.130	48.1	5.285
Center-weighted DDF (k = 4)	1.987	31.3	8.135
DDF (Center-weighted DDF for k = 5)	3.733	57.3	15.135

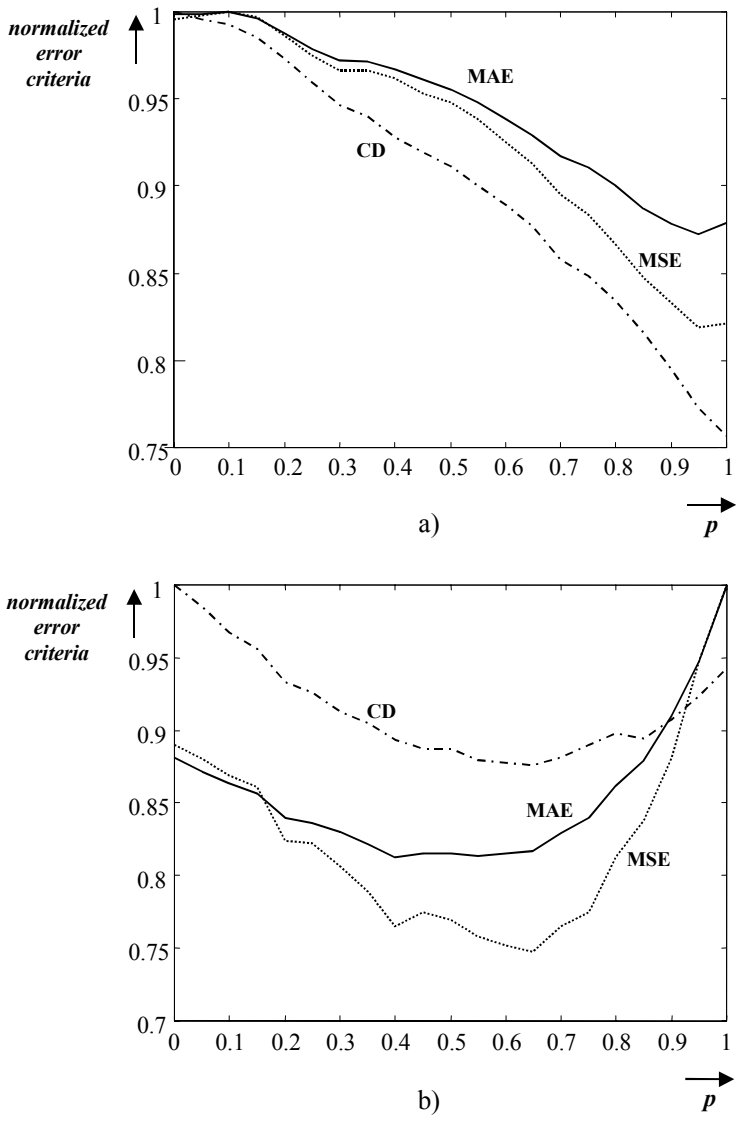
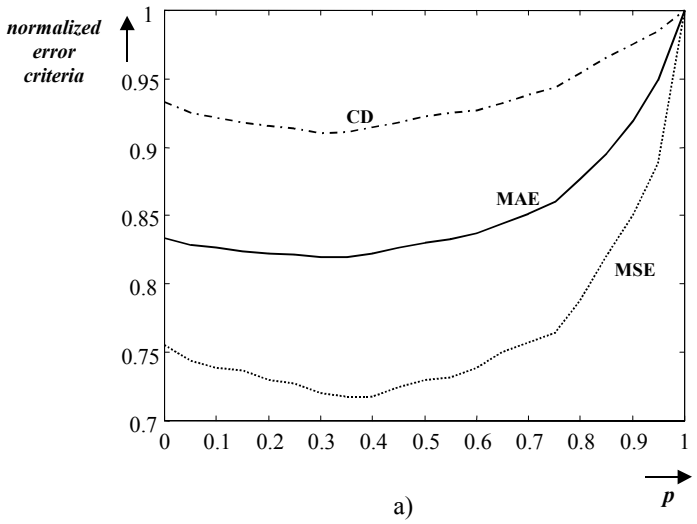
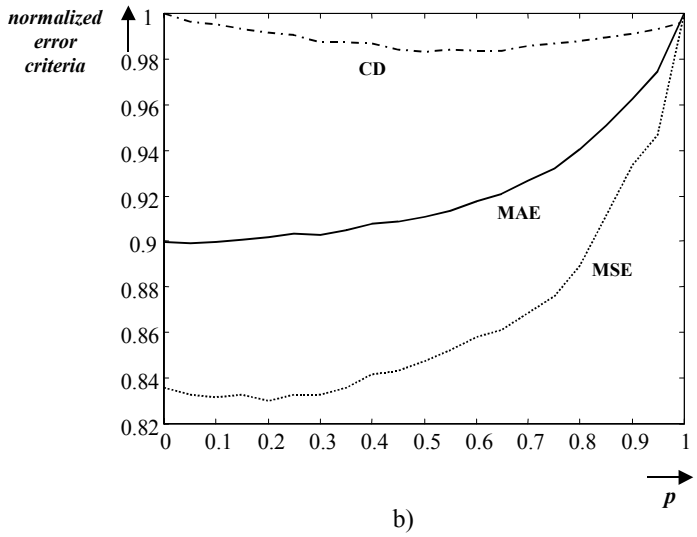


Figure 3 Performance of the proposed method: Dependence of normalized error criteria on power parameter p. a) $k = 2$, b) $k = 3$

In order to determine the efficiency of the proposed method, see Table 2, Figure 1 and Figure 2. From Table 2, it can be seen that the proposed method achieves the significant improvement in comparison with marginal median, VMF, BVDF, DDF and the center weighted structures of above mentioned filters. It is clear, that a noise attenuation capability of the proposed method increases with the increased smoothing parameter k. The small amount of the smoothing results in the impulse presence (Figure 2e), whereas the robust smoothing capability is provided for larger values of parameter k (Figure 2f). In this paper, the best results were achieved by the proposed method with parameter values $k = 3$ and $k = 4$. Additional results are provided in Figure 3 and Figure 4, where is presented the performance (expressed through normalized error criteria) of the proposed method in dependence on a power parameter p and smoothing parameter k.



a)



b)

Figure 4 Performance of the proposed method: Dependence of normalized error criteria on power parameter p. a) $k = 4$, b) $k = 5$

5 Conclusions

The new class of center-weighted directional distance filters (CWDDF), especially for the impulse noise suppression in color images, has been provided. The proposed method represents the generalisation of identity filter, vector median filter, center-weighted vector median filter, basic vector directional filter and center-weighted vector directional filter. The experimental results showed the excellent performance of the proposed method, where the evaluation of color chromaticity preservation was nearly to the threshold of human eyes senselessness. According to the possibility of adaptive controlled power parameter, the future research tasks are related to the searching of the adaptive structures of CWDDFs.

References

[1] Astola, J., Haavisto, P. and Neuvo, Y., "Vector Median Filters", *Proceedings of the IEEE*, Vol. 78, No. 4, pp. 678-689, April 1990.

[2] Gabbouj, M. and Cheickh, F.A., "Vector Median-Vector Directional Hybrid Filter for Color Image Restoration", *Proceedings of EUSIPCO-96*, pp. 879-881, 1996.

[3] Karakos, D.G. and Trahanias, P.E., "Generalized Multichannel Image-Filtering Structure", *IEEE Trans. on Image Processing*, Vol. 6, No. 7, pp. 1038-1045, July 1997.

[4] Lukáč, R., "Vector LUM Smoothers as Impulse Detector for Color Images", *Proceedings of ECCTD '01*, Vol. III, pp. 137–140, August 2001.

[5] Lukáč, R., "Weighted Vector Directional Filters", *IEEE Signal Processing Letters*, submitted.

[6] Lukáč, R., "Adaptive Impulse Noise Filtering by Using Center-Weighted Directional Information", *Proceedings of CGIV '02*, pp. ?, April 2002.

[7] Lukáč, R. and Marchevský, S., "LUM Smoother with Smooth Control for Noisy Image Sequences", *EURASIP Journal on Applied Signal Processing*, Vol. 2001, No. 2, pp. 110-120, 2001.

[8] Lukáč, R. and Marchevský, S., "Adaptive Vector LUM Smoother", *Proceedings of the ICIP 2001*, Vol. 1., pp. 878-881, October 2001.

[9] Lukáč, R. and Marchevský, S., "Boolean Expression of LUM Smoothers", *IEEE Signal Processing Letters*, Vol. 8, No. 11, pp. 292-294, November 2001.

[10] Ochodnický, J.: *Multisenzor Networks. Principle, Data Association and Objects Tracking*. Army Academy, Liptovský Mikuláš, 2001 (in Slovak).

[11] Pitas, I. and Tsakalides, P., "Multivariate Ordering in Color Image Filtering", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 1, No. 3, pp. 247-259, September 1991.

[12] Platanoitis, K.N., Androutsos, D. and Venetsanopoulos, A.N., "Color Image Processing Using Adaptive Vector Directional Filters", *IEEE Transactions on Circuits and Systems II*, Vol. 45, No. 10, pp. 1414-1419, October 1998.

[13] Sharma, G., "Digital Color Imaging", *IEEE Transactions on Image Processing*, Vol. 6, No. 7, pp. 901-932, July 1997.

[14] Tang, K., Astola, J. and Neuvo, Y., "Nonlinear Multivariate Image Filtering Techniques", *IEEE Trans. on Image Processing*, Vol. 4, No. 6, pp. 788-798, June 1995.

[15] Trahanias, P.E. and Venetsanopoulos, A.N., "Vector Directional Filters - a New Class of Multichannel Image Processing Filters", *IEEE Transactions on Image Processing*, Vol. 2, No. 4, pp. 528-534, April 1993.

[16] Trahanias, P.E., Karakos, D. and Venetsanopoulos, A.N., "Directional Processing of Color Images: Theory and Experimental Results", *IEEE Transactions on Image Processing*, Vol. 5, No. 6, pp. 868-881, June1993.

[17] Turan, J.: *Fast Translation Invariant Transforms and their Applications*. Elfa, Košice 1999.

[18] Zheng, J, Valavanis, K.P. and Gauch, J.M.: "Noise Removal from Color Images", *Journal of Intelligent and Robotic Systems*, Vol. 7, pp. 257-285, 1993.

Statistical Model-Based Segmentation of Articulated Structures

Rok Bernard, Boštjan Likar, Franjo Pernuš

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, Ljubljana, Slovenia

e-mail: {rok.bernard, bostjan.likar, franjo.pernus}@fe.uni-lj.si

Abstract

This paper describes a general method for segmenting articulated structures composed of several anatomical structures. The method is based on statistical parametrical models, obtained by principal component analysis (PCA). The models, which describe shape, appearance, and topology of anatomical structures, are incorporated in a two-level hierarchical scheme. Shape and appearance models, describing plausible variations of shapes and appearances of individual anatomical structures, form the lower level, while the topological model, describing plausible topological variations of the articulated structure, forms the upper level. This novel scheme is actually a hierarchical PCA as the topological model is generated by the PCA of the parameters obtained at the lower level. In the segmentation process, we seek the configuration of the model instances that best matches the given image. For this purpose we introduce coarse and fine matching strategies for minimizing an energy function, which is a sum of a match measure and deformation energies of topology, shape, and appearance. The proposed method was evaluated on 36 X-ray images of cervical vertebrae by a leave-one-out test. The results show that the method well describes the anatomical variations of the cervical vertebrae, which confirms the feasibility of the proposed modeling and segmentation strategies.

1 Introduction

Ascertaining the detailed shape and organization of anatomical structures is important not only within diagnostic settings but also for tracking the process of disease, surgical planning, simulation, and intraoperative navigation. Accurate and efficient automated segmentation of articulated structures, composed of several anatomical structures, is difficult because of their complexity and inter-patient variability. Furthermore, the position of the patient during image acquisition, the imaging device itself, and the imaging protocol induce additional variations in shape and appearance. To deal with the variations, a segmentation method should use as much available prior information on shape, location, and appearance of the analyzed structures as possible. When segmenting articulated structures, like the spine, knee, or hand, prior knowledge on topology, i.e. organization of anatomical structures, should also be considered.
In recent years, a great variety of shape and appearance models have been proposed as a source of prior knowledge and applied to various tasks in medical image analysis [1].

Efficient models should be general to deal with inter-patient variability and yet specific to maintain certain anatomical properties [1, 2]. Models, which are trained on a set of labeled training images meet these requirements and have therefore received much attention. For example, point distribution models, active shape models, and active appearance models, all proposed by Cootes *et al.* [3, 4], were successfully applied to bony structures, e.g., vertebrae [5], spine [6], knee joint [4], hand [7], rib cage [8] or hip and pelvis [9], most often for segmentation purposes.
Articulated structures exhibit two kinds of shape variations, i.e. variations in shapes of individual anatomical structures and variations in spatial relationships between them. Such combined variations cannot be optimally described by a single linear model unless variations of spatial relationships are sufficiently small and a sufficiently large training set is used [6]. Therefore, alternative approaches are required to describe the non-linear shape variations. This can be assessed by a piecewise linear models [10] or by separately modeling the variations of spatial relationships and variations of shapes of individual anatomical structures [7]. The problem with piecewise linearization is that it can only approximate the non-linear shape variations without using prior knowledge on organization of articulated structures, while in [7] the prior knowledge is used only for model initialization and not throughout the matching process.

In this paper we propose a general statistical hierarchical modeling of shape, appearance, and topology of articulated structures, which efficiently deals with non-linear shape variations and incorporates prior knowledge on organization of articulated structures. The hierarchical scheme is comprised of two levels. Shape and appearance models, which describe individual anatomical structures form the lower level, while the topological model, which describes the organization of anatomical structures, forms the upper level and supervises spatial relations between individual models at the lower level. The proposed method is applied to the segmentation of cervical spine vertebrae.

2 Hierarchical Scheme

To build up a general scheme that can describe the shape and appearance variations of anatomical structures, such as vertebrae, and the topological variations of the articulated structures, e.g. the cervical spine, we use the principal component analysis (PCA), which is a well-known statistical tool [11]. By PCA the principal variations of average shape, appearance, and topology can be derived from a set of representative training images.

2.1 Principal Component Analysis

Principal component analysis (PCA) is based on the statistical representation of a random variable [11]. Suppose we have a random vector population $\mathbf{x}$ and the mean of that population is denoted by $\bar{\mathbf{x}}$; $\bar{\mathbf{x}} = E(\mathbf{x})$. The covariance matrix of the same data set is $\mathbf{C}$:

$$\mathbf{C} = E\left((\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right) . \quad (1)$$

From a symmetric matrix such as **C** we can define an orthogonal basis by finding its eigenvalues and eigenvectors. By ordering the eigenvectors φ_i in the order of descending eigenvalues $\lambda_i \geq \lambda_{i+1}$, one can create an ordered orthogonal basis with the first eigenvector having the direction of largest variance of the data. Data may be reconstructed by a linear combination of orthogonal basis vectors. Instead of using all the eigenvectors of the covariance matrix, we may represent the data in terms of only a few basis vectors of the orthogonal basis. Let *t* largest eigenvalues and corresponding eigenvectors be retained to form the matrix $\mathbf{\Phi}$; $\mathbf{\Phi}=(\varphi_1|\varphi_2|\ldots|\varphi_t)$. Knowing $\bar{\mathbf{x}}$ and matrix $\mathbf{\Phi}$, we can reconstruct the input data vector **x**:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{y} \ , \tag{2}$$

from the parameters **y** of the statistical model. If the data is concentrated in a linear subspace, this provides a way to compress data without losing much information and simplifies the representation. Alternatively, the input data vector **x** can be transformed into vector **y**:

$$\mathbf{y} = \mathbf{\Phi}^T(\mathbf{x} - \bar{\mathbf{x}}) \ . \tag{3}$$

By the above statistical model we can describe shape, appearance, and topology of an articulated structure as shown below.

2.2 Pose

Pose is for each anatomical structure defined separately in the global coordinate system common to all anatomical structures. The pose parameter vector $\mathbf{y}_p=[x, y, \gamma, m]^T$ is composed of translation in *x* and *y* direction, rotation γ and scale *m*.

2.3 Shape

Each anatomical structure is described by a statistical shape model as proposed by Cootes *et al.* [3]. The model is derived from a set of training shapes. Each training shape is composed of anatomical points defined in training images. Prior to defining the mean shape of a structure, the training shapes are rigidly aligned [3]. Shape variations are found by the PCA of training sets of anatomical points and represented by the most significant eigenshapes. Shape reconstruction is performed by setting shape parameter vector $\mathbf{y}_s$.

2.4 Appearance

The appearance, i.e., the texture of each anatomical structure is modeled on shape-free training images, obtained by elastic registration of training shapes and mean shape. Thin-plate splines interpolation between corresponding anatomical points is used for this purpose [12]. By applying PCA to the set of shape-free training images, defined on a region of interest covering a structure, the mean image and the most significant eigenimages are extracted.

2.5 Topology

To describe topological variations of an articulated structure we need to correlate variations in pose and shape of all anatomical structures. We propose to apply the PCA on pose and shape parameters of all anatomical structures. Let merge pose $\mathbf{y}^i_p$ and shape parameter vectors $\mathbf{y}^i_s$ of structure i into a vector $x^i_t=((\mathbf{y}^i_p)^T \mid (\mathbf{y}^i_s)^T)^T$. Next, merge vectors x^i_t into a vector $x_t=((\mathbf{x}^1_t)^T \mid (\mathbf{x}^2_t)^T \mid \ldots \mid (\mathbf{x}^i_t)^T \mid \ldots)^T$, which holds pose and shape parameters of all anatomical structures in the image. The pose parameters $\mathbf{y}^i_p$ of i-th anatomical structure in the image are obtained by rigid alignment of its average shape to the corresponding shape in the image. Shape parameters $\mathbf{y}^i_s$ are then estimated by its statistical shape model. A model of topology is built from a population of vector x_t, obtained from the set of training images. In this way, the most significant eigentopologies describe the anatomically plausible topological variations.

To make model of topology invariant on global pose variations of the articulated structure in training images we preliminary rigidly align all of the training images by aligning all anatomical points regardless to which anatomical structure they belong.

This novel strategy can be viewed upon as a hierarchical PCA. The topological PCA (upper level of hierarchy), describing plausible topological variations of an articulated structure, is constructed from sets of parameters generated by shape PCAs and corresponding pose parameters (lower level of hierarchy) that describe plausible variations of shapes and poses of individual anatomical structures. In this way, the topological PCA enables the supervision of the spatial relations between shapes of anatomical structures, which form the articulated structure.

3 Segmentation

The above hierarchical scheme consists of parametrical models that describe shape, appearance, and topology of an articulated structure. Once extensively trained, it incorporates a valuable prior knowledge that can be used efficiently for describing the image of the articulated structure. We consider model-based image segmentation by searching the configuration $\mathbf{L}$ of the model instances that best match the given image I. The best configuration $\mathbf{L}^*$ may be found by the maximum a posteriori (MAP) estimation:

$$\mathbf{L}^* = \arg\max_{\mathbf{L}} P(\mathbf{L} \mid I) \ . \tag{4}$$

Bayes rule then implies:

$$\mathbf{L}^* = \arg\max_{\mathbf{L}} P(\mathbf{L})P(I \mid \mathbf{L}) \ . \tag{5}$$

The prior $P(\mathbf{L})$ is given by the probability distributions of shapes, appearances, and topology. The likelihood function $P(I|\mathbf{L})$, measures the probability of observing image I given a particular configuration $\mathbf{L}$. The standard approach to finding the MAP estimation is to minimize the energy function $F(I,\mathbf{L})$ obtained by taking the negative logarithm of a posteriori probability:

$$\mathbf{L}^* = \arg\min_{\mathbf{L}} F(I, \mathbf{L}) \quad . \tag{6}$$

The prior $P(\mathbf{L})$ and likelihood function $P(I|\mathbf{L})$ are turned into internal energy, i.e. deformation, and external energy, i.e. match measure, composing the energy function $F(I,\mathbf{L})$. The required matching strategy and the energy function are given in the following sub-sections.

3.1 Matching Strategy

Consider the configuration **L** describing an articulated structure composed of N anatomical structures where each anatomical structure is described by t_s shape, t_a appearance, and t_p pose parameters. The number of all parameters is $N\cdot(t_s+ t_a + t_p)$, possibly causing a demanding optimization problem. To overcome this problem, we can elegantly omit the appearance parameters as they may be estimated from the image patch defined by the shape model. This reduces the number of parameters to $N\cdot(t_s + t_p)$. We name this optimization strategy a fine matching strategy. We consider also a coarse matching strategy by which the number of parameters can be further significantly reduced to $t_P + t_T$ by tuning only t_P global pose and t_T topological parameters of the articulated structure at the upper level in the hierarchy. Global pose and topological parameters then drive pose and shape parameters of individual anatomical structures at the lower level of hierarchy. The coarse and fine matching strategies are considered in the following.

Coarse Matching
In the coarse matching step, illustrated in Figure 1a, we tune only global pose and topological parameters of the articulated structure that in turn drive pose and shape parameters of the anatomical structures. According to these parameters, each shape model, describing a corresponding anatomical structure, generates a shape that defines a patch on the underlying image. The patch is then elastically transformed to the shape-free form, which is fed into the appearance model that yields appearance parameters and approximates the given shape-free image patch. Finally, the match measure between the shape-free image patch and its approximation is calculated. The obtained match measure is part of the energy function that is used for selecting the global pose and topological parameters for the next iteration in the optimization process. The energy function, which considers also topology, shape, and appearance deformation energies, is described latter.

Fine Matching
In a fine matching strategy, illustrated in Figure 1b, the pose and shape parameters of all anatomical structures are optimized simultaneously, whereas the model of topology only supervises the spatial relations between shapes of anatomical structures via the topology deformation energy in the energy function.

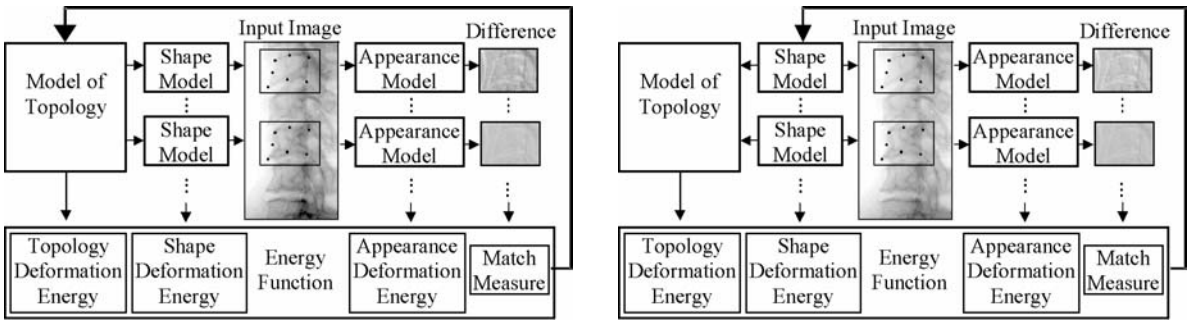


Figure 1. Coarse matching strategy (a): global pose and topological parameters are optimized and fine matching strategy (b): pose and shape parameters of all anatomical structures are optimized simultaneously

3.2 Energy Function

To suppress anatomically implausible configurations we define the energy function $F(I,\mathbf{L})$ as a weighted sum of match measure $M(I,\mathbf{L})$ and topology $F_T(\mathbf{L})$, shape $F_S(\mathbf{L})$, and appearance $F_A(\mathbf{L})$ deformation energies:

$$F(I,\mathbf{L}) = \alpha \cdot M(I,\mathbf{L}) + F_T(\mathbf{L}) + F_S(\mathbf{L}) + F_A(\mathbf{L}) \ , \tag{7}$$

where α is a regularization parameter weighting the match measure against deformation energies.
The mean square of intensity differences between shape-free image patches and their corresponding approximations was chosen as the match measure:

$$M(I,\mathbf{L}) = \sum_{i=1}^{N} \frac{1}{\Omega_i} \frac{1}{V_i} \sum_{j \in \Omega_i} r_{i,j}^2 \ , \tag{8}$$

where $r_{i,j}$ is the intensity difference of j-th pixel in an image patch i defined on a region of interest Ω_i and V_i is the variance of the sum of the squares of intensity differences [13].
The deformation energies are calculated as a weighted sum of corresponding PCA parameters. The weights correspond to the probability density functions of PCA parameters. In this way, a configuration **L** that is not anatomically plausible is penalized.

4 Results

The proposed method was evaluated on 36 X-ray images of cervical vertebrae by a leave-one-out test. The annotated images were taken from the NHANES II X-ray database [14]. Vertebrae 3, 4, 5, and 6 were modeled by placing seven landmarks on each of them (Figure 2a). The number of shape parameters t_s was set to 4 (capturing ~72% of all shape variations), the number of appearance parameters was set to 3 (capturing ~85% of all appearance variations), and the number of topological parameters t_T was set to 2 (capturing ~40% of all topological variations). The regularization parameter α was set to 100. The weighting

function of PCA parameters was defined as $W(y_k)=sign(|y_k|-b)\cdot(|y_k|-b)$, so that the parameters y_k had no influence on the corresponding deformation energy if lying inside the interval [$-b$, b]. The values of b were 1, 0, and 1 for shape, appearance and topological parameters, respectively. The simulated annealing global optimization method was used for energy minimization [14].

Figure 2a. Seven landmarks on vertebrae 3, 4, 5, and 6

Figure 2b. Initialization points in the centre of vertebrae 3 and 6

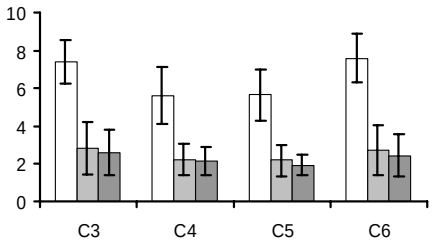
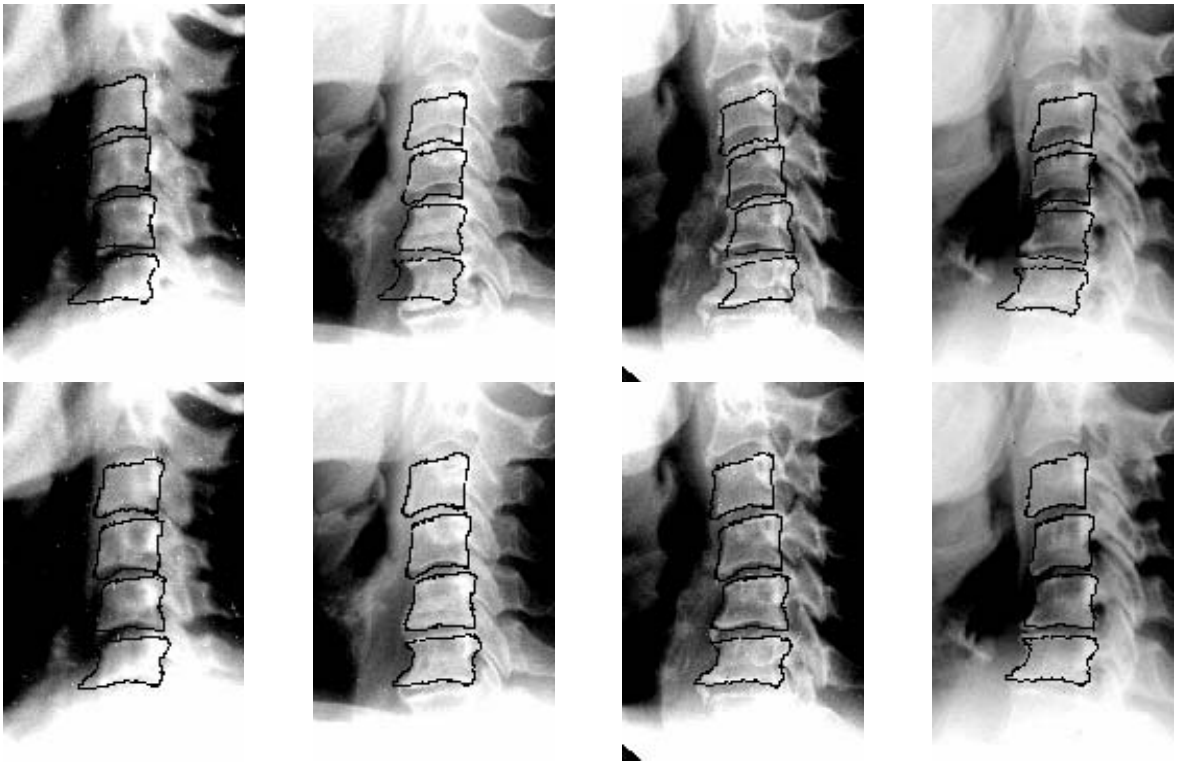


Figure 3. Mean RMS errors and corresponding standard deviations (in pixels) for initial landmark positions (white) and after coarse (light gray) and fine (dark gray) matching steps



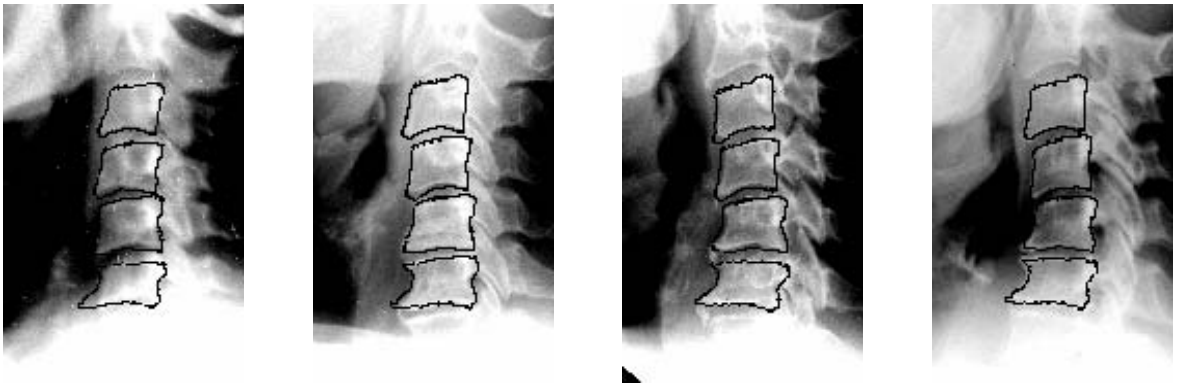


Figure 4. Four cervical spine X-ray images, initial model position in first row, results of coarse and fine matching steps in second and third row

In the leave-one-out test the method was trained on 35 images and then tested on the remaining image. The initialization of the method, which provided global pose parameters of the cervical spine model, was performed by selecting two points, one on vertebra 3 and one on vertebra 6 (Figure 2b). Points were selected in centers of the vertebrae and then perturbed by a constant distance, which was quarter of the vertebra size, in 11 different directions. After applying the method, the resulting landmark positions were compared to the manually defined gold-standard positions by calculating root mean square (RMS) error separately for each of the vertebrae. The RMS errors were calculated for the initial landmark positions and after the coarse and fine matching steps. In 78% of the cases the initial RMS error was reduced. The initial RMS errors and RMS errors after coarse and fine matching steps are shown in Figure 3. The resulting RMS errors were on the average 2.3 pixels, while the initial RMS error ranged on the average from 5.5-7.5 pixels. Figure 4 shows four segmented cervical spine X-ray images. In the first row the model initialisation is presented. Results of coarse and fine matching steps are shown in second and third row, respectively.

5 Conclusions

In this paper we presented a general method for segmenting articulated structures exhibiting variations in shape, appearance and topology. The method is based on statistical parametrical models that are incorporated in a two-level hierarchical scheme. The lower level describes the shape and appearance of individual anatomical structures, while the upper level controls the topology of the articulated structure. When segmenting a given image, the anatomically plausible configuration of the models is searched for in coarse and fine matching steps. The segmentation results on 36 X-ray images confirmed the applicability of the proposed modeling and segmentation strategies. We will focus our future efforts on extensive evaluation of the method on a larger number of spine X-ray images. The proposed hierarchical statistical modeling of shape, appearance, and topology is an important breakthrough for describing non-linear shape variations of articulated structures. Further development and refinement of this methodology should remain an important area of research in the near future.

Acknowledgements

This work was supported by the Ministry of Science and Technology of the Republic of Slovenia under grant J2-0659-1538 and by the IST-1999-12338 project, funded by the European Commission.

References

[1] McInerney T., Terzopoulos D., "Deformable models in medical image analysis: A survey", *Medical Image Analysis*, Vol. 1, No. 2, pp. 91-108, 1996.
[2] Jain A.K., Zhong Y., Dubuisson-Jolly M.P., "Deformable template models: A review", *Signal process*, Vol. 71, pp. 109-129, 1998.
[3] Cootes T.F., Hill A., Taylor C.J., Haslam J., "Use of active shape models for locating structures in medical images", *Image Vision Comput*, Vol. 12, No. 6, pp. 355-365, 1994.
[4] Cootes T.F., Edwards G.J., Taylor C.J., "Active appearance models", *In: Burkhardt, H., Neumann, B. (eds.): European conference on computer vision*, Vol. 2., Springer, pp. 484-498, 1998.
[5] Hill A., Cootes T.F., Taylor C.J., "Active shape models and the shape approximation problem", *Image Vision Comput*, Vol. 14, No. 8, pp. 601-607, 1996.
[6] Smyth P.P., Taylor C.J., Adams J.E., "Automatic measurement of vertebral shape using active shape models", *Image Vision Comput*, Vol. 15, No. 2, pp. 575-581, 1997.
[7] Mahmoodi S., Sharif B.S., Chester E.G., Owen J.P., Lee R., "Skeletal growth estimation using radiographic image processing and analysis", *IEEE T Inf Technol*, Vol. 4, pp. 292-297, 2000.
[8] van Ginneken B., Haar Romeny B.M., "Automatic delineation of ribs in frontal chest radiographs", *In: Hanson, K.M. (ed.): Image processing. Medical Imaging*, Vol. 3979, SPIE San Diego, pp. 825-836, 2000.
[9] Bernard R., Pernuš F., "Statistical approach to anatomical landmark extraction in AP radiographs", *In: Sonka, M., Hanson, K.M. (eds.): Image processing. Medical Imaging*, Vol. 4322, SPIE San Diego, pp. 537-544, 2001.
[10] Heap T., Hogg D., "Improving specifity in PDMs using hierarchical approach", *In: Clark, A.F. (ed): British Machine Vision Conference*, Essex, UK, pp. 80-89, 1997.
[11] Gonzalez R.C., Woods R.E., *Digital image processing*, Addison Wessley, 1992.
[12] Bookstein F.L., "Principal warps: thin-plate splines and the decomposition of deformations", *IEEE T Pattern Anal*, Vol. 11, pp. 567-585, 1989.
[13] Cootes T.F., Page G.J., Jackson C.B., Taylor C.J., "Statistical grey-level models for object location and identification", *Image Vision Comput*, Vol. 14, No. 8, pp. 533-540, 1996.
[14] Long L.R., Pillemer S.R., Lawrence R.C., Goh G-H., Neve L., Thoma G.R., "World Wide Web platform-independent access to biomedical text/image databases", *In: Horii, S.C., Blaine, G. (eds.): PACS design and evaluation: Engineering and clinical issues. Medical Imaging*, Vol. 3339, SPIE San Diego, pp. 52-63, 1998.
[15] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B. P.: *Numerical recipes in C, The art of scientific computing*, University Press, Cambridge, 1992.

Segmentation-based correction of spectral inhomogeneities in color images

Jože Derganc, Boštjan Likar, Franjo Pernuš

Faculty of Electrical Engineering, University of Ljubljana

Tržaška 25, 1000 Ljubljana, Slovenia

tel: +386 1 4768 327, fax : +386 1 4768 279

e-mail: joze.derganc@fe.uni-lj.si

Abstract

A novel color image spectral inhomogeneities correction method, which incorporates nonparametric image segmentation, is presented. Proposed unsupervised segmentation method in 3D RGB color space is based on *max shift* algorithm. It examines cluster membership suitability of feature space points. In this non-parametric way, problems of parametric methods with complicated cluster shapes are avoided. Adverse multipectral inhomogeneity (shading) effects are suppressed by iterative estimation of shading model including segmented corrected color image of the latter iteration. A number of experiments have been performed to test the accuracy, robustness, and speed of the proposed methods. The results indicate that the proposed methods are worth of integrating them into machine vision systems, which are to analyze color scenes.

1 Introduction

Segmentation is a process of partitioning an image into non-intersecting regions in such a way that each region is homogeneous and the union of no two adjacent regions is homogeneous [1,2]. Because color images generally provide more information than gray level images, color image segmentation attracts more and more attention. There are two issues that characterize color image segmentation: (a) the color features used to code the color information and (b) the segmentation method. In color images acquired by electronic imaging devices the color is most often quantitatively specified by *RGB* color features. Normalization of *RGB* features by corresponding intensity leads to *rgb* features, which do not comprise intensity information. Experimentally defined features CIE *XYZ* and *I1I2I3* [7] are derived from *RGB* through linear transformation. Human color perception is mathematically modeled by *IHS*, *Lab*, or *Luv* features [1], each determined by a nonlinear transformation of the *RGB* color system. Generally, segmentation methods can be categorized into the following classes: edge detection approaches, region-based approaches, methods based on physical reflectance models, and statistical methods performed in some color feature space [1,2]. Edge detection cannot segment an image without subsequent higher-level processing or a combination with some other segmentation method [3]. By region growing, splitting, and merging, regions of unique color are extracted recursively [4]. Segmentation can be improved by using physical reflectance models [5] and human perception based interpretation models [6]. Statistical approaches are based on thresholding of color feature histograms [7] or on clustering in some

color feature space [8]. A segmentation algorithm may also combine different methods [9,10]. Some of the above approaches were developed under fuzzy set theory [11]. The statistical segmentation approach is very effective, although the feature space, which can be regarded as a sample drawn from an unknown probability distribution, may have quite a complex structure. This is partially a consequence of the adverse effect of color shading or multispectral inhomogeneities, which manifest as smooth color variations, not present in the original scene. In feature space this is reflected in more dispersed and overlapping clusters. Cluster shapes also depend on the color space. In *RGB* space, the clusters may be quite complex and representing the probability distribution with parametric models, e.g., by a Gaussian mixture [12], may introduce severe segmentation artifacts because parametric models are unable to describe complex clusters. More compact clusters may sometimes be obtained by transforming the *RGB* space to *HSI*, *Lab* or *Luv* color spaces, where the intensity (*I*, *L*) and color (*HS*, *ab*, or *uv*) information are separated. By using only the color features the problem of intensity inhomogeneities may be reduced, but at the cost of unstable color features at small intensity values [13]. Moreover, transformation of the *RGB* space takes time and lack of time is one of the major constraints on automatic visual inspection. Complex cluster shapes, common in *RGB* space, call for a nonparametric clustering-based segmentation approach like kernel estimation [14, 15]. One of the reasons that color image segmentation is a nontrivial task is color shading, which is the adverse affect of illumination spectral inhomogeneities, reflecting itself as smooth color variations, not present in original scene.
In this paper we present a novel segmentation-based iterative method for correcting spectral inhomogeneities, which, when applied, might lead to better segmentation results. Two processes, segmentation, based on fast non-parametric analysis of the color feature space and spectral inhomogeneities correction are iteratively applied, which in the last step gives a spectral inhomogeneities free image and a well segmented image.

2 A novel segmentation algorithm

2.1 Max shift algorithm

Let a color image be represented by vectors $\mathbf{v}(\mathbf{x})=[v_1,v_2,v_3]$, where $\mathbf{x}=[x_1,x_2]$ is the vector in image domain X, containing $N=N_{x1}\times N_{x2}$ pixels. N_{x1} and N_{x2} are the width and height of the image. It is assumed that image $\mathbf{v}(\mathbf{x})$ is digitized with $M_{v1}=M_{v2}=M_{v3}$ levels for each color component. Image pixels with feature vectors $\mathbf{v}(\mathbf{x})$ are accumulated by function $a(\mathbf{y})$ in feature space Y and a 3D histogram is constructed according to iterative assignment:

$$\mathbf{v}(\mathbf{x}\in X)=\mathbf{y} \quad \Rightarrow \quad a(\mathbf{y})=a(\mathbf{y})+1 \quad \forall \mathbf{x}\in X \tag{1}$$

The probability density distribution of colors $p(\mathbf{y})$ (Fig. 1a) is obtained by equation:

$$p(\mathbf{y})=\frac{1}{N_{x1}N_{x2}}\frac{\sum_{\mathbf{y}_i\in\Omega} a(\mathbf{y}_i)}{D_{y1}D_{y2}D_{y3}} \quad \Rightarrow \quad \sum_{\mathbf{y}\in Y} p(\mathbf{y})=1 \tag{2}$$

where the histogram is filtered by a 3D filter Ω with dimensions $D_{v1}=D_{v2}=D_{v3}$ and normalized according to the number of image pixels $N= N_{x1}\times N_{x2}$:

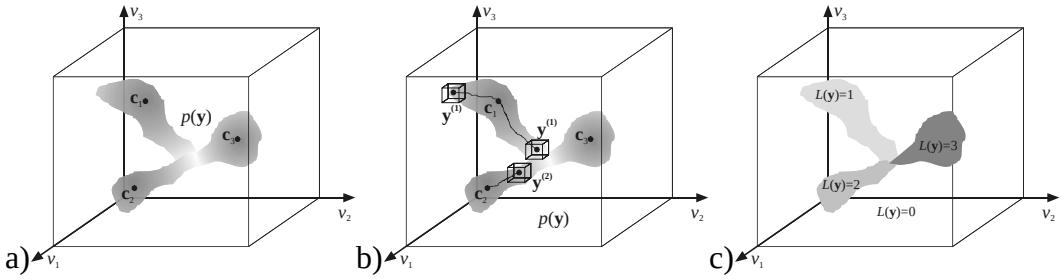


Figure 1: Probability density distribution of color features a), local maximum searching by *max shift* algorithm b), segmented color feature space with labeled clusters c).

Max shift algorithm [15] is used for searching the modes of function $p(\mathbf{y})$ by cube kernel S_y with dimensions along color features $n_{v1}\times n_{v2}\times n_{v3}$:

$$\mathbf{Max} = arg\left[\max_{\mathbf{y}\in S_y}\left(p(\mathbf{y})\right)\right] - \mathbf{y} \tag{3}$$

Position of the kernel center is iteratively moving to the local maximum point of the probability density function $p(\mathbf{y})$. Figure 1b shows, how according to this principle the kernel in position $\mathbf{y}^{(1)}$ converges to the local maximum $\mathbf{c}_1$ of $p(\mathbf{y})$, while the convergence point of the kernel from position $\mathbf{y}^{(2)}$ is in $\mathbf{c}_2$.
Lets define a labeling function $L(\mathbf{y})=i$; $i=1,...,m$ inside the feature space, where each index represents one of the typical image colors. The initial value of this function is:

$$L(\mathbf{y}) = 0 \quad \forall \mathbf{y} \in Y \tag{4}$$

During the clustering process $L(\mathbf{y})$ is set to i according to the convergence of the *max shift* kernel from point $\mathbf{y}$ to convergence point $\mathbf{c}_i$. Iterative convergent shifting process, which starts in point $\mathbf{y}$ and stops in the point of local maximum $\mathbf{c}_i$ is noted:

$$\mathbf{c}_i = MaxShift(\mathbf{y}) \tag{5}$$

Each location $\mathbf{c}_i$ is labeled with $L(\mathbf{c}_i)=i$ (Fig. 1c). The same label is assigned to all points $\mathbf{y}$, from which the kernel converges to point $\mathbf{c}_i$:

$$L(\mathbf{y}) = \mathrm{i} \quad \forall \mathbf{y} \in Y \quad ;\text{if} \quad MaxShift(\mathbf{y}) = \mathbf{c}_i \tag{6}$$

The *max shift* procedure is fundamental for color image segmentation, which is described in the next subchapter.

2.2 Unsupervised color image segmentation

Segmentation process is performed without human intervention. The result is the classification of all image pixels into one of the significant color classes, which represent image. In the image domain the vector function $\mathbf{w}(\mathbf{x})$ representing the segmented image is initiated. After segmentation process, at each point $\mathbf{x}$ this function contains one of the typical

image colors, defined by feature vector $\mathbf{b}_i$. The number of typical image colors is m and initially the m is set to 0. For each feature space point $\mathbf{y}$ with $p(\mathbf{y})>0$ the *max shift* process is performed. Clustering speed is further improved by saving the shifting path of the kernel to a stack and considering already labeled feature points on the shifting path.
In each iteration, the kernel is shifting to the local maximum point, at the same time points $\mathbf{y}_s$ on the path are pushed on the stack. When the first local maximum $\mathbf{c}_1$ is reached, the labeling function gets value $L(\mathbf{c}_1)=1$. Then the stack is sequentially emptied, all saved points $\mathbf{y}_s$ are labeled with $L(\mathbf{y})=1$, and m is increased by 1. Running the *max shift* process on the rest of the feature points, the shifting may stop in a local maximum $\mathbf{y}=\mathbf{c}_i$, which has labeling function value $L(\mathbf{y})=0$ or in an already labeled point $\mathbf{y}$ with label $L(\mathbf{y})>0$. In the first case the convergence point $\mathbf{c}_i$ belongs to a new typical color. Consequently m is increased by 1 and index i is set to m ($\mathbf{c}_i = \mathbf{c}_m$). Labels of this point and saved points $\mathbf{y}_s$ from stack are set to $L(\mathbf{c}_m)=m$; $L(\mathbf{y}_s)=m$, while the stack is sequentially emptied. In the second case the stop point belongs to one of the already obtained m typical colors with label $L(\mathbf{y})=i$. Saved points $\mathbf{y}_s$ from stack are set to $L(\mathbf{y}_s)=i$, while the stack is sequentially emptied. When the clustering process is finished, the center $\mathbf{b}_i$ of each cluster with label i ; $i=1,...m$ is obtained by equation:

$$\mathrm{b}_i = \frac{\sum_{L(\mathrm{y})=i} p(\mathrm{y})\mathrm{y}}{\sum_{L(\mathrm{y})=i} p(\mathrm{y})} \quad i = 1,...,m \tag{7}$$

Final segmented image $\mathbf{w}(\mathbf{x})$ is obtained by mapping image pixel features $\mathbf{v}(\mathbf{x})$ through labeled feature space Y and corresponding cluster centers $\mathbf{b}_i$ (Fig. 2):

$$\mathbf{w}(\mathbf{x}) = \mathbf{b}_{L(\mathbf{v}(\mathbf{x}))} \tag{8}$$

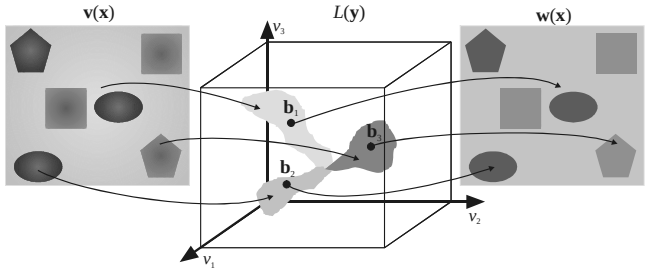


Figure 2: Original image w(x) and image segmented image v(x).

Each image pixel $\mathbf{x}$ belongs to one of the m significant colors $\mathbf{b}_i$; $i=1,...m$. The proposed segmentation method gives optimal boundaries between clusters and optimal repartition of segmented image $\mathbf{w}(\mathbf{x})$ to regions according to the criterion of color classification error. The color of each region is the mean color value $\mathbf{b}_i$ of the corresponding color cluster (Fig. 2).

3 Correction of spectral inhomogeneities

During image acquisition homogenous illumination cannot generally be guaranteed. Grayscale images comprise intensity inhomogeneites, owing to spatial changes of

illumination intensity. In color images, not only the illumination intensity but also the spectrum may vary inside a color scene. Consequently, individual color components comprise different inhomogeneities and it is reasonable to talk about spectral inhomogeneities, which are more distinctive at color scenes with bigger areas. They are induced in color feature space as more dispersed non-compact clusters (Fig. 3). Due to overlaying clusters the determination of the boundary between clusters becomes a problem, and the consequence is bad segmentation (Fig. 3).

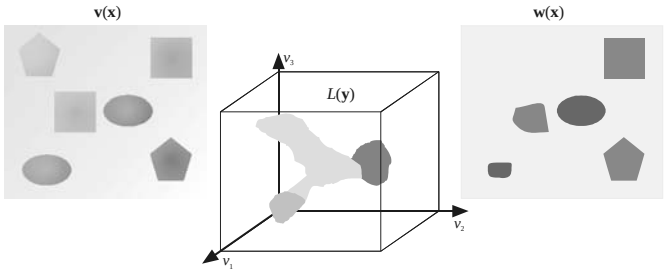


Figure 3: Spectral inhomogeneous image v(x) labeling function *L*(y), and segmented image w(x).

In past several methods have been proposed for correction of intensity inhomogeneities. They may be classified as:

- methods, based on subtraction or division of the image with a previously acquired background image,
- retrospective methods, based just on information content of the acquired image.

The first ones are fast but they cannot correct inhomogeneities which are object dependent. Retrospective methods may reduce shading by homomorfic filtering of an inhomogeneous image or by optimizing a parametric shading model according to a predefined criterion function. This function can be the mean square error between the obtained parametric model [16] and real image background or the entropy. It has been shown that the entropy of a shading free image is smaller than the entropy of an image corrupted by shading [17].
Our approach to spectral inhomogeneity correction is based on the fact that a corrected color image can be better segmented than a shaded color image. Therefore we have incorporated the segmentation process into shading correction.

3.1 Spectral inhomogeneities model

Spectral inhomogeneites are mainly caused by inhomogeneous illumination spectrum and inhomogeneous spectral sensitivity of the camera. Intensity inhomogeneities in grayscale images are usually modeled by a linear image formation model:

$$v(\mathbf{x}) = u(\mathbf{x})S_M(\mathbf{x}) + S_A(\mathbf{x}) \tag{9}$$

which is composed of an additive $S_A(\mathbf{x})$ and multiplicative $S_M(\mathbf{x})$ component. Function $u(\mathbf{x})$ represents the intensity homogeneous image, while function $v(\mathbf{x})$ represents the intensity inhomogeneous image.

Color images are composed of three independent components and therefore we can use the linear image formation model (Eq. 9) individually for each color component *k*:

$$v_k(\mathbf{x}) = u_k(\mathbf{x}) S_{Mk}(\mathbf{x}) + S_{Ak}(\mathbf{x}) \tag{10}$$

where $u_k(\mathbf{x})$ is the *k*-th component of spectral homogeneous image, $v_k(\mathbf{x})$ the *k*-th component of spectral inhomogeneous image, $S_{Ak}(\mathbf{x})$ *k*-th component of additive spectral inhomogeneity, and $S_{Mk}(\mathbf{x})$ *k*-th component of multiplicative spectral inhomogeneity. We define functions $S_{Ak}(\mathbf{x})$ and $S_{Mk}(\mathbf{x})$ as two dimensional second-order polynomials composed of smoothly varying basis functions defined by parameters a_{jk} and m_{jk}:

$$S_{Ak}(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & x_1 x_2 & x_1^2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_{0k} & a_{1k} & a_{2k} & a_{3k} & a_{4k} & a_{5k} \end{bmatrix}^T \tag{11}$$

$$S_{Mk}(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & x_1 x_2 & x_1^2 & x_2^2 \end{bmatrix} \begin{bmatrix} m_{0k} & m_{1k} & m_{2k} & m_{3k} & m_{4k} & m_{5k} \end{bmatrix}^T \tag{12}$$

The correction model is constrained in such a way that it does not change the mean intensity of the input image color components and does not transform the input image to a uniform one. Spectrally homogeneous image $\mathbf{u}(\mathbf{x})$ is obtained from shaded image $\mathbf{v}(\mathbf{x})$ by inverting the shading model for each color component *k* according to the following equation:

$$u_k(\mathbf{x}) = v_k(\mathbf{x}) S_{Mk}^{-1}(\mathbf{x}) - S_{Ak}(\mathbf{x}) S_{Mk}^{-1}(\mathbf{x}) \tag{13}$$

3.2 Retrospective correction of spectral inhomogeneities

A spectral inhomogeneous image $\mathbf{v}(\mathbf{x})$ is segmented into image $\tilde{\mathbf{w}}(\mathbf{x})$, which represents an approximation of the final segmentation $\mathbf{w}(\mathbf{x})$ of the spectral homogeneous image $\mathbf{u}(\mathbf{x})$. The estimation of the correction model is based on the assumption that *k*-th component of homogeneous image $\mathbf{u}(\mathbf{x})$ in Eq. 10 can be replaced with *k*-th component of segmented inhomogeneous image $\tilde{\mathbf{w}}(\mathbf{x})$:

$$v_k(\mathbf{x}) = \tilde{w}_k(\mathbf{x}) \tilde{S}_{Mk}(\mathbf{x}) + \tilde{S}_{Ak}(\mathbf{x}) \tag{14}$$

The model $\{\tilde{\mathbf{S}}_M(\mathbf{x}), \tilde{\mathbf{S}}_A(\mathbf{x})\}$, which is the approximation of the real inhomogeneity correction model, is obtained for each color component *k* by linear regression method from image components $v_k(\mathbf{x})$ and $\tilde{w}_k(\mathbf{x})$. For this purpose the right side of Eq. 14 is rewrote:

$$v_k(\mathbf{x}) = \mathbf{c}_k(\mathbf{x}) \mathbf{p}_k \tag{15}$$

where vector $\mathbf{p}_k$ represents polynomial parameters:

$$\mathbf{p}_k = \begin{bmatrix} m_{0k} & m_{1k} & m_{2k} & m_{3k} & m_{4k} & m_{5k} & a_{0k} & a_{1k} & a_{2k} & a_{3k} & a_{4k} & a_{5k} \end{bmatrix}^T \tag{16}$$

while the basis functions are separated and embeded into a correction vector $\mathbf{c}_k(\mathbf{x})$:

$$\mathbf{c}_k(\mathbf{x}) = \begin{bmatrix} \tilde{w}_k & \tilde{w}_k x_1 & \tilde{w}_k x_2 & \tilde{w}_k x_1 x_2 & \tilde{w}_k x_1^2 & \tilde{w}_k x_2^2 & 1 & x_1 & x_2 & x_1 x_2 & x_1^2 & x_2^2 \end{bmatrix} \tag{17}$$

For *N* image pixels the Eq. 15 is extended to:

$$\mathbf{v}_k = \mathbf{C}_k \mathbf{p}_k \tag{18}$$

where the $\mathbf{v}_k$ is the vector of *N* intensities for k-th image component and $\mathbf{C}_k$ is a matrix of *N* row vectors $\mathbf{c}_k(\mathbf{x})$. Parameters $\mathbf{p}_k$ are determined by linear regression procedure, where mean square error between the model and real data is minimized:

$$\mathbf{p}_k = \left(\mathbf{C}_k^T \mathbf{C}_k\right)^{-1} \mathbf{C}_k^T \mathbf{v}_k \tag{19}$$

Approximative model is employed to obtain the approximation $\tilde{u}_k(\mathbf{x})$ of a spectral homogeneous image $u_k(\mathbf{x})$:

$$\tilde{u}_k(\mathbf{x}) = v_k(\mathbf{x})\tilde{S}_{Mk}^{-1}(\mathbf{x}) - \tilde{S}_{Ak}(\mathbf{x})\tilde{S}_{Mk}^{-1}(\mathbf{x}) \tag{20}$$

After the first iteration image $\tilde{\mathbf{u}}(\mathbf{x})$, which is used as an input to the next iteration, is more homogeneous than image $\mathbf{v}(\mathbf{x})$ (Fig. 4). Consequently the next segmentation result $\tilde{\mathbf{w}}(\mathbf{x})$ is better. The iterative process leads to an optimally segmented image $\mathbf{w}_o(\mathbf{x})$, which is obtained from the optimally corrected input image $\mathbf{u}_o(\mathbf{x})$. In such a way the optimal approximation of the spectral inhomogeneities model $\{\mathbf{S}_{Ao}(\mathbf{x}),\mathbf{S}_{Mo}(\mathbf{x})\}$ is extracted.

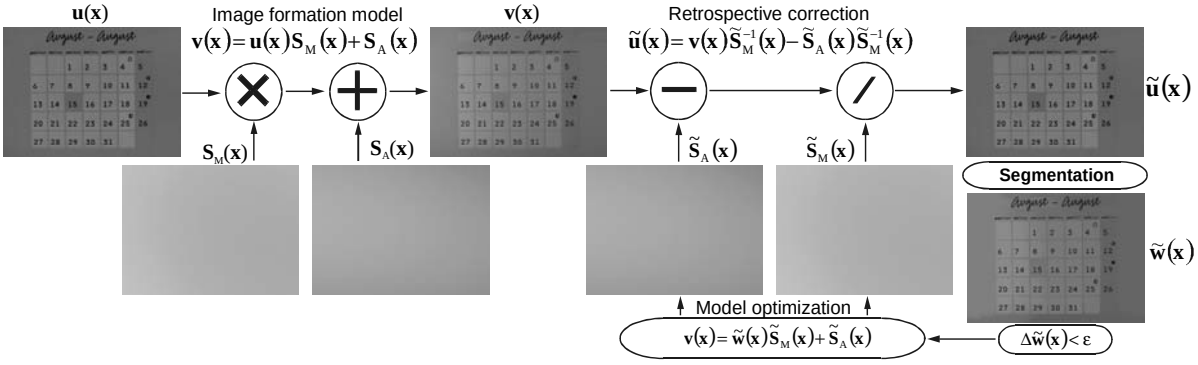


Figure 4: Image formation model (left) and spectral inhomogenities correction (right).

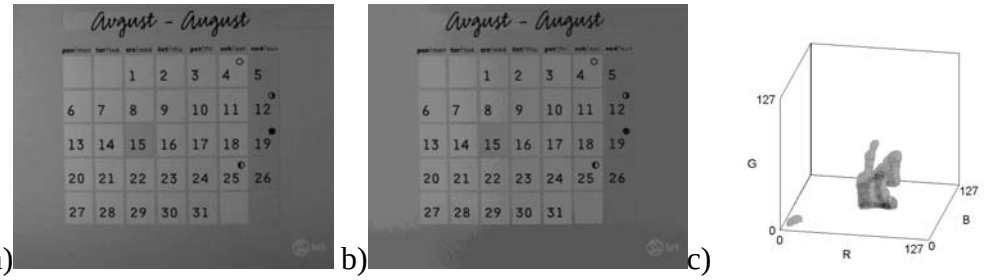
Optimization stops when the difference $\Delta\tilde{\mathbf{w}}(\mathbf{x})$ between consecutive segmented images, say *i* and (*i*-1), is less than ε:

$$\Delta\tilde{\mathbf{w}} = \frac{1}{N}\sum_{j=1}^{N}\left\| {}^{(i)}\tilde{\mathbf{w}}(\mathbf{x}_j) - {}^{(i-1)}\tilde{\mathbf{w}}(\mathbf{x}_j)\right\| < \varepsilon \tag{21}$$

4 Experiments

The proposed method was tested on real spectral inhomogeneous images *Calendar*, *Tablets* and *Brain* captured by a 3CCD color camera. To obtain multispectral inhomogeneities, the *Calendar* scene was illuminated with a red color lamp from the right side (Fig. 5a). The probability density color distribution of image *Calendar* in *RGB* color space shows that color

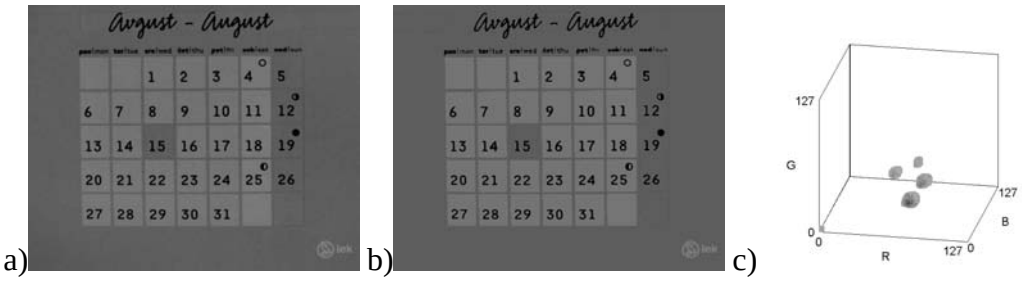
clusters are overlapping (Fig 5c). Segmentation does not correspond to real the image content (Fig. 5b).



a) b) c)

Figure 5: Original spectral inhomogeneous image *Calendar* a), its segmentation b), and distribution of color features in *RGB* color space c).

Suppression of inhomogeneities gives a well separated and compact color clusters (Fig. 6c) and consequently, a good segmentation result (Fig. 6b). Multiplicative components (Fig. 7) and additive components (Fig. 8) are different due to the multispectral nature of illumination inhomogeneities. Spectral inhomogeneities were also successfully reduced in *Tablets* image (Fig. 9). The corrected color image (Fig. 9b) was successively segmented (Fig. 9c). The image *Brain* from [18] in Fig. 10a, which presents a slice of the human brain with spectral inhomogeneities, does not allow an efficient segmentation of brain tissues. Spectral inhomogeneities were suppressed by the proposed algorithm (Fig. 10b) and good segmentation result may be observed in Fig. 10c.



a) b) c)

Figure 6: Corrected spectral homogeneous image *Calendar* a), its segmentation b), and distribution of color features in *RGB* color space c).

a) b) c)

Figure 7: Multiplicative spectral inhomogeneities components of image *Calendar*.

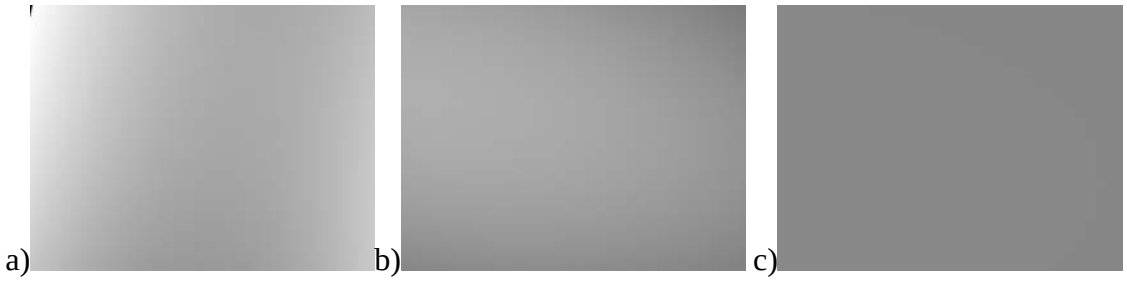


Figure 8: Additive spectral inhomogeneities components of image *Calendar*.

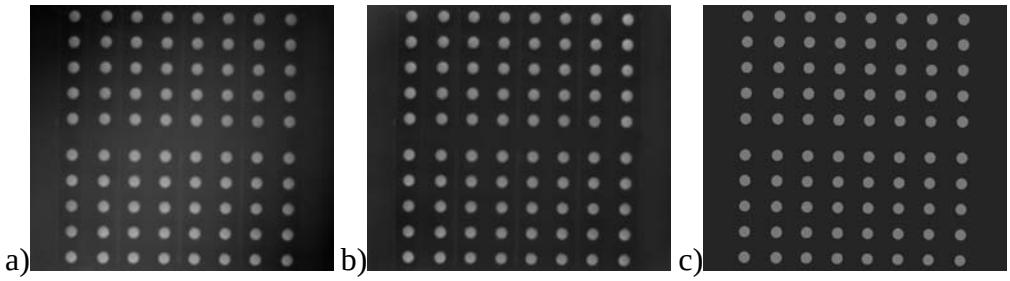


Figure 9: Original *Tablets* image a), corrected image b), and its segmentation c).

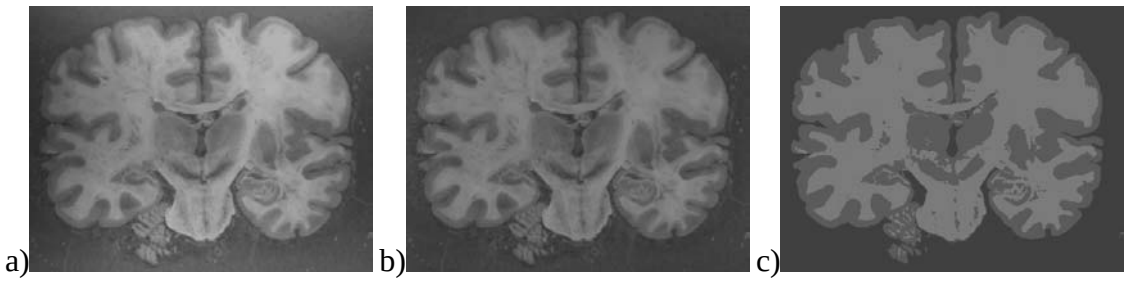


Figure 10: Original *Brain* image a), corrected image b), and its segmentation c).

5 Conclusions

The experimental results show that the proposed automated method efficiently corrects spectral inhomogeneities. Segmentation results of corrected color images are much better than of uncorrected images, enabling correct interpretation of real color scenes. Integration of non-parametric segmentation method into the optimization process proved useful. Multispectral inhomogeneities cannot be suppressed just by transforming the *RGB* features to some other color system, which is a common approach when only intensity inhomogeneities are present. Non-parametric segmentation allows processing in *RGB* color space and there is no need of time consuming transformation between color spaces.

The difference between two consecutive segmented images is reduced during the iterations. After each step the segmented image is closer to the optimal segmentation and gives more

information useful for estimation of the spectral inhomogeneities model at next iteration. Optimization stops when two consecutive segmentations differ by less than a preselected small value. The correction method is robust, repeatable, and fast enough to be used for automated visual inspection in industry. Future work will be focused on the implementation of the method for segmenting multiprotocol 3D MRI medical images.

References

[1] Cheng H.D., Jiang X.H., Sun Y. and Wang J.L., "Color image segmentation: Advances and prospects", *Pattern Recognition*, Vol. 34, pp. 2259-2281, 2001.

[2] Pal N.R. and Pal S.K., "A review on image segmentation techniques", *Pattern recognition*, Vol. 26, pp. 1277-1294, 1993.

[3] Zugaj D. and Lattuati V., "A new approach of color images segmentation based on fusing region and edge segmentation outputs", *Pattern recognition*, Vol. 31, pp. 105-113, 1998.

[4] Tremeau A. and Borel N., "A region growing and merging algorithm to color segmentation", *Pattern Recognition*, Vol. 30, pp. 1191-1203, 1997.

[5] Klinker G.J., Shafer S.A. and Kanade T., "A physical approach to color image understanding", *International Journal of Computer Vision*, Vol. 4, pp. 7-38, 1990.

[6] Maxwell B.A. and Shafer S.A., "Segmentation and interpretation of multicolored objects with highlights", *Computer Vision and Image Understanding*, Vol. 77, pp. 1-24, 2000.

[7] Ohta Y., Kanade T. and Sakai T., "Color information for region segmentation", *Computer Graphics and Image Processing*, Vol. 13, pp. 222-241, 1980.

[8] Park S.H., Yun I.D. and Lee S.U., "Color image segmentation based on 3-D clustering: morphological approach", *Pattern Recognition*, Vol. 31, pp. 1061-1076, 1998.

[9] Schettini R., "A segmentation algorithm for color images", *Pattern recognition letters*, Vol. 14, pp. 499-506, 1993.

[10] Cheng H.D. and Sun Y., "A hierarchical approach to color image segmentation using homogeneity", *IEEE Trans. Image Processing*, Vol. 9, pp. 2071-2082, 2000.

[11] Moghaddamzadeh A. and Bourbakis N., "A fuzzy region growing approach for segmentation of color images", *Pattern recognition*, Vol. 30, pp. 867-881, 1997.

[12] Bergasa L., Duffy N., Lacey G. and Mazo M., "Industrial inspection using Gaussian functions in a colour space", *Image and Vision Computing*, Vol. 18, pp. 951-957, 2000.

[13] Healey G.E., "Segmenting images using normalized color, *IEEE Trans. Systems, Man and Cybernetics*, Vol. 22, pp. 64-73, 1992.

[14] Comaniciu D. and Meer P., "Distribution Free Decomposition of Multivariate Data", *Pattern Analysis and Applications*, Vol. 2, pp. 22-30, 1999.

[15] Derganc J., Likar B., Pernuš F., "Segmentation of shaded color images by non-parametric analysis of feature space". Sixth Computer Vision Winter Workshop, Bled, Slovenia, February 2001, Slovenian Pattern Recognition Society, pp. 13-24, 2001.

[16] Beckers A.L.D, Gelsema E.S., de Bruijn W.C., "An efficient method for calculating the least-squares background fit in electron energy-loss spectrometry", *Journal of Microscopy*, Vol. 171, pp. 87-92, 1993.

[17] Likar B., Maintz J.B.A., Viergever M.A. and Pernuš F., "Retrospective shading correction based on entropy minimization", *Journal of Microscopy*, Vol. 197, pp. 285-295, 2000.

[18] University of California at Los Angeles, Laboratory of Neuro Imaging (LONI), *Human image Dataset, Human Cryotome Data*, Internet address: http://nessus.loni.ucla.edu/data/human/

What space can be reconstructed from multiple catadioptric images

Petr Doubek*
Tomáš Svoboda
ETH Zürich
Gloriastrasse 35, CH-8092 Zürich
tel: +41 1 632 3549, +41 1 632 5769, fax: 41 1 632 1199
e-mail: doubek@vision.ee.ethz.ch, svoboda@vision.ee.ethz.ch

Abstract

The space of the reliable reconstruction from multiple catadioptric images is qualified. This space is shaped by noise in image data and by errors in the camera assembly. The analysis of this shape motivates a simple method for fusing information from multiple views. This fusion widens the space of reliable reconstruction. We demonstrate the usability of the proposed algorithm by experiments with real data.

1 Introduction

The problem of scene reconstruction from two or more perspective projections has been intensively studied in computer vision in last two decades. The results are summarized in the books [4, 5]. The reconstruction from catadioptric images is a relatively new topic. The research that was motivated by mobile robotic application usually relies on the planar motion of the robot. The distance of the detected object (obstacle) is computed only from mutual azimuths, no full 3D reconstruction is estimated. A typical example is [14]. Recently, a very similar approach appeared in [8]. The works where the camera can freely move and where no additional information about the scene is provided is yet relatively rare. An attempt to solve the general reconstruction task is presented in [2]. An experimental study about the sensitivity of the reconstruction with respect to the camera motion is presented. However, a reconstruction method is not proposed. Sturm [9] proposes a method for scene reconstruction from just one catadioptric image by providing coplanarity, perpendicularity and parallelism constraints. The most similar work to ours is [1]. The authors reconstruct points from images taken from arbitrary viewpoints

*The most of the work was done when the first author was with the Center for Machine Perception, CTU Prague.

and fuse information from multiple image pairs. Still, the triangulation method is completely different from ours and no explicit study of the method limitations is presented.

We propose a method for *point reconstruction from multiple catadioptric images*. We impose no constraints on the camera position and the scene structure. The space of reliable reconstruction is qualified. Moreover, it is shown how an inaccurate camera assembly influences the reconstruction results. We suggest a simple method for fusing information from more than two views to overcome the spatial limitation of the reconstruction.

2 Reconstruction

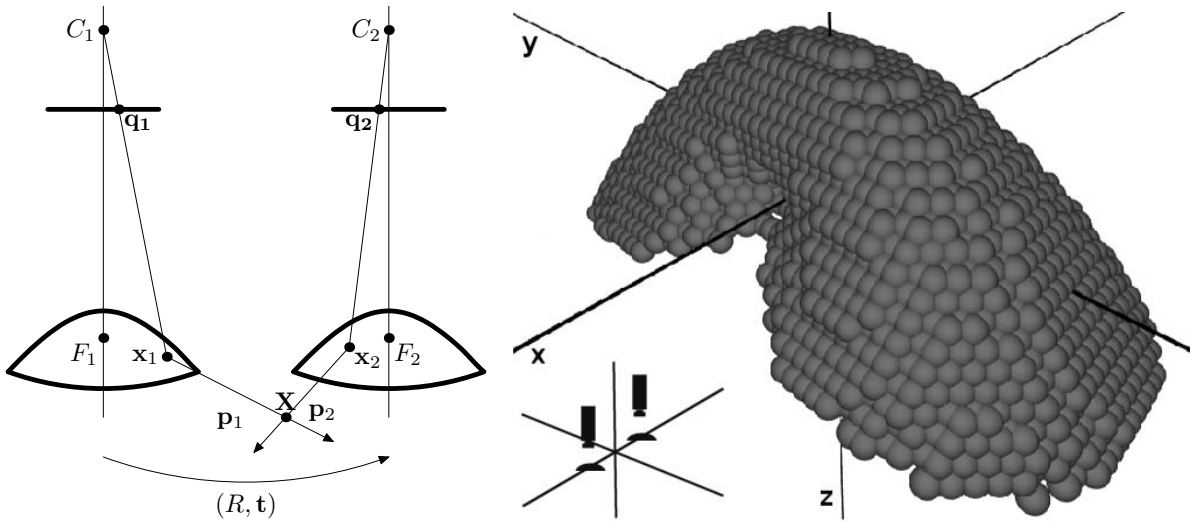
We use central panoramic catadioptric cameras composed of a perspective camera and a hyperbolic mirror or an orthographic camera and a parabolic mirror. For short, we call the cameras hyperbolic or parabolic camera, respectively. A space point $\mathbf{X}$ is reflected by the mirror at point $\mathbf{x}$ and projected to a pixel point $\mathbf{q}$, see Fig. 1(a). If the transformation between world and camera coordinate system is known, the projection is defined by the mirror parameters a, b and the camera calibration matrix K, see [7]. In the following derivation we assume that rotation, R, and translation, $\mathbf{t} \neq \mathbf{0}$, between the cameras are known and the coordinates of the corresponding points are given[1]. The reconstruction is trivial under an unrealistic no-noise assumption. In fact, the rays $\mathbf{p}_1$ and $\mathbf{p}_2$ do not intersect because of noise. Some approximate solution has to be found.

Two methods for reconstruction from a pair of catadioptric images were described in [3]. Firstly, a standard method that finds the reconstructed point as the central point of the shortest transversal, called *mid-point method* [5], and secondly, a modification of the optimal *polynomial* method [6] which has been proposed for perspective cameras. This modified method is called method of epipolar circles and it is based on minimizing the distance in the image plane. The mid-point method proved to be more stable than the method of epipolar circles and it was used in all described experiments exclusively. We do metric reconstruction, hence, it makes sense to directly measure the 3D error what the mid-point method does.

3 The Space of the Reliable Reconstruction

The space of the reliable reconstruction is constrained by noise in the image data, the quality of the perspective camera calibration, and its assembly with the mirror. Considering the image noise, clearly, it makes no sense to perform the reconstruction if the error in the coordinates of a point pair is bigger than the disparity imposed by the camera motion. Concerning the effect of a bad assembly, point projection does not agree with the mathematical model if the camera is not properly assembled.

[1] The camera displacement can be computed from point correspondences by using the method presented in [12]. Correspondences might be found by using the method [10].



(a) (b)

Figure 1: (a) A pair of catadioptric cameras. A space point $\mathbf{X}$ reflects in mirror points $\mathbf{x}_{1,2}$ which project into image points $\mathbf{q}_{1,2}$. Points $F_{1,2}$ denote the foci of the mirrors and $C_{1,2}$ are optical centers of the perspective cameras. (b) The space of the reliable reconstruction for a hyperbolic camera. The motion of the camera is along the x-axis. Camera positions are shown in the lower left part of the image. The distance between cameras is set to 10 mm.

3.1 Errors in Camera Assembly and Calibration

The parabolic camera brings less problems than the hyperbolic one. There is no need to put the camera in some predefined distance from the mirror. With a good telecentric lens and a proper mirror holder we can expect that the imaging model very probably agrees with the reality. The assembly of a hyperbolic mirror with a perspective camera is more difficult. Assuming a good mirror holder we can expect that the coincidence of the mirror and the camera axis will be satisfied. The main issue is then to put the perspective camera in the correct distance from the mirror. To be more precise, the camera center has to lie in the opposite focal point of the mirror[2]. If the camera is not positioned exactly rays $(\mathbf{xX})$ reflected from the mirror do not intersect in one point and their directions are different than computed by using the mathematical model. See the result of simulated experiments.

3.2 Image Noise, Influence of Space Point Position

We suppose an unbiased Gaussian noise model for image point coordinates $\mathbf{q}$. However, the reconstruction is computed from mirror points $\mathbf{x}$, and noise in $\mathbf{x}$ is not exactly Gaussian, more-

[2]Remind that the hyperbolic mirror is one part of the hyperboloid of two sheets.

over, it is biased. Note that image points are projected on a curved surface; the noise character and the mean value, therefore, shift. However, our calculation showed that for small noise, say less than 10 pixels, the bias and non-Gaussian effect can be neglected.

We assume in the rest of this section that we have two images from two different camera positions and no rotation in-between ($R = I_3$). This is to simplify the simulation, it is not a constraint. The length and the direction of the motion are the crucial parameters that influence the size of the space for reasonable reconstruction. Assuming a certain level of noise, some points cannot be reconstructed because of insufficient disparity. Essentially, it is impossible to reconstruct points on the line F_1F_2 because of their zero disparity. The triangulation is in fact the problem of finding the third apex of a triangle given the baseline and two angles. A small angle rays means that a distance between the baseline and the apex is large compared to the baseline length. Further, it means that a small error in the ray angle results in a big error in the apex position.

Going back from the angles to the disparity we can say that a space point can be successfully reconstructed only if the disparity of its correspondences is considerably higher than the image noise level. To qualify the space of the reasonable reconstruction we place the cameras in two positions $F_{1,2} = [\pm\frac{u}{2}, 0, 0]^T$ and calculate the disparity d_{ijk} of the points

$$\mathbf{X}_{ijk} = [iu, ju, ku]^T \ , \text{ where } i, j, k \in \{-n \dots n\} \ .$$

The value of the minimal disparity, l, depends on a noise level estimate and our requirements for the reconstruction precision. The space of reliable reconstruction is composed of points $\mathbf{X}_{ijk}$ for which $d_{ijk} > l$. No discretization noise is considered. A graphical representation of this space is shown in Fig. 1(b). The simulated hyperbolic camera has parameters $a = 28.1\,\text{mm}$, $b = 23.41\,\text{mm}$, $r_{rim} = 30\,\text{mm}$ (mirror radius), resolution 1024×1024 and

$$K = \begin{pmatrix} 1248.383 & 0 & 511.5 \\ 0 & 1248.383 & 511.5 \\ 0 & 0 & 1 \end{pmatrix} \ .$$

We show only the results for a hyperbolic camera since the results for a parabolic camera were the same except for a difference caused by a slightly different field of view. The bottom of the space is given by the field of view which is determined by the mirror parameters. The space is most elongated along the y axis which is perpendicular to the baseline lying on the x axis. Moreover, a cone with axis x and apex at zero is carved from the space. Space points lying close to the line F_1, F_2 induce very small disparity. The small space cones occluded by the bodies of the cameras are not modeled here. Finally, we can notice that the lower z coordinate the wider the space is. It is caused by the mirror curvature. Camera displacement induces bigger disparity for points that are projected closer to the mirror periphery. An approximate space size for different threshold values is shown in Table 1. The baseline length is 10 mm. Borders of the space in axes directions and approximate space volumes were measured.

3.3 Triangulation from More then Two Images

The shape of the reliable reconstruction space motivated us to use more than two images for the reconstruction. We propose a method to widen the space of reliable reconstruction into an

Table 1: Approximate size of the space of reliable reconstruction.

l [pixel]	x [m]		y [m]		z [m]		$V[\mathrm{m}^3]$
1.00	-2.20	2.20	-4.40	4.40	-1.20	1.60	31.46
2.00	-1.08	1.08	-1.98	1.98	-0.54	0.72	3.92
3.00	-0.64	0.64	-1.28	1.28	-0.32	0.48	1.16
4.00	-0.42	0.42	-0.98	0.98	-0.28	0.28	0.46
6.00	-0.30	0.30	-0.70	0.70	-0.20	0.20	0.15
8.00	-0.24	0.24	-0.48	0.48	-0.12	0.18	0.06
10.00	-0.22	0.22	-0.44	0.44	-0.12	0.16	0.03

union of particular reliable spaces for each image pair.

Suppose we have more then two corresponding projections $\mathbf{q}_i$, i is the camera index, of a space point $\mathbf{X}$ in multiple images. All normalized mirror points $\mathbf{x}_i$ can be expressed in a common coordinate system, since the transformations between viewpoints are known. The point pair $\mathbf{x}_i^*, \mathbf{x}_j^*$ with the angular difference closest to the right angle is preferred

$$(\mathbf{x}_i^*, \mathbf{x}_j^*) = \arg\min_{\{i,j\}} \left| \frac{\pi}{2} - \arccos \|\mathbf{x}_i \,.\, \mathbf{x}_j\| \right| \quad . \tag{1}$$

4 Experiments

First, we conducted experiments with synthetic data. We wanted to compare the influence of factors that limit the reconstruction. Parameters of the simulated camera are listed in Section 3.2.

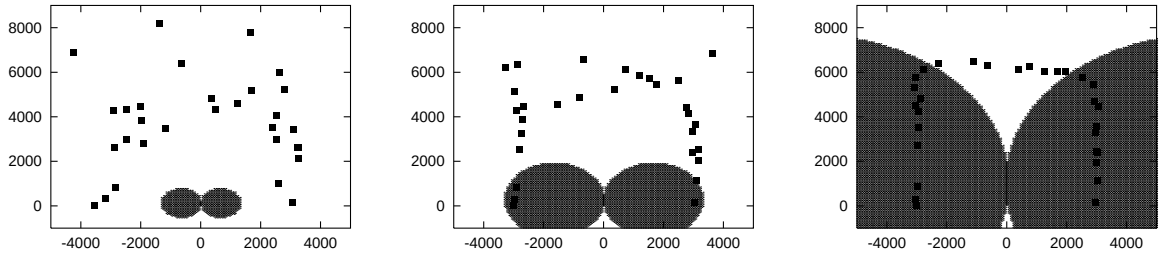
An unbiased Gaussian noise was added to 30 correspondences $\mathbf{q_{1,2}}$ at each of 1000 repetitions. The precision of the reconstruction was described by two different error measures. The 3D error is the distance between the original and the reconstructed space point. The 2D error is the sum of distances between exact images of the original space point and images of the reconstructed space point. Experiments were conducted with both the parabolic and the hyperbolic camera. We show the results for the hyperbolic camera exclusively, since no significant differences between the parabolic and the hyperbolic camera were observed.

4.1 Reconstruction of Surrounding Scene, Baseline Length

The strength of a catadioptric camera lies in the reconstruction of an encompassing scene. A whole room can be, theoretically, reconstructed from just two catadioptric images.

The size of our test room is $6000 \times 6000 \times 3000$ mm. We place test points randomly on three walls, ten on each, and set the image noise level to 1 pixel. The hyperbolic camera moves from the center of one wall to the center of the opposite wall and captures images at positions $0, 200, 500, 2000$ mm from the starting point. Three scene reconstructions are made – the first image and one of the remaining images are used for every reconstruction. The projection of the

reconstructed scene into the xy plane (Fig. 2) is significant. The shape of the reliable reconstruction, shown in Fig. 2, was generated for the minimal disparity $l = 50$ pixels and for points with $z = 0$ while the reconstructed points have $z \in \langle -200, 2800 \rangle$. Such a high value of the disparity threshold was used to make the shape of the space of the reliable reconstruction well visible.



(a) 200 mm (b) 500 mm (c) 2000 mm

Figure 2: Room reconstruction for different baseline lengths. Big points are the reconstructed points, lines denote the respective baselines and small dots show the space of reliable reconstruction.

We can see that the precision is significantly improving when the length of the baseline increases. The influence of the mutual position of a reconstructed point and the baseline is obvious, too.

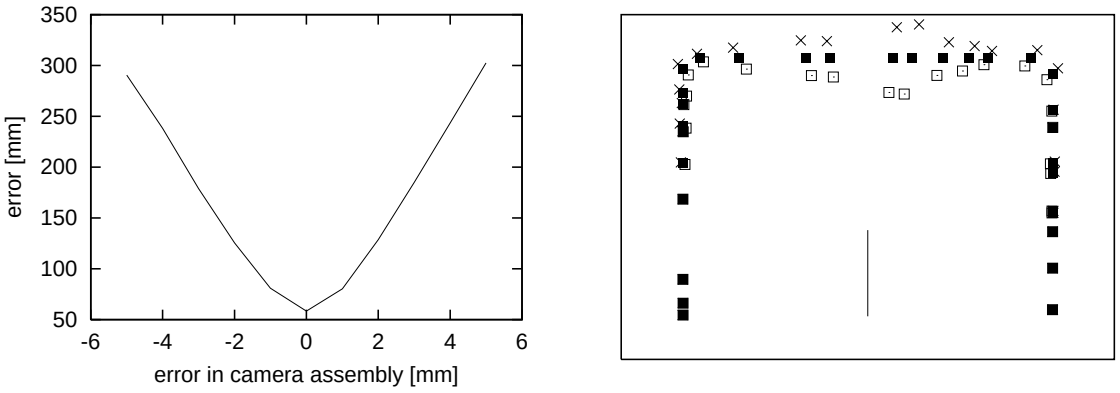
4.2 Error in Camera Assembly

An error in the distance between the hyperbolic mirror center and the camera center is the most probable and the most significant error in the camera calibration and assembly. In the following experiment, we simulate this type of error in the range of -5 to 5 mm and we suppose that the axes of the mirror and the camera coincide. The correct distance $|FF'|$ is 73.15 mm. The scene is the same as in the previous experiment, the baseline length is 1000 mm. One pixel image noise is added to get closer to real conditions.

The 3D error caused by the assembly error is shown in Fig. 3(a). The value of the 3D error at zero tells us that a part of a measured error caused by the image noise is small compared to a part caused by a bad camera assembly. Fig. 3(b) reminds us again of the reliable reconstruction space – points lying in the baseline direction are more influenced by a bad camera assembly.

4.3 Real Data

Seven images of the CMP Lab were captured at different positions, see Fig. 5. The size of the room is the same as in the simulated experiments – $6000 \times 6000 \times 3000$ mm. The camera



(a) 3D error median

(b) -5,0,5 mm error

Figure 3: Error caused by a bad camera assembly.

parameters are different, $a = 28.19$ mm, $b = 9.4$ mm, $r_{rim} = 20$ mm, image resolution 768×512 and

$$K = \begin{pmatrix} 1089.6155 & 0 & 370 \\ 0 & 1106.997 & 267 \\ 0 & 0 & 1 \end{pmatrix} .$$

We selected several objects which are recognizable in at least two images and determined manually correspondences of their corner points. The Rec3D software [13] was used for making correspondences and creating VRML models of the reconstructed scene. The scene reconstructed from all seven images is sufficient for a basic orientation in the real scene. The objects A,B,C,F,G were reconstructed quite successfully. In contrary, objects which were recognizable only in two or three neighboring images and which lied close to the baseline direction (D,E) were reconstructed unsatisfactorily.

We tried a reconstruction from a smaller number of images. No decrease of a quality was observed when only four but carefully chosen images (1,3,4,7) were used. Serious misplacements and deformations of objects appeared for reconstructions from three or two images, see Fig. 6. The real camera used for experiments has a lower resolution, 768×512 compared to 1024×1024 for the simulated camera, which means that the radius of the mirror in pixels is halved. It has a higher a/b ratio, $28.19/9.40$ compared to $28.10/23.41$ used in simulating, which results in a wider field of view but also in a lower spatial resolution.

5 Conclusion

We qualified the shape and the size of the reliable reconstruction space regardless of the used reconstruction method. We showed how the wrong assembly of the catadioptric camera deteriorates the reconstruction. The analysis of the reliable reconstruction motivated us to suggest a

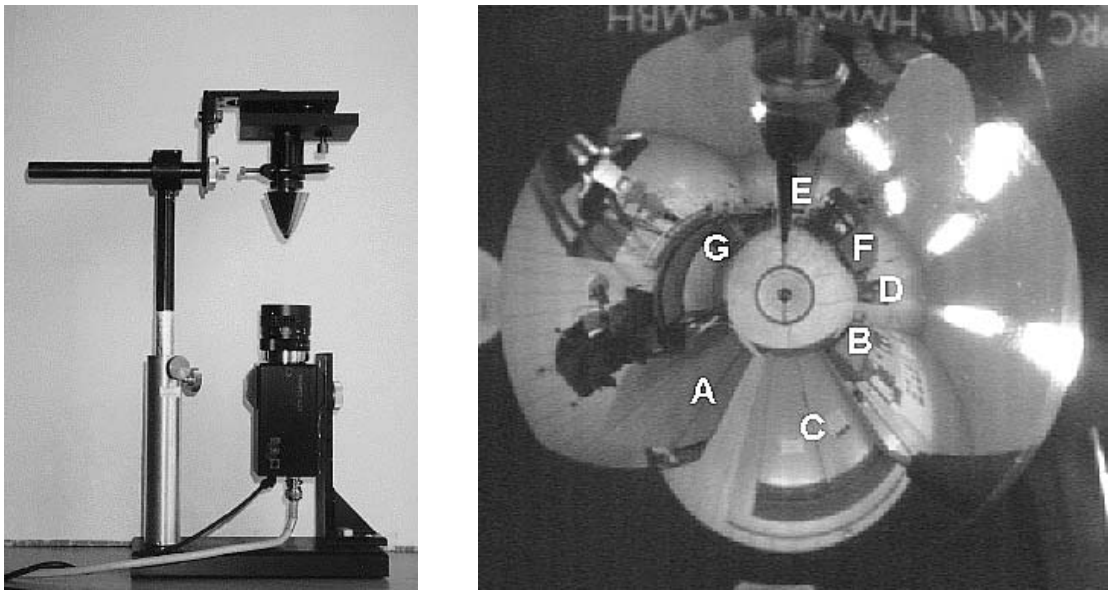


Figure 4: Left: The hyperbolic camera. Right: Hyperbolic image from the first camera position. The image resolution is around 450×450 pixels.

method for fusing information from multiple image pairs.

The experiments with the real camera proved the applicability of the proposed algorithm for scene reconstruction.

Acknowledgments

This research was supported by the Grant Agency of the Czech Republic under the grant GACR 102/01/0971 Omnidirectional Vision.

References

[1] Roland Bunschoten and Ben Kröse. 3D scene reconstruction from cylindrical panoramic images. In *9th International Symposium on Intelligent Robotic Systems*, pages 199–205, 2001. http://carol.wins.uva.nl/˜bunschot/.

[2] Peng Chang and Martial Hebert. Omni-directional structure from motion. In Kostas Daniilidis, editor, *IEEE Workshop on Omnidirectional Vision*, pages 127–133. IEEE Computer Society Press, June 2000.

[3] Petr Doubek. Všesměrové vidění. Master's thesis, Mathematics and Physics Faculty, Charles University in Prague, Ke Karlovu 3, Prague, Czech Republic, September 2001. In Czech.

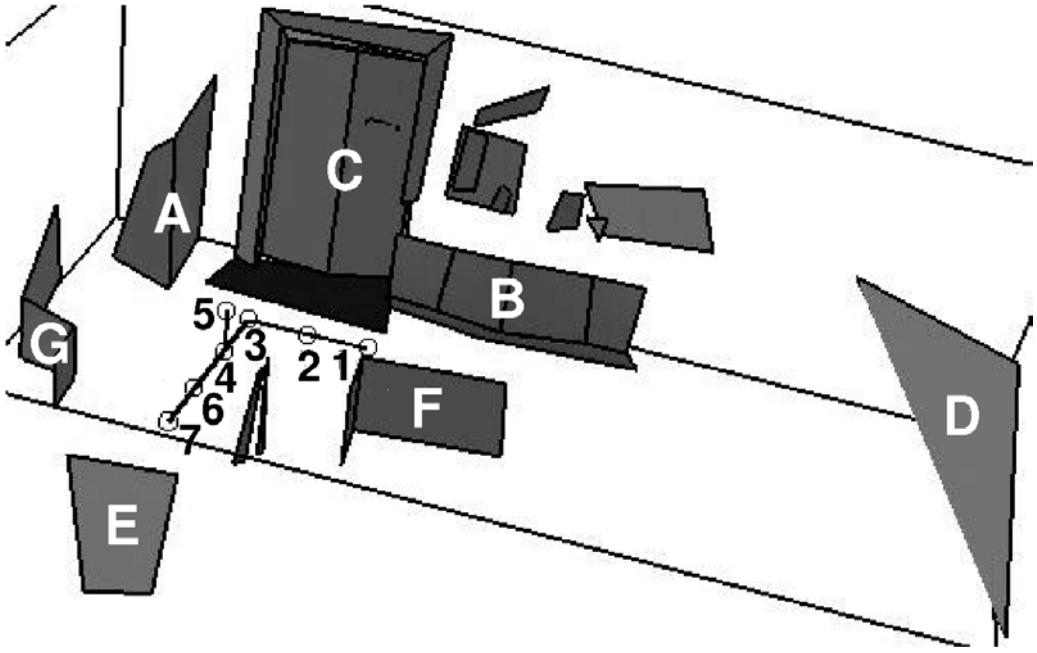


Figure 5: Reconstruction from all 7 images. Camera positions are marked by numbers, the distance between neighbor positions is 600 mm.

[4] Olivier Faugeras, Quang Tuan Luong, and Théo Papadopoulo. *The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, Cambridge, Massachusetts, 2001.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[6] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[7] Tomáš Pajdla, Tomáš Svoboda, and Václav Hlaváč. Epipolar geometry of central panoramic cameras. In Ryad Benosman and Sing Bing Kang, editors, *Panoramic Vision : Sensors, Theory, and Applications*, pages 85–114. Springer Verlag, Berlin, Germany, 1 edition, 2001.

[8] T. Sogo, H. Ishiguro, and M.M. Trivedi. N-ocular stereo for real-time human tracking. In Ryad Benosman and Sing Bing Kang, editors, *Panoramic Vision, Sensors, Theory and Applications*, Monographs in Computer Science, pages 359–375. Springer-Verlag, New York, USA, 1 edition, 2001.

[9] Peter Sturm. A method for 3D reconstruction of piecewise planar objects from single panoramic images. In *International Conference on Computer Vision and Pattern Recognition CVPR*, pages 119–126, June 2000.

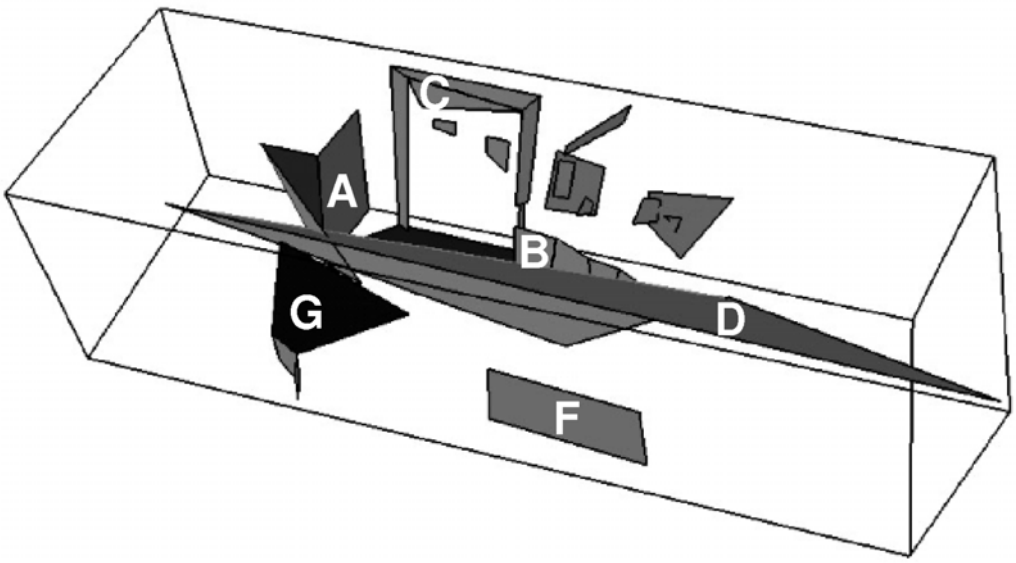


Figure 6: A scene reconstructed using images 1 and 2.

[10] Tomáš Svoboda and Tomáš Pajdla. Matching in catadioptric images with appropriate windows and outliers removal. In Wladyslav Skarbek, editor, *Proc. of the 9th International Conference on Computer Analysis of Images and Patterns*, number 2124 in Lecture Notes in Computer Science, pages 733–740, Berlin, Germany, September 2001. Springer.

[11] Tomáš Svoboda, Tomáš Pajdla, and Václav Hlaváč. Epipolar geometry for panoramic cameras. In Hans Burkhardt and Neumann Bernd, editors, *the fifth European Conference on Computer Vision, Freiburg, Germany*, number 1406 in Lecture Notes in Computer Science, pages 218–232, Berlin, Germany, June 1998. Springer.

[12] Tomáš Svoboda, Tomáš Pajdla, and Václav Hlaváč. Motion estimation using central panoramic cameras. In Stefan Hahn, editor, *IEEE International Conference on Intelligent Vehicles*, pages 335–340, Stuttgart, Germany, October 1998. Causal Productions.

[13] Tomáš Werner, Tomáš Pajdla, and Martin Urban. REC3D: Toolbox for 3D Reconstruction from Uncalibrated 2D Views. Technical Report CTU-CMP-1999-4, Czech Technical University, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic, December 1999.

[14] Kazumasa Yamazawa, Yasushi Yagi, and Masahiko Yachida. Obstacle detection with omnidirectional image sensor hyperomni vision. In *IEEE International Conference on Robotics and Automation 1995*, pages 1062–1067, 1995.

Defining regions within the Combinatorial Pyramid framework

Luc Brun† and Walter Kropatsch‡*

† Laboratoire d'Études et de Recherche en Informatique(EA 2618)

Université de Reims - France

and

‡ Institute for Computer-aided Automation

Pattern Recognition and Image Processing Group

Vienna Univ. of Technology- Austria

Abstract

Irregular Pyramids are defined as a stack of successively reduced graphs. Each vertex of a reduced graph is associated to a set of vertices in the base level graph named its receptive field. If the initial graph is deduced from a planar sampling grid its reduced versions are planar and each receptive field is a region of the initial grid. Combinatorial Pyramids are defined as a stack of successively reduced combinatorial maps. Combinatorial maps are based on half edges named darts and the receptive field of a dart is a sequence of darts in the base level combinatorial map. We present in this paper preliminary results showing how to define regions from the receptive fields of the darts.

1 Introduction

A *Region* is defined as a connected set of pixels. The regions defined by segmentation algorithms fulfill some homogeneity criterion and usually encode either the projections of the different objects of a scene or the main parts of some of these objects. Regions are lot more informative than pixels and a wide variety of internal properties such that the shape, the texture or the set of colors may be extracted from them. External properties such as the adjacency or the inclusion relationships between regions also provide meaningful information about a scene.

Image partitions into region may be defined in parallel using hierarchical data structures. These data structures encode additionally the levels of details of a partition. For example, using such data structures, the hierarchical relation between one region encoding a face and the

*This Work was supported by the Austrian Science Foundation under P14445-MAT.

regions encoding the different parts of this face (e.g. the eyes and the hears) may be encoded explicitly. The *Regular image pyramids* is a hierarchical data structure introduced in 1981/82 [4] as a stack of images with exponentially reduced resolution. Using the neighborhood relationships defined on each image the *Reduction window* relates each pixel of the pyramid with a set of pixels defined in the level below. The pixels belonging to one reduction window are the *children* of the pixel which defines it. This father-child relationship may be extended by transitivity down to the base level image. The set of children of one pixel in the base level is named its *receptive field* (RF) and defines the embedding of this pixel on the original image.

Using regular pyramid the receptive fields are not necessarily connected [1] and may thus contradict the usual definition of regions. This drawback may be overcome by using irregular pyramids defined as a stack of successively reduced graphs. The *Simple graph* and *Dual graph Pyramids* respectively introduced by Meer [6] and Kropatsch [5] define each level of the pyramid by selecting a set of vertices named surviving vertices and mapping all non surviving vertices to surviving ones. The father-child relationship induced by this mapping defines the reduction window of each surviving vertex. The transitive closure of this relation defines, as in regular pyramids, the receptive field of a surviving vertex. Using such a reduction scheme, if the initial graph is defined from a planar sampling grid all the reduced versions of the grid are planar. Moreover, each initial vertex may be associated to one pixel and the receptive field of a surviving vertex is defined as a connected set of pixels.

Combinatorial pyramids are defined as a stack of combinatorial maps successively reduced by contraction and removal operations. Combinatorial Pyramids are equivalent to dual graph pyramids with the exception that they represent the orientation explicitly. The expected advantages of such hierarchies within the image analysis framework are presented in [3]. Combinatorial maps are based on darts. Hence the the reduction window and the receptive fields of Combinatorial Pyramids are expressed in terms of darts. However, using either simple graph or dual graph pyramids the basic entity is the vertex/pixel. Therefore, the receptive fields may be interpreted as regions of an initial image. We present in this paper preliminary results showing how to define regions within the combinatorial pyramid framework.

2 Combinatorial maps

A combinatorial map may be seen as a planar graph encoding explicitly the orientation of edges around a given vertex. Figure 1(a) demonstrates the derivation of a combinatorial map from a plane graph. First edges are split into two half edges called *darts*, each dart having its origin at the vertex it is attached to. The fact that two half-edges (darts) stem from the same edge is recorded in the reverse permutation α. A second permutation σ encodes the set of darts encountered when turning counterclockwise around a vertex.

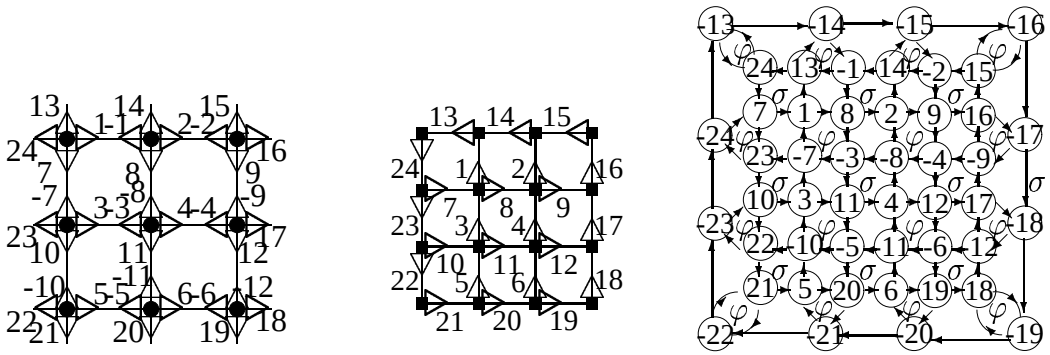
A combinatorial map is thus defined as a triplet $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, where $\boldsymbol{\mathcal{D}}$ is the set of darts and σ, α are two permutations defined on $\boldsymbol{\mathcal{D}}$ such that α is an involution:

$$\forall d \in \boldsymbol{\mathcal{D}} \quad \alpha^2(d) = d \tag{1}$$

Note that, if the darts are encoded by positive and negative integers, the involution α may be implicitly encoded by the sign (Figure 1(a)).

The symbols $\alpha^*(d)$ and $\sigma^*(d)$ stand, respectively, for the α and σ orbits of the dart d. More generally, if d is a dart and π a permutation we will denote the π-orbit of d by $\pi^*(d)$.

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, its dual is defined by $\overline{G} = (\boldsymbol{\mathcal{D}}, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$. The orbits of the permutation φ encode the set of darts encountered when turning around a face. Note that, using a counter-clockwise orientation for permutation σ, each dart of a φ-orbit has its associated face on its right (see e.g. the φ-orbit $\varphi^*(1) = (1, 8, -3, -7)$ in Figure 1(a)).



(a) G (b) $\overline{G}$ (c) $\mathcal{OG}$

Figure 1: A 3×3 grid encoded by a combinatorial map

Figure 1 illustrates the encoding of a 3×3 4-connected discrete grid by a combinatorial map. Each vertex of the initial combinatorial map (Figure 1(a)) encodes a pixel of the grid. The σ-orbit of one vertex encodes its adjacency relationships with neighboring vertices (see e.g. the σ-orbit $(-8, -3, 11, 4)$ encoding the central vertex). The α successors of the darts 13 to 24 are not represented in Figure 1(a) in order to not overload it. These darts encode the adjacency relationships between the external pixels of the grid and its background. The σ orbit of the background vertex is equal to the sequence of darts form -13 to -24 : $(-13, -14, \ldots, -23, -24)$. The dual combinatorial map is represented in Figure 1(b). We also did not represent the α-successor of the positive darts on this Figure to not overload it. Each vertex of this dual map may be associated to a corner of a pixel. Moreover, each of its dart may be understood as an oriented crack, i.e. as a side of a pixel with an orientation. For example, the dart 1 in Figure 1(b) encodes the left side of the upper-left pixel oriented from bottom to top. The dart -1 encodes the same crack with an orientation from top to bottom. Using the above interpretation of darts, the σ-orbit of each pixel defines the sequence of cracks which surrounds it. For example, the upper left pixel is encoded in Figure 1(b) by the σ-orbit $(1, 13, 24, 7)$. In the same way, the pixel located on the first line, second column, is encoded by the σ-orbit $(2, 14, -1, 8)$. The fact that these two pixels share a same crack with a different orientation is recorded by the darts 1 and -1 which belong to a same edge.

Figures 1(c) illustrates the $\sigma - \varphi$ representation of a combinatorial map. Within this alternative representation, a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ is represented by an oriented planar graph $\mathcal{OG} = (V, E)$. The set V of vertices of $\mathcal{OG}$ is equal to the set of darts $\boldsymbol{\mathcal{D}}$ and an ori-

ented edge $e \in E$ connects two vertices d_1 and d_2 iff either $d_2 = \sigma(d_1)$ or $d_2 = \varphi(d_1)$. Using this representation, the σ and φ orbits of the combinatorial map are represented by the faces of the oriented graph $\mathcal{OG}$. Note that each vertex of $\mathcal{OG}$ has two incoming arcs (its σ and φ predecessors) and two outgoing ones (its σ and φ successors).

3 Combinatorial Pyramids

The aim of combinatorial pyramids is to combine the advantages of combinatorial maps with the reduction scheme defined by Kropatsch [5] (Section 1). A combinatorial pyramid is thus defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations.

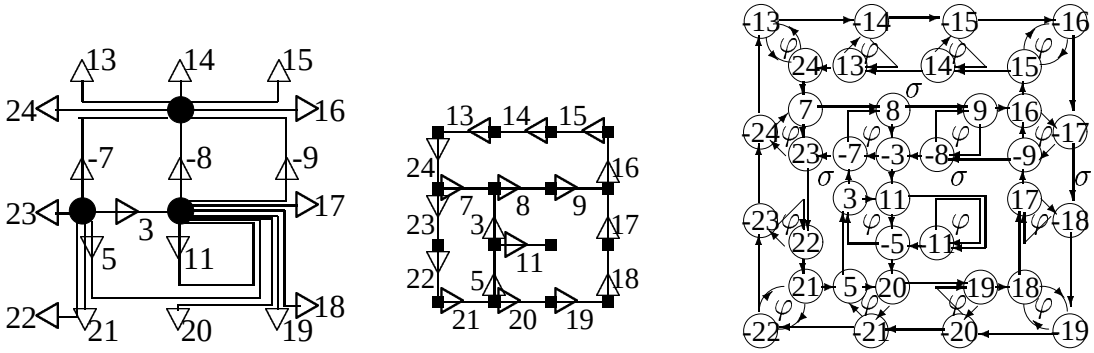
In order to preserve the number of connected components of the initial combinatorial map, the contraction of self-loops must be avoided. This last requirement may be satisfied if the set of edges to be contracted forms a forest of the initial combinatorial map. A forest is defined as a set of non connected trees. Within the combinatorial map framework, a tree may be characterized as a combinatorial map with only one φ-orbit (i.e. only one face). A more formal definition may be found in [2][Def. 4]. A set of edges to be contracted satisfying the above requirement is called a contraction kernel:

Definition 1 Contraction Kernel

Given a connected combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ the set $K \subset \boldsymbol{\mathcal{D}}$ will be called a contraction kernel iff K is a forest of G.

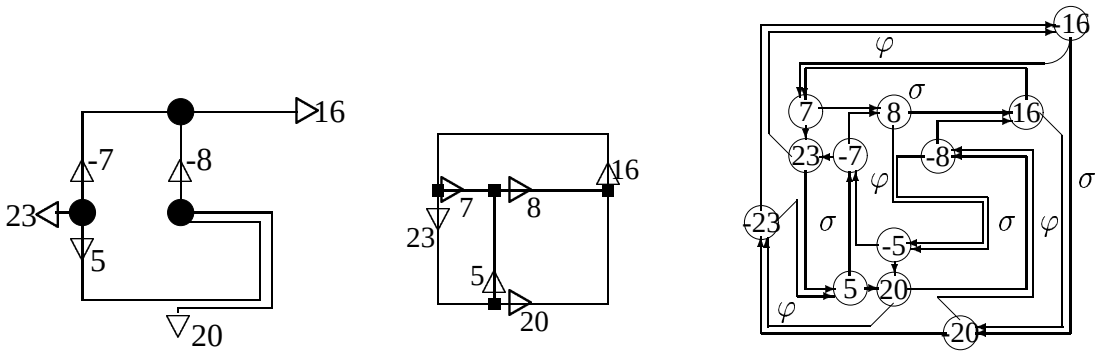
The set $\mathcal{SD} = \boldsymbol{\mathcal{D}} - K$ is called the set of surviving darts.

Given a contraction kernel K, we denote by $\mathcal{CC}(K)$ its set of connected components. Since K is a forest, each $\mathcal{T} \in \mathcal{CC}(K)$ is a tree. Intuitively, a tree $\mathcal{T} \in \mathcal{CC}(K)$ collapses in one vertex a connected set of vertices of the initial combinatorial map. Since each initial vertex is associated to one pixel, the contracted vertex encodes a region.



(a) $G_1 = G/K_1$ (b) $\overline{G_1} = \overline{G/K_1}$ (c) $\mathcal{OG}_1$

Figure 2: Reduction of the initial grid displayed in Figure 1 by the contraction kernel K_1



(a) $(G/K_1) \setminus K_2$ (b) $\overline{(G/K_1) \setminus K_2}$ (c) $\mathcal{OG}_2$

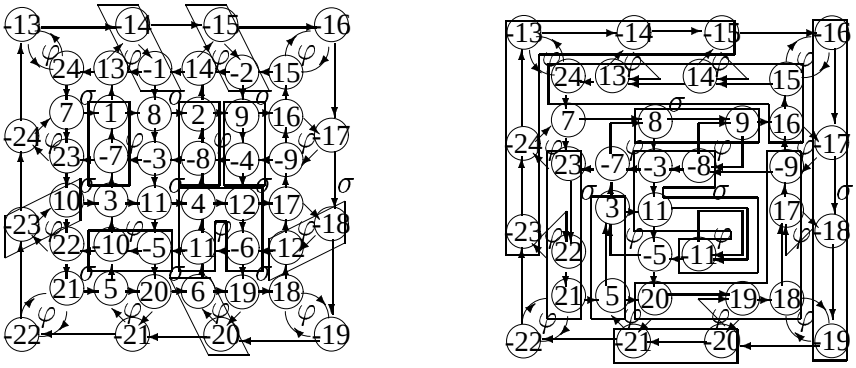
Figure 3: Reduction of the contracted combinatorial map displayed in Figure 2 by the removal kernel K_2

Figure 2 illustrates a contraction of the initial combinatorial map represented in Figure 1 by a contraction kernel K_1 defined by the trees $\alpha^*(1,2)$, $\alpha^*(4,12,6)$ and $\alpha^*(10)$. Since each initial vertex is incident to a contracted edge this forest spans the initial combinatorial map and we obtain 3 surviving vertices encoding 3 regions.

One can note on Figure 2 that many edges encode redundant boundaries. For example the edges $\alpha^*(8)$ and $\alpha^*(9)$ encode a same adjacency relationship between the top vertex and the center one. Such edges correspond to an artificial split of a boundary between two regions (see Figure 2(b)). These edges, named double edges, may be characterized by the relationship $\varphi^2(d) = d$ where d is one of the dart of the double edge. We have for example on Figure 2(b), $\varphi(9) = -8$ and $\varphi(-8) = 9$, thus $\varphi^2(9) = 9$ and $\alpha^*(9)$ is a double edge. Another type of redundant edge is the direct self-loop, characterized by the relationship $\sigma(d) = \alpha(d)$ where d is one of the darts of the direct self-loop $\alpha^*(d)$. We have for example, on Figure 2, $\sigma(11) = -11$. Such edges may be interpreted in the dual combinatorial map as inner-boundaries (see edge $\alpha^*(11)$ in Figure 2(b)). Such redundant edges are removed by a removal kernel defined as a forest of the dual combinatorial map. This last constraint insures that no self-loop will be contracted in the dual combinatorial map and thus that no bridge may be removed in the initial one. Figure 3 represents the simplified combinatorial map deduced from the one represented in Figure 2 by the removal kernel $K_2 = \{\alpha^*(15,14,13,24), \alpha^*(9), \alpha^*(11,3), \alpha^*(19,18,17), \alpha^*(22,21)\}$. Note that given a sequence of double edges, the choice of the surviving edge is arbitrary. For example, a choice of the tree $\alpha^*(20,19,18)$ instead of $\alpha^*(19,18,17)$ would lead to an equivalent simplified combinatorial map with a surviving edge equal to $\alpha^*(17)$ instead of $\alpha^*(20)$.

Contraction and removal kernels specify the set of edges which must be contracted or removed. The creation of the reduced combinatorial map from a contraction or a removal kernel is performed in parallel by using connecting walks [3]. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, a kernel K and a surviving dart $d \in \mathcal{SD} = \boldsymbol{\mathcal{D}} - K$, the connecting walk associated to d is either equal to:

$$CW(d) = d, \varphi(d), \ldots, \varphi^{n-1}(d) \text{ with } n = Min\{p \in \mathbb{N}^* \mid \varphi^p(d) \in \mathcal{SD}\} \quad (2)$$



(a) connecting walks in $\mathcal{OG}$ defined by K_1

(b) Connecting walks in $\mathcal{OG}_1$ defined by K_2

Figure 4: Connecting walks defined by K_1 (a) and K_2(b).

if K is a contraction kernel and

$$CW(d) = d, \sigma(d), \ldots, \sigma^{n-1}(d) \text{ with } n = Min\{p \in \mathbb{N}^* \mid \sigma^p(d) \in \mathcal{SD}\} \tag{3}$$

If K is a removal kernel.

Figure 4 represents the connecting walks defined by K_1 and K_2 superimposed to the oriented graphs $\mathcal{OG}$ (Figure 1(c)) and $\mathcal{OG}_1$ (Figure 2(c)) respectively associated to $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and $G_1 = G/K_1 = (\mathcal{SD}_1, \sigma_1, \alpha)$. Let us consider the surviving dart -5 of G_1 (Figure 4(a)). Since K_1 is a contraction kernel $CW(-5)$ is equal to the sequence of non-surviving φ successors of -5. Since $\varphi(-5) = -10 \in K_1$ and $\varphi(-10) = 3 \in \mathcal{SD}_1$, we have $CW(-5) = -5.-10$. In the same way, let us now consider the combinatorial map $G_2 = G_1 \setminus K_2 = (\mathcal{SD}_2, \sigma_2, \alpha)$ and the surviving dart $5 \in \mathcal{SD}_2$ (Figure 4(b)). Since K_2 is a removal kernel, the connecting walk of 5 is defined as the sequence of non-surviving σ-successors of 5. Since $\sigma_1(5) = 3 \in K_2$ and $\sigma_1(3) = -7 \in \mathcal{SD}_2$ (Figure 2) we have: $CW(5) = 5.3$.

Given a kernel K and a surviving dart $d \in \mathcal{SD}$, such that $CW(d) = d.d_1 \ldots d_p$, the successor of d within the reduced combinatorial map $G' = (\mathcal{SD}, \sigma', \alpha)$ is retrieved from $CW(d)$ by the following equations [3]:

$$\begin{array}{lcll} \varphi'(d) & = & \varphi(d_p) & \text{if } K \text{ is a contraction kernel} \\ \sigma'(d) & = & \sigma(d_p) & \text{if } K \text{ is a removal kernel} \end{array} \tag{4}$$

Using Figure 4, we have for example $\varphi_1(-5) = \varphi(-10) = 3$ (Figure 4(a)) and $\sigma_2(5) = \sigma_1(3) = -7$ (Figure 4(b) see also Figure 3(c)).

Note that, if K is a contraction kernel, the connecting walk $CW(d)$ allows to compute $\varphi'(d)$. The σ-successor of d within the contracted combinatorial maps may be retrieved from $CW(\alpha(d)) = \alpha(d).d'_1, \ldots, d'_p$. Indeed, we obtain by using equations 1 and 4: $\varphi'(\alpha(d)) = \sigma'(\alpha(\alpha(d))) = \sigma'(d) = \varphi(d'_p)$. We may alternatively consider the sequence $d.CW^*(\alpha(d))$ where $CW^*(\alpha(d))$ denotes the sequence $CW(\alpha(d))$ without its first dart $\alpha(d)$. In this case,

using equation 4 the σ successor of a surviving dart d is provided by the last dart of $CW(d)$ if K is a removal kernel and by the last dart of $d.CW^*(\alpha(d))$ if K is a contraction kernel.

If K is a removal kernel (resp. a contraction kernel), $CW(d)$ (resp. $CW^*(\alpha(d))$) defines the sequence of non surviving darts which are mapped to the surviving dart d in the reduced combinatorial map. Such sequences encode thus the notion of reduction window within the combinatorial pyramid framework. In the following section we show how such sequences may be combined to define higher level objects such as regions.

4 Regions

The definition of regions within the combinatorial pyramid framework supposes first to express the notion of a connected set of pixels in terms of darts. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, we define a connected sequence of darts as a sequence $P = d_1 \ldots, d_n$ such that all darts of the sequence are distinct and each dart $(d_i)_{i \in \{2,\ldots,n\}}$ is either the σ or φ successor of d_{i-1}. Intuitively, two darts of such a sequence belong either to the same vertex or to adjacent vertices. If d_1 is the σ or φ successor of d_n, such a sequence is called a cycle. Note that a connected sequence of darts defines either a path of a cycle in the oriented graph $\mathcal{OG}$(Section 2). Given the connected sequence of darts, we can define the notion of connected set of darts. This notion is stronger than the usual notion of connected set of vertices since one can easily show that the set of vertices defined by a connected set of darts is connected. However, a region is usually defined as a connected set of pixels rather than a connected set of darts. Therefore a connected set of darts must contain all the darts of its vertices in order to be called a region. Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and a connected set of darts $R \subset \boldsymbol{\mathcal{D}}$, this last condition may be written: $\sigma^*(R) = R$. The above considerations are resumed in the following definition:

Definition 2 Region

Given a combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$, a set of darts $R \subset \boldsymbol{\mathcal{D}}$ is called a region of G iff:

- *R is connected: Given any two darts $(d, d') \in R^2$ it exists one connected sequence of darts P included in R which connects either d to d' or d' to d.*
- *R contains its vertices: $\sigma^*(R) = R$*

Let us consider an initial combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and a contracted one $G' = (\mathcal{SD}, \sigma', \alpha)$ deduced from G by a contraction kernel K. Each tree of K contracts a connected set of vertices into one surviving vertex. Let us consider such a surviving vertex $\sigma'^*(d_1) = (d_1, \ldots, d_p)$. Since each dart d_i of this σ'-orbit is connected in G to $d_{i+1} = \sigma'(d_i)$ by $d_i.CW^*(\alpha(d_i))$ (see Section 3), the reduction window of the vertex $\sigma'^*(d_1)$ is encoded by:

$$R_{\sigma'^*(d_1)} = d_1 CW^*(\alpha(d_1)) \ldots . d_p CW^*(\alpha(d_p)) \tag{5}$$

In the same way, if G' is deduced from G by a removal kernel, each dart d_i of $\sigma'^*(d_1)$ is connected in G to d_{i+1} by $CW(d_i)$. Therefore, the reduction window associated to this vertex is equal to:

$$R_{\sigma'^*(d_1)} = CW(d_1) \ldots . CW(d_p) \tag{6}$$

Since each vertex of the initial combinatorial map G corresponds to one pixel, the vertex-reduction windows defined above should correspond to the usual notion of region. However, it remains to show that these regions fulfill the requirements of Definition 2.

If K is a removal kernel, any connecting walk is included in a σ orbit by definition (equation 3). Moreover, each connecting walk $CW(d_i)$ is connected to $CW(d_{i+1})$ by σ (equations 4 and 6). Thus $R_{\sigma'^*(d_1)}$ is a sequence of σ successors included in $\sigma^*(d_1)$. Moreover, the σ-successor of the last dart of $CW(d_p)$, is equal to d_1 (equation 4). Therefore, $R_{\sigma'^*(d_1)}$ is a σ-orbit included in $\sigma^*(d_1)$. By definition of an orbit we have: $R_{\sigma'^*(d_1)} = \sigma^*(d_1)$. The region $R_{\sigma'^*(d_1)} = \sigma^*(d_1)$ is thus trivially connected and contains all its vertices.

If K is a contraction kernel, each sequence $CW^*(\alpha(d_i))$ is connected (equation 2). Moreover, each dart d_i is the σ-successor of the second dart of $CW(\alpha(d_i))$ (equation 2 and 1). The sequence $d_i CW^*(\alpha(d_i))$ is thus connected. Finally, the φ successor of the last dart of each sequence $d_i CW^*(\alpha(d_i))$ is equal to d_{i+1}(equation 4). Each sequence $d_i CW^*(\alpha(d_i))$ in $R_{\sigma'^*(d_1)}$ is thus connected to the following one and $R_{\sigma'^*(d_1)}$ is connected.

Note that using the circular order defined on $\sigma'^*(d_1)$, d_1 is either the σ or φ successor of the last dart of $R_{\sigma'^*(d_1)}$. The region $R_{\sigma'^*(d_1)}$ corresponds thus to a closed connected sequence of darts which defines a cycle in the oriented graph $\mathcal{OG}$ associated to G.

The proof that $R_{\sigma'^*(d)}$ contains its vertices, is based on a study of the connections between the trees of a contraction kernel and the connecting walks. We have in particular the following properties:

Proposition 1 *Given a contraction kernel K, an initial combinatorial map $G = (\boldsymbol{\mathcal{D}}, \sigma, \alpha)$ and the contracted combinatorial map $G' = (\mathcal{SD} = \boldsymbol{\mathcal{D}} - K, \sigma', \alpha)$, the trees of K satisfy:*

$$\forall \mathcal{T} \in \mathcal{CC}(K), \quad \forall d_1 \in \sigma^*(\mathcal{T}) \cap \mathcal{SD} \quad \sigma'^*(d_1) = \sigma^*(\mathcal{T}) \cap \mathcal{SD} \tag{7}$$

$$\mathcal{T} = \cup_{j=1}^{p} CW^*(\alpha(d_j)) \tag{8}$$

with $\sigma'^*(d_1) = (d_1, \ldots, d_p)$.

Equation 7 is demonstrated in [2]. Equation 8 may be deduced from equation 7 using the fact that $\cup_{j=1}^{p} CW^*(\alpha(d_j))$ is a connected set of non surviving darts and is thus included in a particular tree $\mathcal{T}$ of K.

Equation 7 may be understood as follows: The set of surviving darts belonging to the leafs of a tree $\mathcal{T}$ define the adjacency relationships between $\mathcal{T}$ and the other vertices of G. Note that we have a circular order on the σ'-orbit $\sigma'^*(d_1)$. Therefore, the set of surviving darts encoding the adjacency relationships of the tree is ordered according to counter-clockwise orientation when turning around the tree.

Each surviving dart d_i of $\sigma^*(T) \cap \mathcal{SD} = \sigma'^*(d)$ is connected in G to $d_{i+1} = \sigma'(d_i)$ by $d_i.CW^*(\alpha(d_i))$ (Section 3). Therefore, the two surviving darts d_i and d_{i+1} are connected in G by a sequence $CW^*(\alpha(d_i))$ of non surviving darts. Equation 8 shows that the union of such sequences cover the whole tree.

Let us consider one surviving vertex $\sigma'^*(d_1) = (d_1, \ldots, d_p)$ and one tree $\mathcal{T} = \cup_{j=1}^{p} CW^*(\alpha(d_j))$. Since $\mathcal{T}$ is a connected component of K, a dart d belonging to $\sigma^*(\mathcal{T}) \cap K$ must belong to $\mathcal{T}$, otherwise, $\mathcal{T}$ would be connected to an other tree of K. We have thus $\sigma^*(\mathcal{T}) \cap K = \mathcal{T}$. Moreover,

since $\mathcal{SD} = \boldsymbol{\mathcal{D}} - K$, we have $\boldsymbol{\mathcal{D}} = \mathcal{SD} \cup K$ and:

$$\begin{array}{rcll} \sigma^*(\mathcal{T}) & = & \sigma^*(\mathcal{T}) \cap \boldsymbol{\mathcal{D}} = \sigma^*(\mathcal{T}) \cap (\mathcal{SD} \cup K) & \\ & = & (\sigma^*(\mathcal{T}) \cap \mathcal{SD}) \cup (\sigma^*(\mathcal{T}) \cap K) & \\ & = & \sigma'^*(d_1) \cup \mathcal{T} & \text{(equation 7)} \\ & = & \sigma'^*(d_1) \cup \bigcup_{j=1}^{p} CW^*(\alpha(d_j)) & \text{(equation 8)} \end{array}$$

where $\sigma'^*(d_1) = (d_1, \ldots, d_p)$.

Therefore the region $R_{\sigma'^*(d_1)}$ associated to the surviving vertex $\sigma'^*(d_1)$ defines an order on the set $(d_1, \ldots, d_p) \cup \bigcup_{j=1}^{p} CW^*(\alpha(d_j)) = \sigma^*(\mathcal{T})$ (see equation 5). Since the operator σ^* is idempotent we have:

$$\sigma^*(R_{\sigma'^*(d)}) = \sigma^*(\sigma^*(\mathcal{T})) = \sigma^*(\mathcal{T}) = R_{\sigma'^*(d)}$$

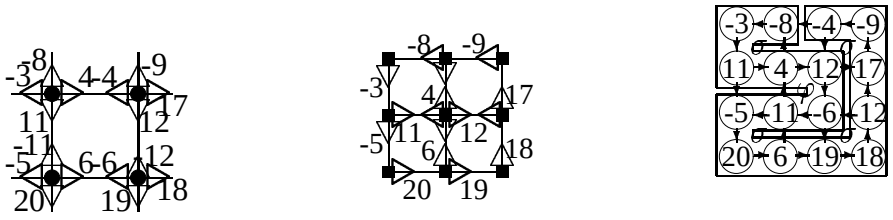
The sequence of darts $R_{\sigma'^*(d)}$ is thus connected and contains its vertices. It is thus a region which defines a connected set of vertices.

Figure 5 illustrates three alternative representations of the region associated to the contracted vertex $\sigma'^*(-8)$ (see central vertex in Figure 2). Note that, all darts in $R_{\sigma'^*(-8)}$ are associated to a triangle in Figure 5(b). However, the name of the darts $-4, -6, -11$ and -12 is not displayed in this figure in order to not overload it. The vertex $\sigma'^*(-8)$ is defined by the sequence of darts -8, -3, 11,-11,-5,20,19, 18, 17 and -9(Figure 2). Using equation 5 the region $R_{\sigma'^*(-8)}$ is defined as:

$$R_{\sigma'^*(-8)} = \boxed{\text{-8}}.\boxed{\text{-3}}.\boxed{\text{11.4.12.-6}}.\boxed{\text{-11}}.\boxed{\text{-5}}.\boxed{\text{20.6}}.\boxed{\text{19}}.\boxed{\text{18.-12}}.\boxed{\text{17}}.\boxed{\text{-9.-4}} \qquad (9)$$

where each box surrounds a sequence $d.CW^*(\alpha(d))$ with $d \in \sigma'^*(-8)$ (Figure 4(a)).

The region $R_{\sigma'^*(-8)}$ is composed of 4 pixels with 4 cracks defining inner boundaries and 8 cracks defining the boundary of the region. We can note on equation 9 that all the edges defining inner boundaries are included in $R_{\sigma'^*(-8)}$. We have indeed, $\alpha^*(4, 11, 6, 12) \subset R_{\sigma'^*(-8)}$. This property is a direct consequence of the fact that a region contains its vertices (Definition 2). Indeed, if an edge $\alpha^*(d)$ defines an inner boundary of a region R both $\sigma^*(d)$ and $\sigma^*(\alpha(d))$ must



(a) $R_{\sigma_1^*(-8)}$ in G (b) $R_{\sigma_1^*(-8)}$ in $\overline{G}$ (c) $R_{\sigma_1^*(-8)}$ in $\mathcal{OG}$

Figure 5: The region $R_{\sigma_1^*(-8)}$ defined by K_1 (see Figure 2)

be included in R. Therefore, d and $\alpha(d)$ must belong to R. Conversely, all darts defining the boundary of a region R cannot have their α-successor in R (see e.g. the darts 17 to 20 in equation 9 and Figure 5). We can also note, on equation 9 (see also Figure 5(b)) that the sequence of darts $-8, -3, -5, 20, 19, 18, 17, -9$ defining the boundary of $R_{\sigma'^*(-8)}$ with a counter-clockwise orientation is included in $R_{\sigma'^*(-8)}$. Moreover, the order of this sequence is respected in $R_{\sigma'^*(-8)}$. This last property has been verified in all our experiments but is not yet fully demonstrated.

5 Conclusion

We have defined in this paper the notion of regions within the combinatorial pyramid framework. This result allows us to either draw the image partition associated to one level of the pyramid or to extract parameters from regions. Moreover, the reduction window R of any high level pixel has been found to form a directed Hamiltonian circuit in the sub-graph of $\mathcal{OG}$ restricted to R. However, the region defined in this paper are based on connecting walks which correspond to the notion of reduction window within the pyramid framework. A general definition based on the receptive fields of darts should be studied. Moreover, we also plan to study finer properties of regions. Such results should allow us to retrieve the set of pixels of one region or its boundary without traversing all the darts of the region. Finally, combinatorial maps being formally defined in any dimensions, these results should be extended to higher dimensions in order to define nD Combinatorial Pyramids.

References

[1] M. Bister, J. Cornelis, and A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognit Letter.*, 11(9):605–617, Sept. 1990.

[2] Luc Brun and Walter Kropatsch. Pyramids with combinatorial maps. Technical Report PRIP-TR-057, PRIP, TU Wien, 1999.

[3] Luc Brun and Walter Kropatsch. Contraction kernels and combinatorial maps. In Jean Michel Jolion, Walter Kropatsch, and Mario Vento, editors, 3^{rd} *IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, pages 12–21, Ischia Italy, May 2001. IAPR-TC15, CUEN.

[4] Peter Burt, Tsai-Hong Hong, and Azriel Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchial computation. *IEEE Transactions on Sustems, Man and Cybernetics*, 11(12):802–809, December 1981.

[5] Walter G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, Vol. 142(No. 6):pp. 366–374, December 1995.

[6] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):307–316, APRIL 1991.

Removal and contraction for n-dimensional generalized maps

Guillaume Damiand and Pascal Lienhardt

Laboratoire S.I.C., B.P. 30179, 86962 Futuroscope Chasseneuil Cedex, France

Tel.: +33 (0)5 49 49 65 67, Fax.: +33 (0)5 49 49 65 70

{damiand, lienhardt}@sic.sp2mi.univ-poitiers.fr

Abstract

In this paper we define removal and contraction operations in the generalized maps framework. These two operations are often used in graph theory to define images pyramids that allow multi-scale representation. However graphs don't represent the whole topological information of an image, even in the 2-dimensional case, contrary to generalized maps. We define here operations for removing or contracting cells of any dimension, in n-dimensional space. This work is the starting point for defining generalized map pyramids in any dimension. We prove in this paper the validity of these operations, and show that several different operations can be applied at the same time when some preconditions are satisfied.

The removal and contraction operations are studied since several years in graph theory to define image irregular pyramids [17, 18]. These operations allow to gradually simplify a graph, while keeping some important properties (for example the connectivity). Pyramids are used for image analysis and provide a multi-scale representation that allows multi-level analysis and treatments.

However graphs don't represent the whole topological information contained in an image. For instance some graphs don't represent the inclusion notion, or multi-adjacency, or different images can correspond to a same graph. These problems are even more present in upper dimension.

To solve these problems which are crucial in an analysis aim, several topological models have been proposed these last few years, in particular models based on combinatorial maps. The use of these models for image representation has been first studied in dimension 2 [6, 15], then extended in dimension 3 [4, 1, 2, 3] and to dimension n (some works remains to be done here) [12].

For example, we can use combinatorial maps and some removal operations to define a model that represents discrete objects. We start with a model where each voxel of the object is represented (figure 1.*a*), then remove each face incident to two voxels of the object (figure 1.*b*). Then we remove also each degree two edge incident to two coplanar faces (figure 1.*c*), and at last we remove each degree two vertex incident to two aligned edges (figure 1.*d*). The obtained model represents the boundaries of the discrete object with only few elements.

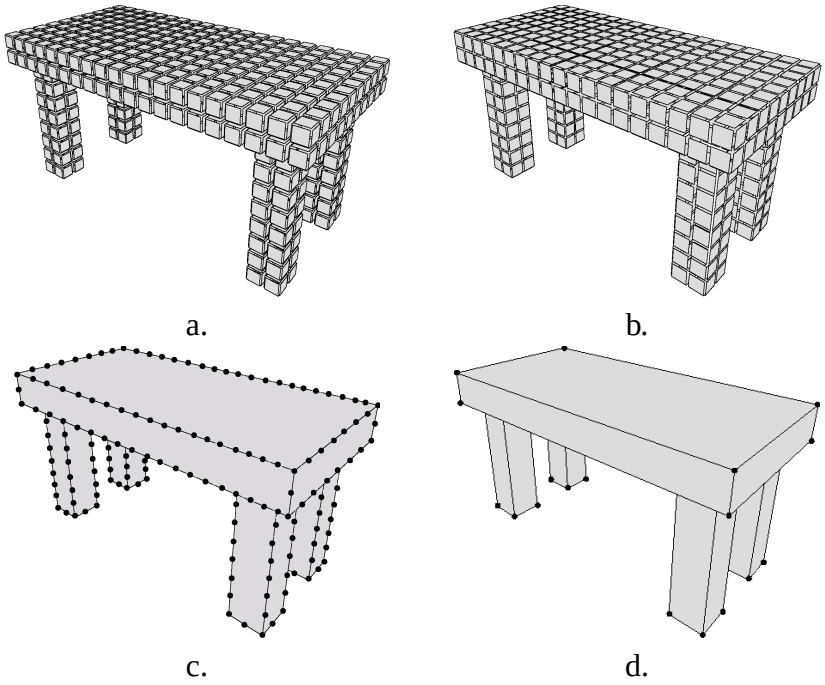


Figure 1: Simplification of a discrete object.

To define 2-dimensional combinatorial map pyramids, Luc Brun and Walter Kropatsch revisit works made on graphs in the combinatorial maps framework [7, 8, 9, 10]. To initiate an extension of their works for any dimension, we propose in this paper a general definition of the removal and contraction operations. For any dimension n, cells of any dimension i ($0 \leq i \leq n$) can be contracted or removed. In some cases, according to the respective values of i and n, a simple precondition have to be satisfied. In an aim of effectiveness, we have studied the possibility of simultaneously performing a set of operations. We show in this paper that it is possible when the contracted or removed cells are disjoined.

These operations are here defined for n-dimensional generalized maps (or n-G-maps) [20]. They are an extension of combinatorial maps, initially defined in dimension 2 for representing planar graphs [13, 16, 11]. See [19] for a comparison between these structures and other topological models. Generalized maps and related notions are recalled in section 1. Contraction and removal operations for 1D and 2D cases are presented in section 2 and 3, and the definition in nD is given in section 4. These definitions are then generalized in section 5, in order to perform simultaneously sets of any operations. We conclude section 6.

1 Generalized maps recall

A subdivision of an n-dimensional topological space is a partition of this space into i-dimensional cells (or i-cells), for $0 \leq i \leq n$. Two cells are *incident* if one belongs to the boundary of the other. Two *adjacent* i-cells are incident to a same cell.

n-dimensional generalized maps, or n-G-maps, are a combinatorial model, defined in order to represent the topology of space subdivisions. More precisely, an n-G-maps represents the topology of an n-quasi-manifold, orientable or not, with or without boundaries [20].

Definition 1 (Generalized map) *Let $n \geq -1$. A n-dimensional generalized map (or n-G-map) is an algebra $G = (B, \alpha_0, \ldots, \alpha_n)$ where:*

1. *B is a finite set of darts;*
2. *$\forall i,\ 0 \leq i \leq n$, α_i is an* involution[1] *on B;*
3. *$\forall i,\ 0 \leq i \leq n, \forall j,\ i+2 \leq j \leq n$, $\alpha_i\alpha_j$ is an* involution.

Let G be an n-G-map, and S be the corresponding subdivision. A dart of G corresponds to an $(n+1)$-tuple of cells $(c_0, \ldots, c_n)$, where c_i is an i-dimensional cell, and c_i and $c_{(i+1)}$ are incident (cf. [5] and figure 2). α_i associate darts corresponding with $(c_0, \ldots, c_n)$ and $(c'_0, \ldots, c'_n)$, with $c_j = c'_j$ for $0 \leq j \leq n$ and $j \neq i$, $c_i \neq c'_i$ (α_i swaps the two i-cells that are incident to the same $(i-1)$ and $(i+1)$-cells).

In the figures, two darts in relation by α_0 are represented by two segments sharing a little vertical segment. Two darts in relation by α_1 share a same point; two darts in relation by α_2 are parallel and close to each other. When two darts b_1 and b_2 are such that $b_1\alpha_i = b_2$ ($0 \leq i \leq n$), b_1 is *i-sewn* with b_2. A dart invariant for α_i ($0 \leq i \leq n$) is not sewn, it is *i-free*.

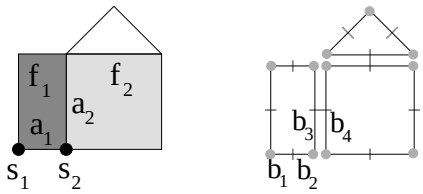


Figure 2: A 2D subdivision and the corresponding 2-G-map.

Figure 2.*a* shows a 2-dimensional object, and figure 2.*b* the 2-G-map representing this object. On figure 2.*a*, b_1 corresponds to (s_1, a_1, f_1), $b_2 = b_1\alpha_0$ corresponds to (s_2, a_1, f_1). $b_3 = b_2\alpha_1$ corresponds to (s_2, a_2, f_1), and $b_4 = b_3\alpha_2$ corresponds to (s_2, a_2, f_2).

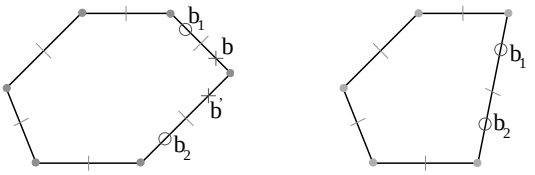
G-maps represent cells in an implicit way:

Definition 2 (i-cell) *Let G be an n-G-map, b be a dart and $i \in \{0, \ldots, n\}$. The i-cell incident to b is the orbit*[2] *$< \alpha_0, \ldots, \alpha_{(i-1)}, \alpha_{(i+1)}, \ldots, \alpha_n > (b)$, noted $<>_{N-\{i\}}$.*

Intuitively, an i-cell is the set of all darts which can be reached starting from b, by using all involutions except α_i. The set of i-cells is a partition of the darts of the G-map, for each i between 0 and n. Two cells are disjoined if their intersection is empty, i.e. when no dart is shared by the cells.

[1] An involution f on S is a bijection of S on S such that $f = f^{-1}$.

[2] Let $\{\Pi_0, \ldots, \Pi_n\}$ be a set of permutation on B. $< \Pi_0, \ldots, \Pi_n > (b) = \{\Phi(b), \Phi \in < \Pi_0, \ldots, \Pi_n >\}$, where $< \Pi_0, \ldots, \Pi_n >$ denotes the group of permutations generated by $\Pi_0, \ldots, \Pi_n$.



a. Initial 1-G-map. b. Result.

Figure 3: 0-removal in the general case.

2 Dimension 1

Intuitively and in a general way for a n-dimensional space, the removal of an i-cell consists in removing this cell and in merging (when it is necessary) its two incidents $(i+1)$-cells. The contraction of an i-cell is the dual operation. It consists in contracting this cell into an $(i-1)$-cell. For any dimension n, removal is defined for $0 \ldots (n-1)$-cells, and contraction for $1 \ldots n$-cells. For dimension 1, there exists only the 0-removal and the 1-contraction.

2.1 0-removal

The 0-removal consists in removing a vertex $C =< \alpha_1 > (b)$. Let $C\alpha_0 = \{b'' \mid \exists b' \in C$ such that $b'\alpha_0 = b''\}$, the "neighbor" darts of C for α_0, and $B^S = C\alpha_0 - C$, the "neighbor" darts of C for α_0 that don't belong to C (see figure 3). The G-map resulting from the 0-removal of C is obtained by redefining α_0 for the darts of B^S as follow: $\forall b' \in B^S, b'\alpha'_0 = b'(\alpha_0\alpha_1)^k\alpha_0$, where k is the smallest integer such that $b'(\alpha_0\alpha_1)^k\alpha_0 \in B^S$. Note that α_1 is not modified by the 0-removal.

Figure 3 shows an example of a 0-removal in the general case. Darts marked with crosses belong to C, and those marked with circles belong to B^S. Figure 4 shows all other possible configurations (indeed, a 1-G-map defines either a cycle or a path of edges).

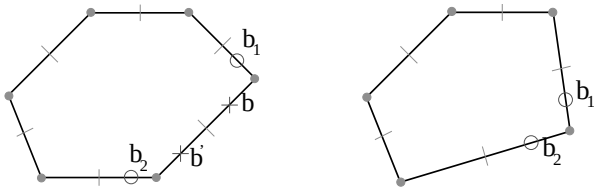
a. Initial 1-G-map. b. Result. c. Initial 1-G-map. d. Result.

Figure 4: Particular configurations : for both cases, the removal produces a 1-G-map, in which b_1 is 0-free.

2.2 1-contraction

1-contraction is the dual operation of 0-removal. It consists in contracting an edge $C =< \alpha_0 > (b)$ into a vertex. Let $B^S = C\alpha_1 - C$. The G-map resulting from the 1-contraction is obtained

by redefining α_1 for the darts of B^S as follow: $\forall b' \in B^S, b'\alpha'_1 = b'(\alpha_1\alpha_0)^k\alpha_1$ where k is the smallest integer such that $b'(\alpha_1\alpha_0)^k\alpha_1 \in B^S$. Note that α_0 is not modified by the 1-contraction.



a. Initial 1-G-map. b. Result.

Figure 5: 1-contraction in the general case.

Figure 5 shows an example of 1-contraction in the general case, and figure 6 shows the two possible particular cases. We can check that a correct 1-G-map is produced in any case, and that b_1 becomes 1-free for the two particular configurations.

a. Initial 1-G-map. b. Result. c. Initial 1-G-map. d. Result.

Figure 6: 1-contraction for particular configurations.

3 Dimension 2

We have now two different removal operations (0 and 1-removal) and two contraction operations (1 and 2-contraction). Figures 7-11 show examples for the general case: in order to simply, only darts concerned by the operation appear in these figures.

3.1 0-removal

0-removal consists in removing a 0-cell $C =< \alpha_1, \alpha_2 > (b)$. Let $B^S = C\alpha_0 - C$. This operation can be applied only if the following precondition is satisfied: $\forall b' \in C$, $b'\alpha_1\alpha_2 = b'\alpha_2\alpha_1$. This constraint corresponds, in the general case, to the fact that the degree of the vertex is equal to 2 (2 edges are incident to the vertex). If this constraint is not satisfied, we can't know how to join the darts incident to C, and it is then impossible to define the removal in a simple way. [14] proposes a generalization of this operation, but it is complex and cannot be used for an automatic process (e.g. automatic image processing).

The G-map resulting from the 0-removal is obtained by redefining α_0 for the darts of B^S as follow: $\forall b' \in B^S, b'\alpha'_0 = b'(\alpha_0\alpha_1)^k\alpha_0$ where k is the smallest integer such that $b'(\alpha_0\alpha_1)^k\alpha_0 \in$

B^S. Note that this redefinition of α_0 is the same as for dimension 1: concerned darts are different (here, it is a 0-cell in 2D).

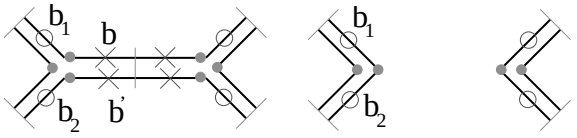
a. Initial 2-G-map. b. Result.

Figure 7: 0-removal in the general case.

Figure 7 shows an example of 0-removal in the general case. This figure represents a part of a subdivision where the two central edges are shared by two faces. Intuitively, this operation consists in applying twice the 0-removal in dimension 1. This operation is also valid for particular configurations (cf. section 4).

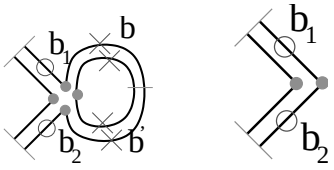
3.2 1-removal

1-removal consists in removing a 1-cell $C =< \alpha_0, \alpha_2 > (b)$. This can be achieved without any precondition. Let $B^S = C\alpha_1 - C$. The G-map resulting from the 1-removal is obtained by redefining α_1 for the darts of B^S as follow: $\forall b' \in B^S, b'\alpha'_1 = b'(\alpha_1\alpha_2)^k\alpha_1$, where k is the smallest integer such that $b'(\alpha_1\alpha_2)^k\alpha_1 \in B^S$. Figure 8 shows an example of 1-removal in the general case, and figure 9 for a particular case. For this last example, $k = 2$ since the removed edge is incident twice to the same vertex.



a. Initial 2-G-map. b. Result.

Figure 8: 1-removal in the general case.

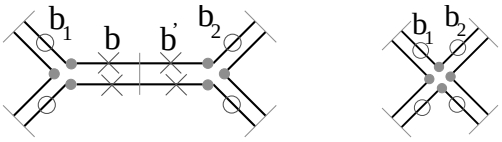


a. Initial 2-G-map. b. Result.

Figure 9: 1-removal of a loop.

3.3 1-contraction

1-contraction is the dual operation of 1-removal. The G-map resulting from the 1-contraction of the 1-cell $C =< \alpha_0, \alpha_2 > (b)$ is obtained by redefining α_1 for the darts of $B^S = C\alpha_1 - C$ as follow: $\forall b' \in B^S, b'\alpha'_1 = b'(\alpha_1\alpha_0)^k\alpha_1$ where k is the smallest integer such that $b'(\alpha_1\alpha_0)^k\alpha_1 \in B^S$. There is no precondition for this operation. Figure 10 shows an example of 1-contraction in the general case. This definition is the same than the 1-contraction in dimension 1, the difference lies in the definition of the cell.

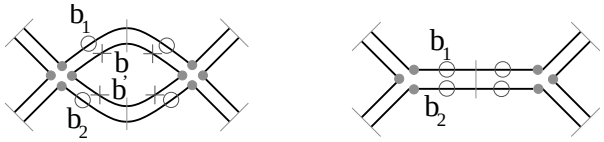


a. Initial 2-G-map. b. Result.

Figure 10: 1-contraction in the general case.

3.4 2-contraction

2-contraction is the dual operation of 0-removal. It consists in contracting a 2-cell $C =< \alpha_0, \alpha_1 > (b)$ into a 1-cell. Let $B^S = C\alpha_2 - C$. This operation can be applied only if the following precondition is satisfied: $\forall b' \in C,\ b'\alpha_0\alpha_1 = b'\alpha_1\alpha_0$. This constraint corresponds to the fact that the degree of the face is equal to 2 (2 edges are incident to the face). The G-map resulting from the 2-contraction is obtained by redefining α_2 for the darts of B^S as follow: $\forall b' \in B^S, b'\alpha'_2 = b'(\alpha_2\alpha_1)^k\alpha_2$, where k is the smallest integer such that $b'(\alpha_2\alpha_1)^k\alpha_2 \in B^S$. Figure 11 presents an example of 2-contraction in the general case.



a. Initial 2-G-map. b. Result.

Figure 11: 2-contraction in the general case.

4 Dimension n

The general definition of i-cell removal in a n-dimensional space is the following:

Definition 3 (i-cell removal) *Let $G = (B, \alpha_0, \ldots, \alpha_n)$ be an n-G-map, $i \in \{0, \ldots, n-1\}$ and $C =<>_{N-\{i\}} (b)$ an i-cell, such that: $\forall b' \in C, b'\alpha_{(i+1)}\alpha_{(i+2)} = b'\alpha_{(i+2)}\ \alpha_{(i+1)}$. Intuitively, this condition corresponds to the fact that the degree of the i-cell is 2. Note that if $i = n-1$ this condition doesn't apply and we can always perform the $(n-1)$-removal of any $(n-1)$-dimensional cell. Let $B^S = C\alpha_i - C$, the set of darts α_i-sewn to C that do not belong to C. The n-G-map resulting from the removal of this i-cell is $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ defined by:*

- $B' = B - C$;
- $\forall j \in \{0, \ldots, n\} - \{i\}$, $\alpha'_j = \alpha_j|_{B'}$;
- $\forall b' \in B' - B^S$, $b'\alpha'_i = b'\alpha_i$;
- $\forall b' \in B^S$, $b'\alpha'_i = b'(\alpha_i\alpha_{(i+1)})^k\alpha_i$,
  where k is the smallest integer such that $b'(\alpha_i\alpha_{(i+1)})^k\alpha_i \in B^S$.

Note that this operation consists only in redefining α_i for the darts of B^S. Note also that, for darts $b' \in B' - B^S$, $b'\alpha'_i = b'(\alpha_i\alpha_{(i+1)})^k\alpha_i$, where $k = 0$ is the smallest integer such that $b'\alpha'_i \in B' - B^S$.

The contraction operation is defined in a similar way: more precisely, the i-contraction is the dual operation of the $(n-i)$-removal.

Definition 4 (i-cell contraction) *Let $G = (B, \alpha_0, \ldots, \alpha_n)$ be an n-G-map, $i \in \{1, \ldots, n\}$ and $C =<>_{N-\{i\}} (b)$ an i-cell, such that: $\forall b' \in C, b'\alpha_{(i-1)}\alpha_{(i-2)} = b'\alpha_{(i-2)}\ \alpha_{(i-1)}$. Note that if $i = 1$ this condition doesn't apply and we can always perform the 1-contraction of any edge. Let $B^S = C\alpha_i - C$, the set of darts α_i-sewn to C, that do not belong to C. The n-G-map resulting from the contraction of this i-cell is $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ defined by:*

- $B' = B - C$;
- $\forall j \in \{0, \ldots, n\} - \{i\}$, $\alpha'_j = \alpha_j|_{B'}$;
- $\forall b' \in B' - B^S$, $b'\alpha'_i = b'\alpha_i$;
- $\forall b' \in B^S$, $b'\alpha'_i = b'(\alpha_i\alpha_{(i-1)})^k\alpha_i$,
  where k is the smallest integer such that $b'(\alpha_i\alpha_{(i-1)})^k\alpha_i \in B^S$.

Theorem 1 *Given an n-G-map G and a i-cell C to remove that satisfies the precondition of the operation, G' is an n-G-map.*

Proof 1 *We differentiate three cases. First for $j \neq i$, involutions α_j aren't redefined but only restricted to the darts of the final G-map. Then, for $j = i$, we distinguish two cases, depending on if darts belonging or not to B^S.*

1. ***for $j \neq i$:*** *Let $b_1 \in B'$. We have to show that $b_2 = b_1\alpha_j \in B'$. Suppose that $b_2 \notin B'$. Then, $b_2 \in C$ since $B' = B - C$. As $C =< \alpha_0, \ldots, \alpha_{(i-1)}, \alpha_{(i+1)}, \ldots, \alpha_n > (b)$, if $b_2 \in C$ then $b_1 = b_2\alpha_j \in C$; $b_1 \notin B'$: contradiction. So $b_1\alpha'_j = b_1\alpha_j \in B'$ and $b_1\alpha'_j\alpha'_j = b_1\alpha_j\alpha_j = b_1$: α'_i is well defined and is an involution.*
2. ***for $j = i$:***
   (a) ***for $b_1 \in B' - B^S$:*** *We show that $b_2 = b_1\alpha_i \in B' - B^S$ in a similar way as in the previous case. Assume $b_2 \in C$. In this case, $b_1 = b_2\alpha_i \in B^S \cup C$ by definition of B^S and this is in contradiction with $b_1 \in B' - B^S$. If $b_2 \in B^S$, then $b_1 = b_2\alpha_i \in C$; this is also in contradiction with $b_1 \in B' - B^S$. So $b_1\alpha'_i = b_1\alpha_i \in B' - B^S$. Thus $b_1\alpha'_i\alpha'_i = b_1\alpha_i\alpha_i = b_1$; α'_i is well defined and is an involution.*
   (b) ***for $b_1 \in B^S$:*** *We show that b_2 exists, such that $b_2 \in B^S$ and $b_2 = b_1(\alpha_i\alpha_{(i+1)})^k\alpha_i$. There are two different cases, depending on if $< \alpha_i\alpha_{(i+1)} > (b_1)$ is a cycle or a path.*
      - *If it is a cycle, no dart of this cycle is i-free or $(i+1)$-free. Then, k exists such that $b_1(\alpha_i\alpha_{(i+1)})^k\alpha_i = b_1\alpha_{(i+1)} \notin C$. Let k_1 be the smallest value such that $b_1(\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i \notin C$: all darts $b_1\alpha_i$, $b_1\alpha_i\alpha_{(i+1)}$, …, $b_1(\alpha_i\alpha_{(i+1)})^{k_1}$ belong to C. $b_2 = b_1(\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i \in B^S$, and so $b_1\alpha'_i = b_2$. α'_i is well defined. Moreover, α'_i is an involution, since $b_1\alpha'_i\alpha'_i = b_1(\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i\ (\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i = b_1$.*

- *If it is a path, then there are three possibilities. Let k_1 the smallest integer such that $b_3 = b_1(\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i$ is i-free or $(i+1)$-free. If k_2 exists, such that $k_2 < k_1$ and $b_1(\alpha_i\alpha_{(i+1)})^{k_2}\alpha_i \in B^S$, the proof is similar to the previous case. If b_3 is i-free, $b_1(\alpha_i\alpha_{(i+1)})^{k_1} Id \alpha_{(i+1)}(\alpha_i\alpha_{(i+1)})^{k_1-1}\alpha_i = b_1 \in B^S$ ($k = 2k_1$). If b_3 is $(i+1)$-free, $b_1(\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i Id\ (\alpha_i\alpha_{(i+1)})^{k_1}\alpha_i = b_1 \in B^S$ ($k = 2k_1 + 1$). In these two cases, we have $b_1\alpha'_i = b_1$ and so $b_1\alpha'_i\alpha'_i = b_1$.*

We have now to prove that $\forall j,\ 0 \leq j \leq n$, $\forall k,\ j+2 \leq k \leq n$, $\alpha'_j\alpha'_k$ is an involution.

- ***for $j \neq i$ and $k \neq i$:*** *This is obvious since $\alpha'_j = \alpha_j$ and $\alpha'_k = \alpha_k$. As G is a G-map, $\alpha_j\alpha_k = \alpha'_j\alpha'_k$ is an involution.*
- ***for $j = i$:*** *We are going to show that $\forall b_1 \in G'$, we have $b_1\alpha'_i\alpha'_k = b_1\alpha'_k\alpha'_i$.*
  1. ***for $b_1 \in B' - B^S$:*** *$b_1\alpha'_i = b_1\alpha_i$ and $\alpha'_k = \alpha_k$. Since G is a G-map, $b_1\alpha_i\alpha_k = b_1\alpha_k\alpha_i$; since $b_1\alpha_k = b_1\alpha'_k \in B - B^S$, $b_1\alpha_k\alpha_i = b_1\alpha'_k\alpha'_i$; thus $b_1\alpha'_i\alpha'_k = b_1\alpha'_k\alpha'_i$.*
  2. ***for $b_1 \in B^S$:*** *$\forall b \in B$, $b\alpha_i\alpha_k = b\alpha_k\alpha_i$ (def. of G-maps), and $\forall b \in C$, $b\alpha_i\alpha_{(i+1)} = b\alpha_{(i+1)}\alpha_i$ (precondition of the operation). So, $b_1\alpha_k(\alpha_i\alpha_{(i+1)})^p\alpha_i = b_1(\alpha_i\alpha_{(i+1)})^p\alpha_i\alpha_k$, since α_k commutes with α_i, and α_k with $\alpha_{(i+1)}$ for all darts belonging to C (each dart of the path $b_1\alpha_k(\alpha_i\alpha_{(i+1)})^p$ belongs to C). So, $b_1\alpha'_i\alpha'_k = b_1(\alpha_i\alpha_{(i+1)})^p\alpha_i\alpha_k = b_1\alpha_k(\alpha_i\alpha_{(i+1)})^p\alpha_i$; since $b_1 \in B^S$, $b_1\alpha_k \in B^S$ and $b_1\alpha_k\ (\alpha_i\alpha_{(i+1)})^p\alpha_i = b_1\alpha'_k\alpha'_i$.*
- ***for $k = i$:*** *Similar to the previous case.* □

The proof for the contraction operation is equivalent by duality (exchange $\alpha_{(i+1)}$ and $\alpha_{(i-1)}$).

5 Generalisations

Previous definitions allow to perform the removal or contraction of an unique cell. It is interesting to apply simultaneously several different operations, for efficiency reasons. Concretely, let G be an n-G-map. Darts belonging to cells that have to be removed or contracted are marked with the dimension and type of the corresponding operation. Operations can be simultaneously applied if the cells are disjoined. The resulting G-map can be directly computed. This generalization is presented in several steps. First, we show that it is possible to simultaneously perform removals (resp. contractions) of several i-cells for a given i ($0 \leq i \leq n$).

Generalisation 1 *The previous definition of removal (resp. contraction) stands for the removal (resp. contraction) of a set of cells of same dimension. The (possible) precondition of the initial operation has to be satisfied by each cell.*

Let C^i be a set of i-cells to remove (resp. contract). Let $B^{SI} = C^i\alpha_i - C^i$. Darts for which α_i is modified are those belonging to B^{SI}. The definition of α'_i for these darts is: $\forall b \in B^{SI}$, $b\alpha'_i = b(\alpha_i\alpha_{(i+1)})^k\alpha_i$ (resp. $b\alpha'_i = b(\alpha_i\alpha_{(i-1)})^k\alpha_i$) where k is the smallest integer such that

$b(\alpha_i\alpha_{(i+1)})^k\alpha_i \in B^{SI}$ (resp. $b(\alpha_i\alpha_{(i-1)})^k\alpha_i \in B^{SI}$). The proof that we get a G-map (k exists, it is unique, α'_i is an involution, $\alpha'_j\alpha'_k$ is an involution for $k \neq j-1, j+1$) is similar to the previous initial operation.

Moreover, removing a set of i-cells or applying simultaneously and in any order the initial operation for any contracted cell, produces the same result. Indeed, when we apply the generalized operation on a set of cells, we have: $\forall b \in B^{SI}$, $b\alpha'_i = b(\alpha_i\alpha_{(i+1)})^k\alpha_i$. The darts of this path can be partitioned, depending on removed cells they belong to, i.e. $b\alpha'_i = b(\alpha_i\alpha_{(i+1)})^{k_1}(\alpha_i\alpha_{(i+1)})^{k_2}\dots(\alpha_i\alpha_{(i+1)})^{k_p}\alpha_i$ (it is possible that darts corresponding to different subpath belong to the same cell). Removing one of these cells implies the redefinition of α_i, and the corresponding path in the new G-map is deduced from the initial one by removing the subpathes corresponding to the removed cell. When all cells have been removed successively, in any order, we get $b\alpha'_i$. This proof is exactly the same for the contraction operation.

Generalisation 2 *The previous generalization can be extended for simultaneously removing and contracting cells of same dimension. A cell is either removed or contracted, but not both at the same time. The (possible) precondition of the initial operation has to be satisfied by each cell.*

More precisely, let CS^i (resp. CC^i) be a set of i-cells to remove (resp. contract), such that $CS^i \cap CC^i = \emptyset$ and such that the (possible) precondition of i-removal (resp. i-contraction) operation is satisfied for each cell of CS^i (resp. CC^i). Let $B^{SI} = (CC^i \cup CS^i)\alpha_i - (CC^i \cup CS^i)$. As before, α_i is redefined for these darts: $\forall b \in B^{SI}$, $b\alpha'_i = b' = b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_p})\alpha_i$ where p is the smallest integer such that $b' \in B^{SI}$ and $\forall 1 \leq j < p$, if $b_c = b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_{j-1}})\alpha_i \in CS^i$ then $k_j = i+1$ else if $b_c \in CC^i$ then $k_j = i-1$.

Intuitively, for each dart $b \in B^{SI}$, we cover the removed or contracted cells, starting from $b\alpha_i$, by applying either $(\alpha_{(i+1)}\alpha_i)$ if the current dart belongs to a removed cell, or $(\alpha_{(i-1)}\alpha_i)$ if it belongs to a contracted cell. The coverage is done when the current dart doesn't belong to a removed or contracted cell: this current dart is $b' = b\alpha'_i$.

α'_i is well defined, i.e. for each dart b of B^{SI}, it exists an unique dart $b' = b\alpha'_i$. If it is not the case, then a cycle exists, starting from b, leading to an infinite coverage of the darts of removed and contracted cells. Since the set of darts is finite, it exists n and m such that $b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_n}) = b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_m})$ and $m < n$. Let n be the smallest integer satisfying this relation.

- If $k_n = k_m$, then $b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_{n-1}}) = b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_{m-1}})$ since α_i and α_{k_n} are involutions. So n is not the smallest integer satisfying the relation.

- Thus $k_n \neq k_m$, and darts $b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_{n-1}})\alpha_i$ and $b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_{m-1}})\alpha_i$ belong to a same i-cell, that has to be simultaneously removed and contracted, contrary to the hypothesis.

α'_i is an involution, since $b\alpha'_i = b' = b(\alpha_i\alpha_{k_1})\dots(\alpha_i\alpha_{k_p})\alpha_i$ and $b'\alpha'_i = b'(\alpha_i\alpha_{k_p})\dots(\alpha_i\alpha_{k_1})\alpha_i$: the two paths are opposite to each other, i.e. the same contracted or removed darts are reached, but in reverse order.

The last property of G-maps (i.e. α_j and α_k commutes when $j \neq k-1, k+1$), is proved as for the initial operation by the definition of G-maps and by the (possible) precondition of the operation. At last, removing and contracting simultaneously a set of cells, or applying the initial operation successively for all cells, in any order, produce the same G-map. The proof is similar to that of generalization 1.

Generalisation 3 *The previous definition can be extended for the removal and/or contraction of a set of disjoined cells of any dimension. The (possible) precondition of the initial operation has to be satisfied for each cell.*

Definition 5 (Simultaneous removal and contraction of cells of any dimension)
Let $G = (B, \alpha_0, \ldots, \alpha_n)$ be an n-G-map, $CS^0 \ldots CS^{n-1}$ be sets of 0-cells ... $(n-1)$-cells to be removed and $CC^1 \ldots CC^n$ be sets of 1-cells ... n-cells to be contracted. Let $CC = \cup_{i=1}^{n} CC^i$ and $CS = \cup_{i=0}^{n-1} CS^i$. Two preconditions have to be satisfied: cells are disjoined (i.e. $\forall C, C' \in CC \cup CS$, $C \cap C' = \emptyset$), and "the degree of each cell is equal to 2", i.e.:
- $\forall i, 0 \leq i \leq n-2$, $\forall b \in CS^i$, $b\alpha_{(i+1)}\alpha_{(i+2)} = b\alpha_{(i+2)}\alpha_{(i+1)}$
- $\forall i, 2 \leq i \leq n$, $\forall b \in CC^i$, $b\alpha_{(i-1)}\alpha_{(i-2)} = b\alpha_{(i-2)}\alpha_{(i-1)}$
Let $B^{Si} = (CS^i \cup CC^i)\alpha_i - (CS^i \cup CC^i)$ $\forall i, 0 \leq i \leq n$. The resulting n-G-map is $G' = (B', \alpha'_0, \ldots, \alpha'_n)$ defined by:

- $B' = B - (CC \cup CS)$;
- $\forall i, 0 \leq i \leq n$, $\forall b \in B' - B^{Si}$, $b\alpha'_i = b\alpha_i$;
- $\forall i, 0 \leq i \leq n$, $\forall b \in B^{Si}$, $b\alpha'_i = b' = b(\alpha_i\alpha_{k_1}) \ldots (\alpha_i\alpha_{k_p})\alpha_i$, *where p is the smallest integer such that $b' \in B^{Si}$, and $\forall 1 \leq j < p$, if $b_c = b(\alpha_i\alpha_{k_1}) \ldots (\alpha_i\alpha_{k_{j-1}})\alpha_i \in CS^i$ then $k_j = i+1$ else ($b_c \in CC^i$) $k_j = i-1$.*

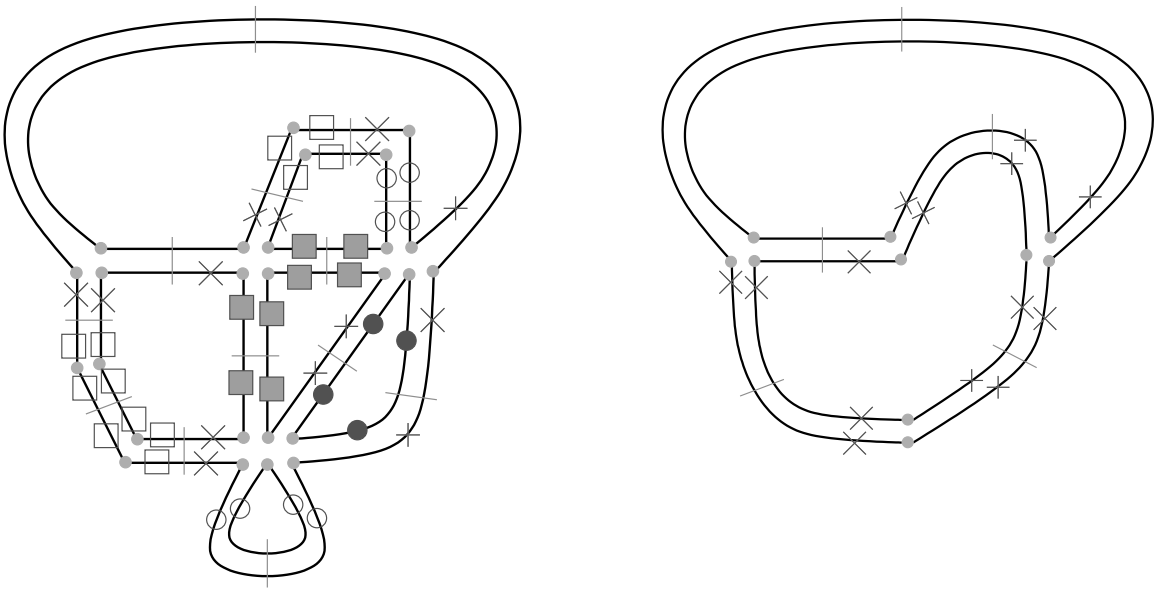
This definition covers all the previous operations. This general operation can be applied if each dart belongs either to a removed cell or a contracted cell, and when each cell satisfies the possible precondition of the initial operation. As for the previous generalizations, α_i is redefined only for the darts of B^{Si}, but this redefinition is now done for any i, $0 \leq i \leq n$.

Theorem 2 *The general removal and contraction operation produces an n-G-map.*

Proof 2 *Let C_1 and C_2 be two disjoined cells of respective dimensions d_1 and d_2 such that $d_1 \neq d_2$. Let $B^{SC_1} = C_1\alpha_{d_1} - C_1$: If b exists such that $b \in B^{SC_1} \cap C_2$, then $b\alpha_{d_1} \in C_1$ (def. of B^{SC_1}) and $b\alpha_{d_1} \in C_2$ (since $b \in C_2$, and $d_1 \neq d_2$): contradiction with the precondition $C_1 \cap C_2 = \emptyset$. So $b \in B^{SC_1} \cap C_2 = \emptyset$. So, any dart belonging to B^{Si} can't belong to a removed or contracted cell.*

For any i, α_i is redefined only for the darts of B^{Si}, by covering removed or contracted i-cells. Since involutions are not modified for darts of removed or contracted cells, for each dart b of B^{Si}, $b\alpha'_i$ corresponds to an unique path in the original G-map, that can not be modified by other redefinitions (i.e. all redefinitions are independent for each other). We can prove, as before, that this operation produces a G-map. Moreover, for the same reason, this resulting G-map can also be constructed by successive removals and contractions of the cells of $CC \cup CS$, applied in any order. □

An example of this last generalization is described in Figure 12. Figure 12.*a* shows a 2-G-map. Darts belonging to a contracted 1-cell (resp. contracted 2-cell, removed 0-cell, removed 1-cell) are marked with a circle (resp. a disk, an empty square, a filled square). Darts marked with crosses belong to $\cup B^{Si}$. One or several involutions are redefined for these darts. This example uses simultaneously all possible removal and contraction operations. Figure 12.*b* shows the



a. 2-G-map before operations. b. The resulting 2-G-map.

Figure 12: One example of simultaneously removal and contraction of several cells of different dimensions.

resulting 2-G-map. We can check that we get the same result when all cells are successively removed or contracted by the initial operation.

6 Conclusion and perspectives

In this paper, we have defined the removal and contraction operations of any dimensional cells in any dimensional space. Moreover, we have studied how to perform simultaneously a set of different operations. These definitions are homogeneous for any dimension (for dimension i, we mainly use $\alpha_{(i+1)}$ for removal and $\alpha_{(i-1)}$ for contraction). Since combinatorial maps [4, 7] can be easily deduced from orientable generalized maps [20], these operations can also be defined on combinatorial maps.

We have a particular interest in applying this work for image representations handling. On one hand, images satisfy particular topological properties that limit the possible configurations for these operations. On the other hand, removing $(n-1)$-cells (resp. contracting 1-cells) is generally distinguished from other removal or contraction operations. During a simplification

process, $(n-1)$-cells (resp. 1-cells) are chosen (many methods have been proposed, for instance for split and merge algorithms); the removal or contraction of other cells "clean" the resulting representation, in order to avoid the representation of redundant topological information. For instance in figure 12.*b*, the two vertices of degree 2 can be removed without loss of topological information.

We intend now to study how to define pyramids of any dimensional generalized maps with removal and contraction operations. We are also studying the evolution of some topological characteristics, in order to control the construction of coherent pyramids (for instance to control the connectivity). Some results are presented in [12] in the particular framework of 2D and 3D images representation; they have to be generalized in upper dimension and for the general case.

Another interesting perspective concerns the parallelization of the application of a set of operations. We think that, in the general case, checking preconditions can be distributed on each concerned cell: it is then possible to simultaneously compute sets $C\alpha_i - C$; the application of the operations can be distributed on the surviving darts (but this has to be more deeply studied). Parallelization has also to be studied for particular cases, for instance for controlling topological evolutions, or when removed or contracted cells satisfy some particular properties (a well known example consists in removing a tree of edges).

Acknowledgements

We wish to thank Éric Andres for useful comments and careful reading of this paper.

References

[1] Y. Bertrand, G. Damiand, and C. Fiorio. Topological encoding of 3d segmented images. In *Discrete Geometry for Computer Imagery*, number 1953 in Lecture Notes in Computer Science, pages 311–324, Uppsala, Sweden, december 2000.

[2] Y. Bertrand, G. Damiand, and C. Fiorio. Topological map: minimal encoding of 3d segmented images. In *Workshop on Graph based representations*, pages 64–73, Ischia, Italy, may 2001. IAPR-TC15.

[3] J.P. Braquelaire, P. Desbarats, and J.P. Domenger. 3d split and merge with 3-maps. In *Workshop on Graph based representations*, pages 32–43, Ischia, Italy, may 2001. IAPR-TC15.

[4] J.P. Braquelaire, P. Desbarats, J.P. Domenger, and C.A. Wüthrich. A topological structuring for aggregates of 3d discrete objects. In *Workshop on Graph based representations*, pages 193–202, Austria, may 1999. IAPR-TC15.

[5] E. Brisson. Representing geometric structures in d dimensions: topology and order. *Discrete Comput. Geom.*, 9(1):387–426, 1993.

[6] L. Brun. *Segmentation d'images couleur à base Topologique*. PhD thesis, Université de Bordeaux I, december 1996.

[7] L Brun and W.G. Kropatsch. Dual contraction of combinatorial maps. In *Workshop on Graph based representations*, Austria, may 1999. IAPR-TC15.

[8] L Brun and W.G. Kropatsch. Pyramids with combinatorial maps. Technical report 57, Institute of Computer Aided Design, Vienna University of Technology, Austria, december 1999. URL: http://www.prip.tuwien.ac.at/.

[9] L Brun and W.G. Kropatsch. The construction of pyramids with combinatorial maps. Technical report 63, Institute of Computer Aided Design, Vienna University of Technology, Austria, june 2000. URL: http://www.prip.tuwien.ac.at/.

[10] L Brun and W.G. Kropatsch. Contraction kernels and combinatorial maps. In *Workshop on Graph based representations*, pages 12–21, Ischia, Italy, may 2001. IAPR-TC15.

[11] R. Cori. *Un code pour les graphes planaires et ses applications*. PhD thesis, Université de Paris VII, 1973.

[12] G. Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. PhD thesis, Université de Montpellier II, december 2001.

[13] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.

[14] H. Elter. *Etude de structures combinatoires pour la représentation de complexes cellulaires*. PhD thesis, Université Louis-Pasteur, Strasbourg, september 1994.

[15] C. Fiorio. A topologically consistent representation for image analysis: the frontier topological graph. In *Discrete Geometry for Computer Imagery*, number 1176 in Lecture Notes in Computer Science, pages 151–162, Lyon, France, november 1996.

[16] A. Jacques. Constellations et graphes topologiques. In *Combinatorial Theory and Applications*, volume 2, pages 657–673, 1970.

[17] W.G. Kropatsch. Building irregular pyramids by dual graph contraction. Technical report PRIP-TR-35, Dept. for Pattern Recognition and Image Processing, Institute for Automation, Technical Univerity of Vienna, Austria, july 1994.

[18] W.G. Kropatsch. Building irregular pyramids by dual-graph contraction. *Vision, Image and Signal Processing*, 142(6):366–374, december 1995.

[19] P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Commputer Aided Design*, 23(1):59–82, 1991.

[20] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3):275–324, 1994.

Equivalence Between Order and Cell Complex Representations

Alayrangues Sylvie, Lachaud Jacques-Olivier
LaBRI
351, cours de la libération, 33405 TALENCE
tel : (+33) 5 56 84 69 00 - fax : (+33) 5 56.84.66.69
e-mail: Sylvie.Alayrangues@labri.fr, Jacques-Olivier.Lachaud@labri.fr

Abstract

In order to define consistent models and algorithms for image analysis, many topological representations of images have been proposed. Unfortunately the most generic ones are often not explicitly related, and properties exhibited on one representation are unknown for other representations. The aim of this paper is to show how two different topological representations of images, namely the *order* representation (developped by Bertrand *et al.*) and the *complex representation using strong weak lighting functions* (studied by Dominguez *et al.*) may be related in such a way that the results and algorithms proved on one may be applied to the other and conversely.

1 Introduction

The study of the topological properties of discrete spaces is an active field of research. Different approaches have been proposed to represent the support of the images and describe the links between their elements (e.g. pixels in 2D, voxels in 3D). Most of their results and algorithms seem to be model dependent. Nevertheless there are often bridges between these approaches which once discovered could allow the transfer of theoretical results and practical algorithms from one to another. This paper takes an interest in two of these approaches and aims to explicit their relationship. The first one is a work conducted by Bertrand *et al.* [6, 7] and use the *order representation*. From this point of view the elements of the image are represented by the "smallest" elements relative to the order whereas the other elements define the links between them. The second one is based on the approach of Dominguez *et al.* [3, 4, 5] which considers the support of an image as a *finite polyhedral complex* equipped with a particular function called *strong weak lighting function*. In this case the elements of the image are represented by the cells of maximal dimension and the cells of lesser dimension connect them. Both models propose formal ways to define several local connectedness relations between the elements of the image, that is several methods to select only the links of interest (for a given purpose) between

the elements of the image. These two approaches are compliant with classical connectedness relations (for example those introduced by Rosenfeld) and also create new ones. Our contribution is to show that, one can construct complexes from orders and orders from complexes in such a way that the results found by Bertrand *et al.* and by Dominguez *et al.* hold for both models. We first present the main characteristics of both orders and abstract cell complexes and the ways to go from one to the other. We then explicit the relations between their topologies and finally show the correspondence between the construction of connectedness on orders and on complexes equipped with a strong weak lighting function. We will finally list briefly the advantages each model may take of the other.

2 Order and Complexes

2.1 Definitions

This section presents some definitions related to orders and complexes and introduces mappings for building complexes from orders and orders from complexes. For the orders we follow the notations defined by Bertrand *et al.* in [6, 7].

An *order* is a pair $|X| = (X, \alpha)$, where X is a set and α a reflexive, antisymetric, and transitive binary relation. In the sequel we denote β the inverse of α and θ the union of α and β. Moreover we only consider a restricted family of orders, called *CF orders*, that are *countable*, i.e. X is countable, and *locally finite*, i.e. $\forall x \in X, \theta(x)$ is finite.

Some particular sets can be associated to each element x of X. The simplest ones are its α*-adherence*, $\alpha(x) = \{y \in X; (x, y) \in \alpha\}$ and its *strict* α*-adherence*, $\alpha^{\square}(x) = \alpha(x) \backslash \{x\}$. We also call α*-chain* every fully α-ordered subset of $|X|$. If $|X|$ is CF, every α-chain is finite and its length is equal to the number of its elements less one. If we only consider a subset S of X we denote $\alpha_{|S}$ the reflexive, antisymetric, and transitive binary relation $\alpha \cap S \times S$.

An *abstract cell complex* $C = (E, <, dim)$ is a set E of abstract elements associated with an irreflexive, antisymetric, and transitive binary relation $< \subseteq E \times E$ called *border (or face) relation* and a mapping *dimension* dim : $E \rightarrow I \subseteq \mathbb{Z}^+$ such that $\forall (e, e') \in E \times E$, with $e' < e$, $dim(e') < dim(e)$.

In the sequel, the *star* of a cell e of C will be denoted $st(e)$ and its *combinatorial closure* $cl(e)$.

2.2 Associated orders and complexes

It may be easily seen that there exists a $1-1$ mapping f that builds an order $|X_C| = (X_C, \alpha_C)$ from an abstract cell complex $C = (E, <, dim)$ such that $\forall e \in E$, $\exists! f(e) \in X_C$ and $\forall (e, e') \in E \times E$ with $e < e'$ or $e = e'$, $(f(e), f(e')) \in \alpha_C$. This transformation loses, in general, the notion of dimension. Proving the existence of a mapping that constructs an abstract cell complex from an order and effectively building one is less trivial since orders have no notion of dimension. We first restrain our study to CF-orders since non countable orders cannot be

mapped onto a complex. The intuition is to partition X in a finite number of subsets and to attribute to each element a number according to the unique subset it belongs to.

As the order is locally finite, there exists a subset X_0 of X, whose elements have an empty strict α-adherence. Moreover all other elements of X are linked by at least one α-chain to an element of X_0 and the α-chains between an element of X_0 and another element of X have a finite length. There exists then an integer number k such that the length of every α-chain on $|X|$ is less than or equal to k. Moreover for each $x \in X$ there exists an integer $i \leq k$ such that i is the maximal length of all the α-chains beginning at x and ending at an element of X_0. X may then be partitionned in X_i, $i = 0..k$ such that $x \in X_i$ if the maximal length of the α-chains from X_0 to x is i. This partition of X is called α-*decomposition* of $|X|$ and is indeed the family $\mathcal{F} = \{X_0, X_1, ..., X_k\}$ such that :
(i) $X_0 = \{x \in X,\ \alpha^{\Box}(x) = \emptyset\}$,
(ii) $\forall i \in \{1, \dots, k\}, X_i = \{x \in S, S = X \setminus \bigcup_{j=0}^{j=i-1} X_j, \alpha^{\Box}_{|S}(x) = \emptyset\}$,
(iii) $X_k \neq \emptyset$ and $X = \bigcup_{i=0}^{i=k} X_i$.

For each $x \in X$, there exists then $i \in [0..k]$ such that $x \in X_i$. Let x_i and x_j be respectively elements of X_i and X_j such that $x_j \in \alpha^{\Box}(x_i)$ then we deduce from the construction of the partition that $i > j$. This means that an element of $|X|$ "less" than another according to α is associated to a lesser integer number. We call therefore this number the dimension of x and denote it $dim_\alpha(x)$. The dimension of each element of $|X|$ may be recursively computed.

Property 1 : Let x be an element of $|X|$,
- $dim_\alpha(x) = 0\ iff\ \alpha^{\Box}(x) = \emptyset$
- $dim_\alpha(x) = 1 + \max_{i=1}^{m}(dim_\alpha(y_i))$ with $\alpha^{\Box}(x) = \{y_i, i \in [1..m]\}$

In order to make easier the understanding of the different notions defined on CF-orders and their correspondence with the ones defined on complexes, we choose to represent CF-orders as simple directed acyclic graphs and abstract cell complexes as a particular kind of polyedral complexes. The elements of a CF-order (i.e. elements of X) will be represented by the nodes of the graph. An element x' of X belonging to $\alpha(x)$ will be represented by a node belonging to the subgraph rooted at the node representing x. Moreover its depth in this subgraph will be equal to the maximum depth of the subgraph less $dim_\alpha(x')$. The figure 1 shows an example of the α-decomposition of an order.

Thanks to the definition of dim_α we are able to construct a function that builds an abstract cellar complex from a CF-order (see figure 2).

Theorem 1 (Order and abstract cell complex) : The *abstract cell complex* $C = (E, <, dim)$, associated with the CF-order $|X| = (X, \alpha)$, is defined by the map ψ such that :
- $\forall x \in X$, $\exists! \psi(x) \in E$ (ψ 1-1 mapping from X to E),
- $\forall (x, x') \in X \times X$ such that $x' \in \alpha^{\Box}(x)$, $\psi(x') < \psi(x)$ (ψ isomorphism from $(X, \alpha^{\Box})$ into $(E, <)$),
- $\forall x \in X$, $dim(\psi(x)) = dim_\alpha(x)$

In such a complex, the faces of a cell e are precisely the images by ψ of the elements of the

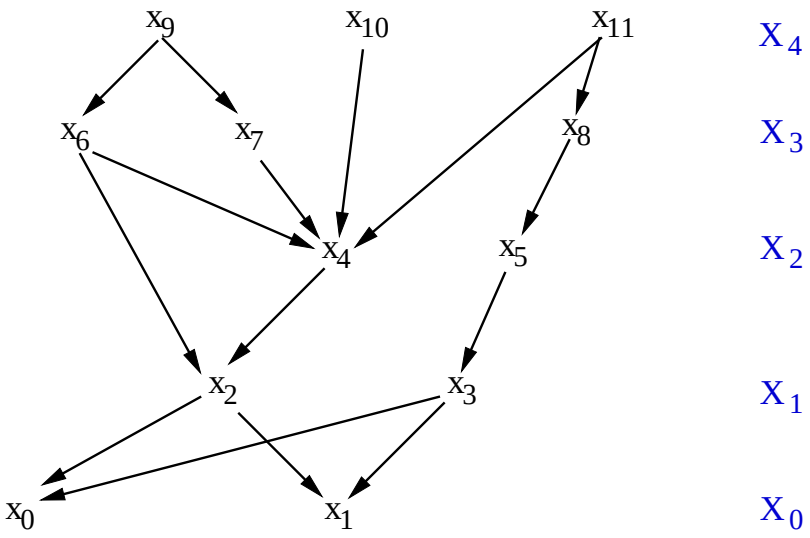


Figure 1: Representation of an order and its decomposition

strict α-adherence of $\psi^{-1}(e)$. In the sequel we will be interested in another complex that may be built from a CF-order. We call it the *dual abstract cell complex* of the order (see figure 2).

Theorem 2 (Order and dual abstract cell complex) : The *dual abstract cell complex* $C^* = (E^*, <^*, dim^*)$, associated with the CF-order $|X| = (X, \alpha)$, is defined by the map ψ^* such that :
- $\forall x \in X$, $\exists! \psi^*(x) \in E^*$ (ψ^* 1-1 mapping from X to E^*),
- $\forall (x, x') \in X \times X$ such that $x' \in \beta^{\square}(x)$, $\psi^*(x') <^* \psi^*(x)$ (ψ^* isomorphism from $(X, \beta^{\square})$ into $(E^*, <^*)$),
- $\forall x \in X$, $dim(\psi^*(x)) = dim^*_\alpha(x)$ with $dim^*_\alpha = max_{x \in X}\{dim_\alpha(x)\} - dim_\alpha$.

The dual abstract cell complex of an order (X, α) is generally *not the same* as the abstract cell complex of the dual order (X, β). The dimension dim^*_α is indeed different from dim_β in most cases. We prefer using the dual abstract cell complex of an order because it is, by construction, a *pure*[1] complex. In such a complex, the faces of a cell e are precisely the images by ψ^* of the elements of the strict β-adherence of $\psi^{*^{-1}}(e)$.

Finally, with an informal notation, we remark that a n-complex C such that $\psi(\psi^{-1}(C)) = C$ (resp. $\psi^*(\psi^{*^{-1}}(C)) = C$) must have the following property : every i-cell of C ($i \in [0..n]$) has at least a k-face for each $k \in [0, i-1]$ (resp. is face of a k-cell, $k \in [i+1, n]$). The order built with ψ^{-1} (resp. $\psi^{*^{-1}}$) from such a complex keeps implicitly the information of dimension for its elements : the ψ^{-1}-image (resp $\psi^{*^{-1}}$-image) of each i-cell is an element of X_i (resp X_{n-i}).

[1] a pure n-complex is a n-complex whose k-cells, $k < n$, are faces of at least one n-cell

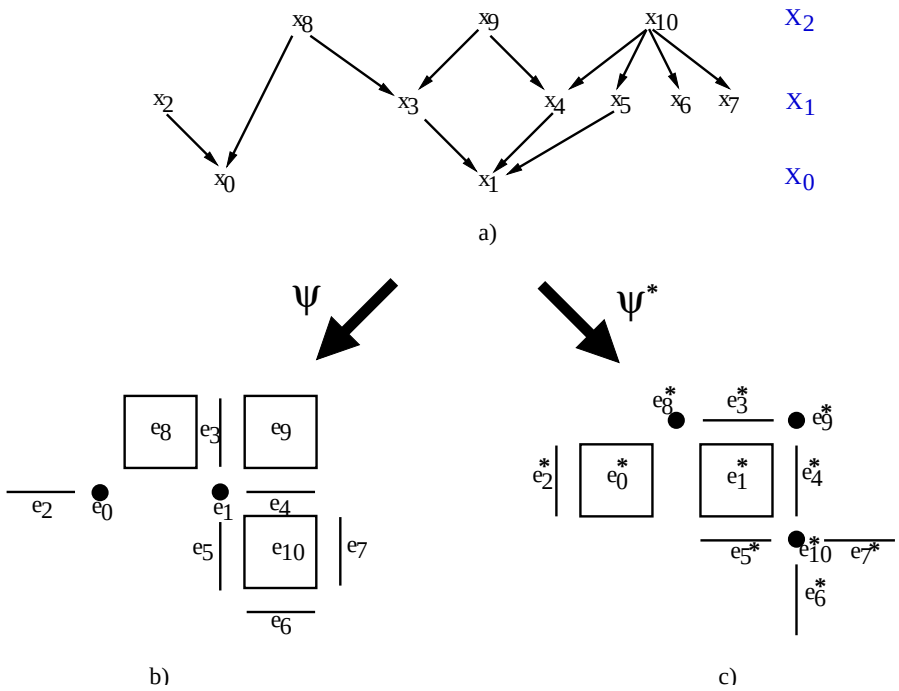


Figure 2: a) Graph-representation of an order $|X|$. b) Abstract complex associated to $|X|$. c) Dual abstract complex associated to $|X|$.

2.3 Restriction to particular complexes/orders

In most cases the notion of abstract cell complex is too general to represent the support of images. We deal with particular complexes (such as unrestricted simplicial and polyhedral complexes). We would like then to know how to characterize orders so that ψ or ψ^* builds a suitable complex. There are ways to characterize *simplicial* orders (i.e. orders such that its abstract cell complex is simplicial). It is not clear whether we can define *polyhedral orders*, because the definition of polyhedral complexes involve geometric constraints. In the sequel, we will be interested by a wider kind of complexes. We will call them *strongly normal* complexes[2].

Intuitively a complex is said *strongly normal* if the combinatorial frontier between two maximal cells of the complex is either empty or path connected. Formally, a complex C is said *strongly normal* if for each cell $e \in C$, the set of the maximum cells containing e in their combinatorial closures, that is the set $\{e_i \in st(e)\backslash\{e\} \ / \ st(e_i)\backslash\{e_i\} = \emptyset\}$, is finite and if the intersection of the combinatorial closures of the e_i is either empty (iff $st(e)\backslash\{e\} = \emptyset$) or the combinatorial closure of a cell of C.
We define similarly the notion of *strongly normal* order, and will prove that the dual abstract cell complex of such an order is a strongly normal one.
An order is said *strongly normal* if it is CF and if the intersection of the β-adherences of the

[2]unrestricted simplicial complexes and unrestricted polyhedral complexes are special cases of strongly normal complexes

elements of every subset of X_0 is either empty or equal to the β-adherence of an element $x \in X$.

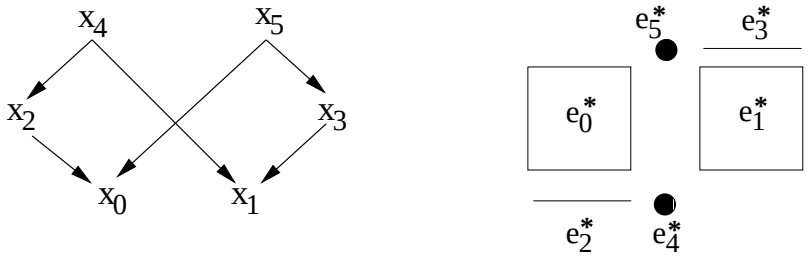


Figure 3: Example of a CF-order and its dual abstract cell complex that are not strongly normal

Lemma 1 : A *pure* n-complex C is strongly normal if for each cell $e \in C$, $dim(e) < n$, the *set of* n*-cells*, which belong to $st(e)$, is finite and if the intersection of their combinatorial closures is the combinatorial closure of a cell of C.

Theorem 3 (strongly normal order and complex) : The dual abstract cell complex of a strongly normal order is a strongly normal complex.

Proof : *Let $|X|$ be a strongly normal order. The dual abstract cell complex $C^*_{|X|}$ of $|X|$ is pure. The order is CF and then the star of each cell of $C^*_{|X|}$ is finite. Moreover, let e be a cell of $C^*_{|X|}$, ${\psi^*}^{-1}(e) \in X$, $|X|$ is strongly normal so $\exists x' \in X$ such that $\beta(x') = \bigcap\{\beta(x_0),\ x_0 \in \alpha({\psi^*}^{-1}(e)) \cap X_0\}$. Considering $C^*_{|X|}$, it means that $\exists e' = \psi^*(x')$ such that $cl(\psi^*(x')) = \bigcap\{cl(e_n), e_n \in \psi^*(X_0)\ with\ e < e_n\}$. From Lemma 1, we can deduce that $C^*_{|X|}$ is strongly normal.* □

2.4 Links between their topologies

The notions defined on orders and complexes are linked through the mapping ψ.
We first describe those that allow to define a topology on an order and its associated complexes. We consider an order $|X|$ and its abstract cell complex $C_{|X|}$. The image of the α-*adherence* of an element x (resp. a subset S) of X is the *combinatorial closure* of $\psi(x)$ (resp. $\psi(S)$). The image of the *strict* α-*adherence* of an element x (resp. a subset S) of X is the *combinatorial frontier* of $\psi(x)$ (resp. $\psi(S)$).

The α-*interior* of a subset S of $|X|$ is the set : $\star\alpha(S) = \overline{\alpha(\overline{S})} = \{x \in S / \beta(x) \subseteq S\}$. The image of the α-interior of a subset S of $|X|$ is the set of cells $\psi(x) \in \psi(S)$ whose star in $C_{|X|}$ belongs to $\psi(S)$. A subset S of $|X|$ is α-*closed* if $S = \alpha(S)$, and α-*open* if $S = \star\alpha(S)$.

The notions of open and closed subcomplex are the ones usually used (see [2]). With these definitions, a *discrete topology in the sense of Alexandroff*[3] may be defined on both orders and

[3] A topology is said discrete in the sense of Alexandroff iff the intersection of every family of open sets (finite or infinite) is an open set

complexes. Equipped with this topology, orders and complexes become *Alexandroff spaces* [4]. Moreover the image of an α-closed (resp. α-open) subset of $|X|$ is a closed (resp. open) subcomplex of $C_{|X|}$. The inverse is also true. ψ is hence an *homeomorphism* between $|X|$ and $C_{|X|}$.
We examine then an order $|X|$ and its dual abstract cell complex $C^*_{|X|}$. The image of the α-*adherence* of an element x (resp. a subset S) of X is the *combinatorial star* of $\psi^*(x)$ (resp. $\psi^*(S)$). The image of the *strict* α*-adherence* of an element x (resp. a subset S) of X is the *strict combinatorial star* of $\psi^*(x)$ (resp. $\psi^*(S)$). The image of the α-interior of a subset S of $|X|$ is the combinatorial closure of $\psi^*(S)$. Moreover the image of a α-closed (resp. α-open) subset of $|X|$ is an open (resp. closed) subcomplex of $C^*_{|X|}$. ψ^* is not an homeomorphism between $|X|$ and $C^*_{|X|}$.

In the sequel we call θ*-path* on $|X|$ the image by ψ^{-1} or ${\psi^*}^{-1}$ of a path (defined by the border relation) on $C_{|X|}$ or $C^*_{|X|}$, i.e. a sequence of elements such that every couple of consecutive elements is in $\alpha \cup \beta$.

3 Connectedness between image elements

We consider now the notions and properties that are linked with the connectedness of orders and complexes. It may first be proved that *connectedness* and *path-connectedness*[5] are equivalent on both orders and complexes. In the sequel we determine which elements of an order $|X|$ (resp. a complex C equipped with a strong weak lighting function) must be kept to grant a given connectivity between the α-terminals of $|X|$(resp. its n-cells of C). We will call "inessential" elements of either $|X|$ or K, which are not needed to characterize the chosen connectedness between the main elements. For each notion introduced in one of the models we give an interpretation into the other. Finally we prove the equivalence between the definition of inessential elements in the two models when the considered order and the associated pure n-complex are *strongly normal*.

3.1 Connectedness between α-terminals in an order

As said in introduction, the order approach considers the elements in X_0, i.e. the α-terminals of $|X|$, as the points of the image and the elements of the X_i,for $i > 0$, as the connections between them. The images of the α-terminals in $C_{|X|}$ are the cells without faces, i.e. by construction of $C_{|X|}$, the 0-cells, and their images in $C^*_{|X|}$ are the cells with an empty strict star, i.e. by construction of $C^*_{|X|}$, the n-cells where n is the maximal dimension of $C^*_{|X|}$. Determining whether an element of $|X|$ is inessential or not consists in a local observation.
We are then interested in the α*-closeness* of each element x of $|X|$, $\alpha^\bullet(x)$. Formally, $\alpha^\bullet(x) = \{y \in X / y \in \alpha^\square(x), \alpha^\square(x) \cap \beta^\square(y) = \emptyset\}$, i.e. it is the set of elements of $|X|$ that are linked to

[4] An Alexandroff space is a $\mathcal{T}_0$-separable space with a discrete topology in the sense of Alexandroff
[5] Path-connectedness on orders means θ-path-connectedness

x by a single α-chain of length 1. In our graph representation, the α-closeness of an element is precisely represented by its direct childs. And it may be proved that :

$$\alpha^{\square}(x) = \bigcup_{x_i \in \alpha^{\bullet}(x)} \alpha(x_i).$$

The image of $\alpha^{\bullet}(x)$ by ψ is the smallest subset S of the combinatorial frontier of $\psi(x)$ whose combinatorial closure is equal to the combinatorial frontier of $\psi(x)$. Otherwise said it is the smallest set of cells that allows to uniquely determine the combinatorial frontier of $\psi(x)$. The image of $\alpha^{\bullet}(x)$ by ψ^* is the smallest subset S^* of the strict star of $\psi^*(x)$ whose star is equal to the star of $\psi^*(x)$. Otherwise said it is the smallest set of cells that allows to uniquely determine the strict star of $\psi^*(x)$.

The elements whose α-closeness contains one and only one element (see figure are called α-*unipolar* (we note that an α-terminal cannot be α-unipolar). These points are the simplest inessential points in the order, if x is α-unipolar then $\exists x' \in \alpha^{\square}(x)$ (namely $\{x'\} = \alpha^{\bullet}(x)$) such that $\alpha^{\square}(x) = \alpha(x')$. The other inessential points of the orders are called k-α-*free* and are α-unipolar for the order obtained after the recursive deletion of a sequence of α-unipolar points (or 0-α-free points) and i-α-free points ($i \in \{1, \ldots, k-1\}$). Their deletion do not disconnect any couple of α-terminals of their common adherences. To prove this we first show how the strict α-adherence of any k-α-free element ($k \geq 0$) may be decomposed.

Lemma 1 : Let $|X|$ be a CF-order, if $x \in |X|$ is α-free then there exists $x' \in \alpha^{\square}(x)$ such that $\alpha(x) \cap X_0 = \alpha(x') \cap X_0$.

This lemma may be proved thanks to a suitable decomposition of the strict α-adherence of the α-free element (see [1]).

The deletion of the images of α-free elements in $C_{|X|}$ do not remove the shortest θ-paths whose elements have the smaller dimension between any two 0-cells of their neighborhood. The suppression of their images in $C^*_{|X|}$ do not delete the shortest θ-paths whose elements have the higher dimension between any two n-cells of their neighborhood.
Finally an element of $|X|$ that is not α-free is called α-*link* and the set consisting of all the α-links of an order is called the α-*kernel* of this order.

3.2 Strong Weak Lighting Functions

The weak lighting functions introduced by Dominguez *et al.* are usually defined on a homogeneously n-dimensional locally finite polyhedral complex K, but the formal definition is valid on a wider range of complexes. These functions allow to "light" the cells required to define a chosen connexity on the complex.
In the sequel we denote by O a subset of n-cells of K, by $st_n(e, O)$ the set of n-cells in O belonging to $st(e)$, with $e \in K$.

We denote by $supp(O)$ the support of O that is the set of cells e whose combinatorial closure is equal to the intersection of the combinatorial closure of the elements of $st_n(e, O)$. We define

on a complex K a *Strong Weak Lighting Function* (s.w.l.f.) any map,
f : $\mathcal{P}(cell_n(K)) \times K \rightarrow \{0,1\}$ with $\forall O \in \mathcal{P}(cell_n(K))$ and $e \in K$, such that :
1. if $e \in O$ then $f(O,e) = 1$;
2. if $e \notin supp(O)$ then $f(O,e) = 0$;
3. $f(O,e) \leq f(cell_n(K),e)$;
4. $f(O,e) = f(st_n(e,O),e)$;

The first property of the s.w.l.f. specifies that all the n-cells of an objet are lighted, the second one expresses that cells that are not intersection of combinatorial closures of subsets of O are not useful to connect it, the third one imposes that a cell lighted for an object is also lighted for the whole image, and the fourth one induces that for a given object, the lighting of a cell is a local property of the objet. Many s.w.l.f. may be constructed on a complex depending on which connexity we want to associate to the complex. These functions are equal to 0 on inessential elements of the complex and to 1 on the others.

3.3 Connectedness in strongly normal orders/complexes

We are going to prove that it is possible to build a fonction on a strongly normal order $|X|$ that will define a s.w.l.f. on its dual abstract cell complex $C_{|X|}$. To do so, we first express some useful lemma (their proofs are given in [1]).

The first one indicates that the strong normality property is hereditary for some subcomplex of a strongly normal complex.

Lemma 2 : Let C be a pure strongly normal complex, every subcomplex of C built without removing any cell belonging to the support of at least one set of n-cells, is also pure and strongly normal.

The next lemma means that if two different cells are faces of exactly the same n-cells and if one of them has a lower dimension then this latter cell is inessential.

Lemma 3 : Let $|X|$ be a CF-order and $C^*_{|X|}$ its dual abstract cell complex, if x and x' are such that $\alpha(x) \cap X_0 = \alpha(x') \cap X_0 = S_0$ and $x' \in \alpha^{\square}(x)$ then $\psi^*(x) \notin supp(\psi^*(S_0))$

The following lemma is a corollary of Lemma 3 , it shows that the images of the α-unipolar elements of $|X|$ in $C^*_{|X|}$ are inessential.

Lemma 4 : Let $|X|$ be a CF-order and $C^*_{|X|}$ its dual abstract cell complex, if x is α-unipolar then $\psi^*(x) \notin supp(\psi^*(\alpha(x) \cap X_0))$

The next lemma concerns only the orders, it says that an element linked to only one α-terminal is α-free and hence inessential in the order.

Lemma 5 : Let $|X|$ be a CF-order, $\forall x \in X$, such that $x \notin X_0$ and $\alpha(x) \cap X_0$ only contains one element, then x is α-free in $|X|$.

The following lemma shows that every inessential cell linked to more than one n-cell is face of an element of the support of this group of n-cells.

Lemma 6 : Let $|X|$ be a strongly normal order, $\forall S_0 \subseteq X_0$, every cell of $C^*_{|X|}$ that does not belong to $supp(\psi^*(S_0))$ and whose star contains at least two n-cells of $\psi^*(S_0)$ is face of at least one k-cell, $k < n$, of $supp(\psi^*(S_0))$.

In the following we call *closest face* of a cell e, a cell e' such that $e' < e$ and $\nexists e'' \in \psi^*(\beta(S_0))$ such that $e' < e'' < e$. It corresponds to the notion of β-closeness defined on orders.

The next lemma indicates that an inessential cell may be the closest face of at most one cell of the support of an object O.

Lemma 7 : Let $|X|$ be a strongly normal order, $\forall S_0 \subseteq X_0$, a cell e of $C^*_{|X|}$ such that $e \notin supp(\psi^*(S_0))$ cannot be the closest face of more than one cell $e' \in supp(S_0)$.

The last lemma shows that the cells not belonging to the support of an object and whose dimensions are maximal are some of the first inessential cells that can be removed.

Lemma 8 : Let $|X|$ be a strongly normal order, $\forall S_0 \subseteq X_0$, a cell e of $C^*_{|X|}$ such that $e \notin supp(\psi^*(S_0))$ and whose dimension is maximal is a closest face of one cell of $supp(\psi^*(S_0))$. Moreover its image by $\psi^{*^{-1}}$ is $\alpha_{|\beta(S_0)}$-unipolar.

We now prove our main result that is the α-kernel of any suborder of a strongly normal order $|X|$ is transformed by ψ^* into the support of the corresponding n-cells in $C^*_{|X|}$.

Theorem 1 (α-kernel and support) : Let $|X| = (X, \alpha)$ be a strongly normal order and $C^*_{|X|}$ its dual abstract complex. $\forall S_0 \subseteq X_0$:

$$x \text{ } \alpha\text{-free in the order } |\beta(S_0)| = (\beta(S_0), \alpha_{|S_0}) \Leftrightarrow \psi^*(x) \notin supp(\psi^*(S_0)).$$

Proof : *We are going to prove the theorem in two stages :*

$\Rightarrow$ *If x is $\alpha_{|\beta(S_0)}$-free in $|\beta(S_0)|$, we prove that its image by ψ^* does not belong to $supp(\psi^*(S_0))$, indeed :*

• *if x is $\alpha_{|\beta(S_0)}$-unipolar, then, by Lemma 4, $\psi^*(x) \notin supp(\psi^*(S_0))$;*

• *if x is $\alpha_{|\beta(S_0)}$-free, then, by Lemma 1, $\exists x' \in \alpha^{\square}(x)$ such that $\alpha(x) \cap S_0 = \alpha(x') \cap S_0$ and by Lemma 3 $\psi^*(x) \notin supp(\psi^*(S_0))$.*

$\Leftarrow$ *Let $e \in cl(\psi^*(S_0))$, with $e \notin supp(\psi^*(S_0))$, and $dim(e) < n$, we show that $\psi^{*^{-1}}(e)$ is $\alpha_{|\beta(S_0)}$-free*

(1). *if e is face of only one n-cell, e_n then $\alpha(\psi^{*^{-1}}(e)) \cap S_0$ contains only one element : $\psi^{*^{-1}}(e_n)$. $\psi^{*^{-1}}(e)$ is hence $\alpha_{|\beta(S_0)}$-free, by Lemma 5.*

(2). *if e is face of at least two n-cells of $\psi^*(S_0)$, we know by Lemma 6 that $\exists e_p \in$*

$supp(\psi^*(S_0))$ such that $e < e_p$.
$(2\text{-}a)$. If e has a maximal dimension, we know by Lemma 8 that $\psi^{*^{-1}}(e)$ is $\alpha_{|\beta(S_0)}$-unipolar.
$(2\text{-}b)$. We consider then the subcomplex C^1 of $\psi^*(\beta(S_0))$ obtained from $\psi^*(\beta(S_0))$ by removing all cells considered in (1). and $(2\text{-}a)$.. By Lemma 2, we know that C^1 is pure and strongly normal. We consider the cells of C^1 not belonging to $supp(\psi^*(S_0))$ and having a maximal dimension. Lemma 8 allows us to conclude as in $(2\text{-}a)$ that the image of these cells by ψ^{-1} are $\alpha_{|\psi^{*^{-1}}(C^1)}$-unipolar for the order $|\psi^{*^{-1}}(C^1)|$ deduced from $\beta(S_0)$ by removing only α-unipolar and α-free elements. These cells are hence $\alpha_{|\beta(S_0)}$-free. Finally, by recursively deleting cells of maximal dimension not belonging to $supp(\psi^*(S_0))$, we remove all cells not belonging to $supp(\psi^*(S_0))$. As each of them is a cell of maximal dimension for a strongly normal subcomplex of $\psi^*(\beta(S_0))$, it is α-unipolar for a suborder of $\beta(S_0)$ obtained by removing only $\alpha_{|\beta(S_0)}$-unipolar and $\alpha_{|\beta(S_0)}$-free elements of $\beta(S_0)$. Hence each of these cells is $\alpha_{|\beta(S_0)}$-free. □

We now define an *analogous of s.w.l.f. for strongly normal orders*. Let $|X|$ be a strongly normal order, a s.w.l.f. ϕ on $|X|$ is defined by : $\phi : \mathcal{P}(X_0) \times X \rightarrow \{0, 1\}$
- $\phi(S_0, x) = 1$ if $x \in X_0$,
- $\phi(S_0, x) = 0$ if $x \notin \alpha$-kernel$(\beta(S_0))$,
- $\phi(S_0, x) \leq \phi(X_0, x)$;
- $\phi(S_0, x) = \phi(\alpha(x) \cap S_0, x)$;

Theorem 2 (Order and complex with s.w.l.f.) : Let $|X|$ be a strongly normal order, the map f defined by : $\forall x \in X \forall S_0 \subseteq X_0$, $f(\psi(S_0), \psi(x)) = \phi(S_0, x)$ is a s.w.l.f. on $C^*_{|X|}$.

Proof : *Properties 1,3 and 4 of s.w.l.f. clearly hold for f. Property 2 is a consequence of theorem 1* □

4 Conclusion and Perspectives

We have shown that a strongly normal order and a pure strongly normal complex can be built from one another, in such a way that the inessential elements of one match the inessential ones of the other. Both models are interesting in the context of image representation because they are general and in particular not dimension dependent.They propose a general framework to represent the support of images and to express topology relations within. Nevertheless they are not equivalent and we list briefly the benefits that each model may take of the other. Concerning the notion of connectedness, Bertrand and al. had to consider different orders to define different connectedness, whereas the use of a s.w.l.f. allows to define several connectedness on the same order. Moreover the polyhedral complex used by Dominguez *et al.* is the basis of a multilevel architecture that allow to have discrete results consistent with continuous ones. They propose for example the definition of a fundamental group that may be transferred on orders. They also

prove a Seifert-Van Kampen theorem to compute fundamental groups. Dominguez *et al.* have in interesting theoretical results but their work has not been exploited yet in concrete applications. Bertrand *et al.* have more practical results and propose for example a definition of simple points which allows the thinning of objects in parallel. It may be interesting to see how their method may be applied to complexes.
In future works we intend to effectively transfer the tools of one model onto the other. We are also studying if the notions of surface defined on orders and checking if it is consistent with the classical definition of surface with polyhedral complexes.

References

[1] Alayrangues S., Lachaud J.-0., "Equivalence Between Order and Cell Complex Representations", *Internal Research Report RR-1272-02*, `http://www.labri.fr/Labri/Publications/Publications.htm`

[2] P.S. Alexandrov, *Combinatorial Topology, 1960*, DOVER publications.

[3] Ayala R., Dominguez E., Francès A.R. and Quintero A., "Digital Lighting Functions", *Discrete geometry for computer imagery, Springer Verlag, Vol. 1347, 1997.*

[4] Ayala R., Dominguez E., Francès A.R. and Quintero A., "Homotopy in Digital Spaces", *Discrete geometry for computer imagery, LNCS, Springer Verlag, Vol. 1953, 2000.*

[5] Ayala R., Dominguez E., Francès A.R. and Quintero A., "Digital Homotopy with Obstacles", *Electronic Notes in Theoretical Computer Science, vol 46, 2001*.

[6] Bertrand G., "New Notions for Discrete Topology", *Discrete geometry for computer imagery, Springer Verlag, Vol. 1568, pp. 216-226, 1999.*

[7] Bertrand G. and Couprie M., "A Model for Digital Topology", *Discrete geometry for computer imagery, Springer Verlag, Vol. 1568, pp. 227-239, 1999.*

[8] Graham - Grötschel - Lovasz editors, *Handbook of combinatorics vol II 1995*, Elsevier Science Publishers.

The Taming of the Hue, Saturation and Brightness Colour Space

Allan Hanbury

Centre de Morphologie Mathématique, Ecole des Mines de Paris

35 rue St-Honoré, 77305 Fontainebleau cedex, France

telephone: +33 1 64 69 48 05, fax: +33 1 64 69 47 07

e-mail: Hanbury@cmm.ensmp.fr

Abstract

The transformation of the RGB colour space to a hue, saturation and brightness colour space is essentially a conversion from a rectangular coordinate system to a cylindrical coordinate system. Nevertheless, a bewildering array of such conversions exist. We show that one of the main reasons for this is the dependence of the saturation values on the choice of the brightness function, and suggest a definition of saturation which is independent of the brightness. The usual ways of calculating brightness and hue are reviewed. Lastly, we examine some of the characteristics of the cylindrical colour coordinates and give a simple example in which the suggested cylindrical colour coordinates are used.

1 Introduction

The transformation of the RGB colour space to a hue, saturation and brightness colour space is essentially a conversion from a set of rectangular coordinates to a set of cylindrical coordinates. One could therefore ask how such a seemingly simple procedure could have given rise to the plethora of such transformations described in the literature, such as HSV [10], HSI [4], Triangle [10] and HLS[1]. It is shown in this article that in the definitions of these spaces, the saturation values obtained depend intimately on the expression chosen for calculating brightness, even though it is usually claimed that the saturation and brightness measures are independent. We then propose a definition of the saturation which is completely independent of the brightness function, and which therefore allows the free choice of the brightness function most suited to the task at hand. In order to complete the description of the space, we review the methods used to calculate brightness and hue, and we examine some of their characteristics.

Why, it may be asked, is such a colour representation space necessary? Surely it is better to use a standardised colour space such as the CIE L*a*b* space or its cylindrical coordinate version. The obvious objection to the use of the L*a*b* space is that one needs calibration

[1]The transformations to and from these spaces and some others are summarised by Shih [9].

information on the image which is being transformed from the RGB space, namely the colour coordinates of the source of illumination (the white point). This information is not always available for the images that are encountered in computer vision applications. If we do not have the necessary information, might it not be better to avoid making assumptions and estimations, and to use an alternate and more intuitive coordinate system for representing the information that is known, namely the coordinates of the colours in the RGB space.

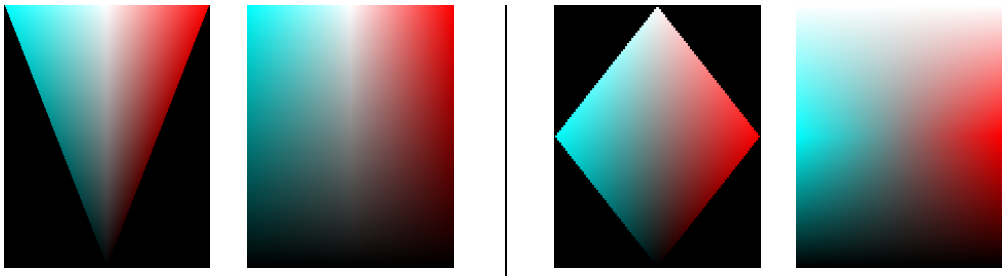
We begin with a discussion of the existing cylindrical coordinate colour representations (section 2), followed by a discussion and derivation of the suggested representation (section 3). In section 4, we present some of the characteristics of this representation. Finally, a simple example of its use is given in section 5.

2 Discussion of the existing transforms

In the RGB space, colours are specified as vectors (R, G, B) which give the amount of each red, green and blue primary in the colour. For convenience, we take $R, G, B \in [0, 1]$ so that the valid coordinates form the cube $[0, 1] \times [0, 1] \times [0, 1]$. The basic idea behind the transformation to a hue, saturation and brightness coordinate system is to place a new axis between $(0, 0, 0)$ and $(1, 1, 1)$, and to specify the colours in terms of cylindrical coordinates based on this axis. The new axis passes through all the achromatic or grey points (i.e. with $R = G = B$), and will therefore be referred to as the *achromatic axis*. The *brightness* gives the coordinate of a colour on this axis, the *hue* corresponds to the angular coordinate and the *saturation* corresponds to the distance from the achromatic axis.

One of the causes of the variety of such spaces is the number of different definitions of brightness. These definitions lead to spaces which have shapes which are not simply constructed as a pile of planar cross-sections of the cube taken perpendicular to the achromatic axis. Further problems with the existing transforms are due to them originally being developed for the easy numerical specification of colours in computer graphics applications [10]. Due to the associated brightness functions, the "natural" shape of the HSV space is a cone, and of the HLS space, a double cone. A vertical slice through the achromatic axis of each of these spaces is shown in figures 1a and 1c. The problem with using these representations when specifying a colour is that there are large regions which lie outside the cones. In order to avoid complicated verification of the validity of a specified colour, these spaces were often artificially expanded into cylinders by dividing the saturation values by their maximum possible values for the corresponding brightness. Slices of the cylindrical versions of the HSV and HLS spaces are shown in figures 1b and 1d. The cylindrical versions have often been carried over into image processing and computer vision, for which they are ill-suited.

We now consider two cases of the confusion that the cylindrical forms can cause. Demarty and Beucher [3] applied a constant saturation threshold in the cylindrical HLS space (figure 1d) to differentiate between chromatic and achromatic colours. This threshold can be represented by a vertical line on either side of the achromatic axis in figure 1d, and it is clear that this does not correspond to a constant saturation. Demarty [2] later improved the threshold by using a hyperbola in the cylindrical HSV space (figure 1b), which corresponds to a constant threshold



(a) Conic HSV (b) Cylindrical HSV (c) Bi-conic HLS (d) Cylindrical HLS

Figure 1: Slices through the conic and cylindrical versions of the HSV and HLS colour spaces. Colours to the right of the central achromatic axis have hues of $0°$, and colours to the left have hues of $180°$.

in the conic HSV space (figure 1a). Smith [11] makes the assumption that the cylindrical HSV space is perceptually uniform when a Euclidean metric is used, but upon examining figure 1b, one sees that a certain distance in the high brightness (top) part of the space corresponds to a far larger perceived change in colour than the same distance in the low brightness part of the space.

3 Derivation of a useful hue, saturation and brightness space

In this section, we examine a derivation of a cylindrical coordinate system in the RGB space, pointing out the choices which could (and have) lead to characteristics which are disadvantageous, and ending up with a cylindrical coordinate representation of the RGB space which is useful for computer vision. This derivation is based on the derivation of a Generalised Lightness, Hue and Saturation (GLHS) model [7] suitable for computer graphics applications.

3.1 Brightness

In order to conform to the terminology suggested by the CIE, we call a subjective measure of luminous intensity the *brightness*. The brightness function of the GLHS model is

$$L(\mathbf{c}) = w_{\min} \cdot \min(\mathbf{c}) + w_{\text{mid}} \cdot \text{mid}(\mathbf{c}) + w_{\max} \cdot \max(\mathbf{c}) \tag{1}$$

in which the functions $\min(\mathbf{c})$, $\text{mid}(\mathbf{c})$ and $\max(\mathbf{c})$ return respectively the minimum, median and maximum component of a vector $\mathbf{c}$ in the RGB space, and $w_{\min}$, w_{mid} and $w_{\max}$ are weights set by the user, with the constraints $w_{\max} > 0$ and $w_{\min} + w_{\text{mid}} + w_{\max} = 1$. Specific values of the weights give the brightness functions used by the common cylindrical colour spaces: $w_{\min} = 0$, $w_{\text{mid}} = 0$ and $w_{\max} = 1$ for HSV; $w_{\min} = \frac{1}{2}$, $w_{\text{mid}} = 0$ and $w_{\max} = \frac{1}{2}$ for HLS; and $w_{\min} = \frac{1}{3}$, $w_{\text{mid}} = \frac{1}{3}$ and $w_{\max} = \frac{1}{3}$ for HSI.

The *luminance* is the radiant intensity per unit projected area weighted by the spectral sensitivity associated with the brightness sensation of human vision [8]. This objective measure takes into account the fact that if one looks at red, green and blue light sources of the same

radiant intensity in the visible spectrum, the green will appear the brightest and the blue the darkest. The luminance function which corresponds to contemporary video displays is [8]

$$Y(\mathbf{c}) = 0.2125R + 0.7154G + 0.0721B \tag{2}$$

In the RGB space, we can visualise surfaces of iso-brightness (or iso-luminance). The surfaces of iso-brightness l contain all the points such that $L(\mathbf{c}) = l$ and intersect the achromatic axis at l. For the HSV and HLS spaces, these surfaces have a complicated shape (see [7] for details), and for the HSI space these surfaces are planes perpendicular to the achromatic axis. The surfaces of iso-luminance (equation 2) are planes oblique to the achromatic axis.

3.2 Hue

The hue angle is traditionally measured starting at the direction corresponding to pure red. The simplest way to derive an expression for this angle is to project the vector $(1, 0, 0)$ corresponding to red in the RGB space and an arbitrary vector $\mathbf{c}$ onto a plane perpendicular to the achromatic axis, and to calculate the angle between them. This gives the expression

$$H' = \arccos\left[\frac{R - \frac{1}{2}G - \frac{1}{2}B}{(R^2 + G^2 + B^2 - RG - RB - BG)^{\frac{1}{2}}}\right] \tag{3}$$

after which, in order to give a value of $H \in [0^\circ, 360^\circ]$, we apply

$$H = \begin{cases} 360^\circ - H' & \text{if } B > G \\ H' & \text{otherwise} \end{cases} \tag{4}$$

An approximation to this trigonometric expression is often used, and it is shown in [7] that the approximated value differs from the trigonometric value by at most 1.12°. A further comparison between the trigonometric hue and the approximated hue is given in section 4.

3.3 Saturation

For the derivation of an expression for the saturation of an arbitrary colour $\mathbf{c}$, we begin by looking at the triangle which contains all the points with the same hue as $\mathbf{c}$, as shown in figure 2. The intersection of this triangle and the iso-brightness surfaces are lines parallel to the line between $\mathbf{c}$ and its brightness value on the achromatic axis $\mathbf{L}(\mathbf{c}) = [L(\mathbf{c}), L(\mathbf{c}), L(\mathbf{c})]$.

Traditionally, the saturation is calculated as the length of the vector from $\mathbf{L}(\mathbf{c})$ to $\mathbf{c}$ divided by the length of the extension of this vector to the surface of the RGB cube. This definition, however, results in colour spaces in the form of cylinders discussed in section 2. Moreover, it is clear that this definition of the saturation depends intimately on the form of the brightness function chosen (i.e. on the slopes of the iso-brightness lines). An example of this dependence is shown in figure 3, in which the saturation of figure 3a is shown in figure 3b for the HSV space and figure 3c for the HLS space. In the original image, not all the pixels which appear white have RGB coordinates of exactly $(1, 1, 1)$. The slight variations in these RGB values are

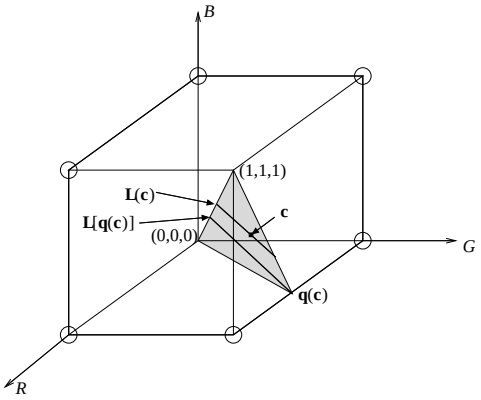


Figure 2: The triangle which contains all the points with the same hue as $\mathbf{c}$. The circled corners mark the edges of the cube containing the points furthest away from the achromatic axis.

amplified by the artificial expansion of the cones into cylinders, leading to the noisy regions in the saturation images and clearly demonstrating the dependence of the saturation on the brightness function.

In order to keep the conic or bi-conic forms of the spaces, it is necessary to change the definition of the saturation. Instead of the definition given above, we divide the length of the vector from $\mathbf{L}(\mathbf{c})$ to $\mathbf{c}$ (in figure 2) by the length of the vector between $\mathbf{L}[\mathbf{q}(\mathbf{c})]$ and $\mathbf{q}(\mathbf{c})$, that is, the longest vector parallel to $[\mathbf{L}(\mathbf{c}), \mathbf{c}]$ included in the iso-hue triangle, the vector which necessarily intersects the third corner $\mathbf{q}(\mathbf{c})$ of the triangle. We then end up with the following expression for the saturation

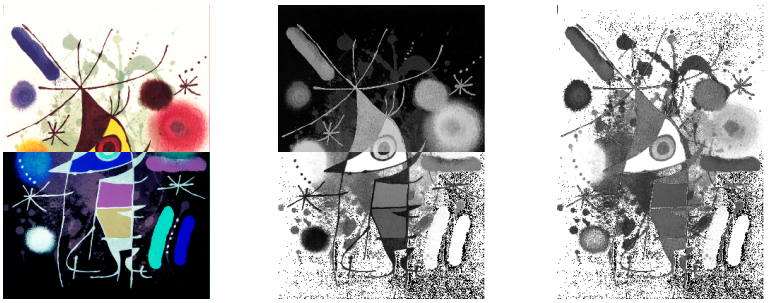
$$S = \frac{\|\mathbf{L}(\mathbf{c}) - \mathbf{c}\|}{\|\mathbf{L}[\mathbf{q}(\mathbf{c})] - \mathbf{q}(\mathbf{c})\|} \tag{5}$$

which is independent of the choice of the brightness function. This independence can be shown by using similar triangles [6]. An example of this saturation measurement is shown in figure 3d, where it should be compared to the corresponding HSV and HLS examples. The most visible improvement resulting from this definition is that both the white and black regions of the colour image are assigned a low saturation value.

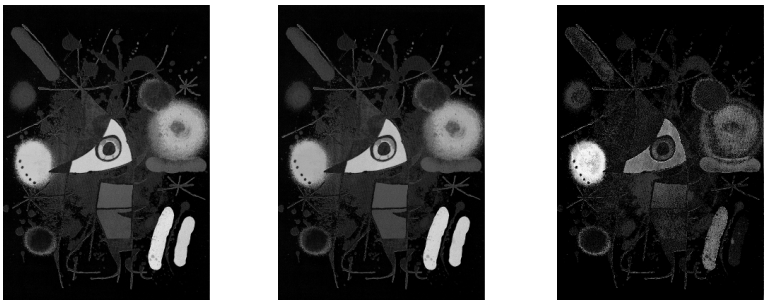
The points the furthest away from the achromatic axis are those on the edges of the RGB cube between the circled corners in figure 2. These points correspond to the most highly saturated colours, and if we project them onto a plane perpendicular to the achromatic axis, they form the edges of a hexagon, which correspond to the maximum distance a point can be from the achromatic axis for a given hue. A simpler expression for the saturation of point $\mathbf{c}$ can be obtained by projecting it onto this hexagon, and dividing the distance of the projected point from the centre of the hexagon by the distance from the centre to the hexagon edge at the same hue value.

3.4 Chroma

Carron [1] suggests the use of the distance of a point from the achromatic axis without the maximum distance normalisation as an approximation to the saturation, which he calls *chroma*.



(a) Colour image (b) Cylindrical HSV (c) Cylindrical HLS



(d) Saturation S (e) Chroma C (f) $S - C$

Figure 3: (a) "Le Chanteur" by Joan Mirò, with the bottom half inverted (by subtracting the values in the three colour channels from 255). The cylindrical saturation is shown in (b) for the HSV space and (c) for the HLS space. The brightness-independent (d) saturation and (e) chroma are shown, as well as (f) the difference between images d and e (contrast-enhanced).

This distance is multiplied by a constant so that for the six vertices of the projected hexagon, the chroma has a value of one. An example of the chroma is shown in figure 3e, and the difference between the chroma and the saturation images is shown in figure 3f (the contrast has been enhanced for better visibility, the maximum pixel value in the image is 0.107). The maximum possible difference between a saturation and a chroma value for a colour is 0.134.

3.5 Summary of the transform

A simple method to calculate the luminance, trigonometric hue, chroma and saturation coordinates is given here, based on the one suggested by Carron [1]. The changes with respect to the version given by Carron are the extension to calculate the saturation from the chroma, and the use of luminance instead of brightness. The first step is

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0.2125 & 0.7154 & 0.0721 \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6}$$

followed by the calculation of the chroma $C \in [0, 1]$

$$C = \sqrt{C_1^2 + C_2^2}$$

and the hue $H \in [0°, 360°]$

$$H = \begin{cases} \text{undefined} & \text{if} \quad C = 0 \\ \arccos\left(\frac{C_1}{C}\right) & \text{if} \quad C \neq 0 \text{ and } C_2 \leq 0 \\ 360° - \arccos\left(\frac{C_1}{C}\right) & \text{if} \quad C \neq 0 \text{ and } \mathrm{C}_2 > 0 \end{cases}$$

and, if required, the saturation $S \in [0, 1]$

$$S = \frac{2C \sin\left(120° - H^*\right)}{\sqrt{3}}$$

in which

$$H^* = H - k \times 60° \text{ where } k \in \{0, 1, 2, 3, 4, 5\} \text{ so that } 0° \leq H^* \leq 60° \tag{7}$$

The inverse of this transform is easily derived.

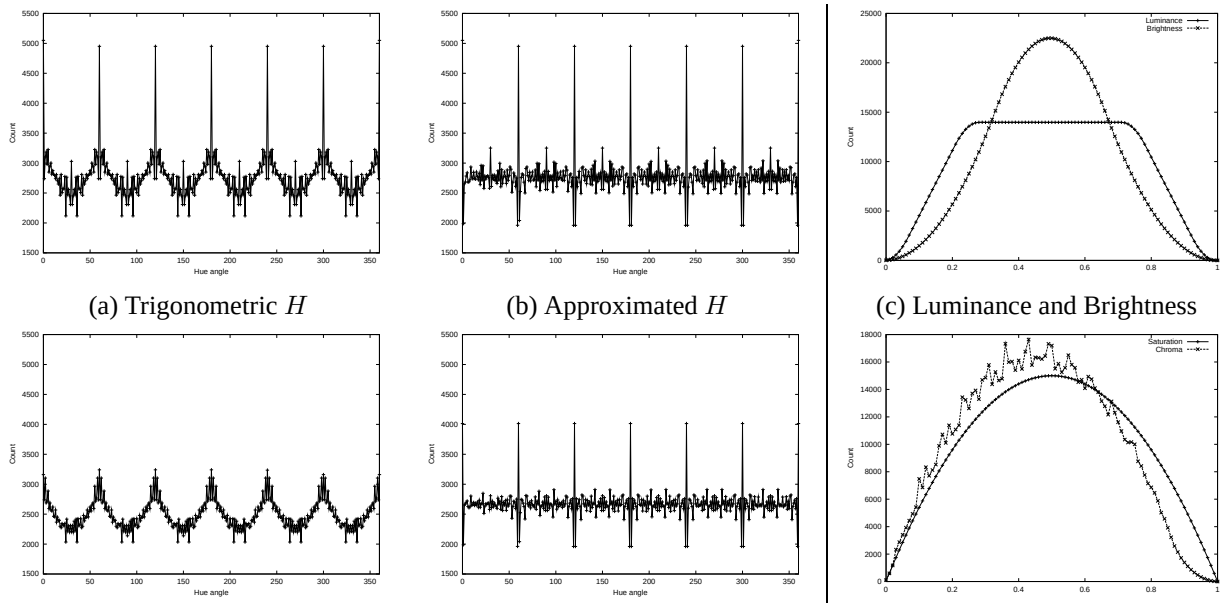
4 Characteristics of the space

In this section we examine the distributions of the hue, saturation, chroma and brightness coordinates for a transformation of a set of points equidistantly spaced in the RGB space to completely fill up the cube $[0, 1] \times [0, 1] \times [0, 1]$ (the distance between the points is 0.01).

We begin by examining the hue using the two calculation methods available, the trigonometric method and the approximate method. Histograms of 360 bins, with each bin corresponding to one degree, were calculated (the value in bin 360 is equal to the value in bin 0). The histograms for the trigonometric approach and for the approximate approach are shown in figures 4a and 4b respectively. The distributions are not smooth because we are calculating the angular coordinates of points distributed on a grid, but the most striking characteristic of these histograms are the strong peaks at each multiple of $60°$. If we ignore the peaks, it appears as if the approximate calculation gives a flatter distribution than the trigonometric calculation, for which the hexagonal structure of the planar cross-sections of the space is clearly visible.

What causes the peaks? Their distribution at multiples of $60°$ suggests that the hexagonal shape of the planar cross-sections are the cause. The fact that we are piling up many such hexagons to form the colour space could lead to a surplus of points in these directions forming the exaggerated peaks. To test this, we re-calculated the histograms using only the points with saturation values larger than 0.2, which gave the histograms shown in figures 4d (trigonometric method) and 4e (approximate method). By removing the interior part of the space, we have removed the peaks for the trigonometric hue. However, they are still present for the approximate hue, always accompanied by a one bin wide depression on either side. This demonstrates that the approximate method has a tendency to inflate the number of points assigned hues which are multiples of $60°$.

We now look at the brightness and luminance distributions, for which histograms (with 100 bins) are shown in figure 4c. The brightness measure used here is $L = \frac{1}{3}R + \frac{1}{3}G + \frac{1}{3}B$. These distributions do not have any particular features, and the choice is dependent on the preferences of the user or the requirements of the application.



(a) Trigonometric H (b) Approximated H (c) Luminance and Brightness

(d) Trig. H with threshold on S (e) Approx. H with threshold on S (f) Saturation and Chroma

Figure 4: The distributions of the hue, saturation, chroma, luminance and brightness coordinates after a transformation from an RGB cube containing a uniform distribution of points.

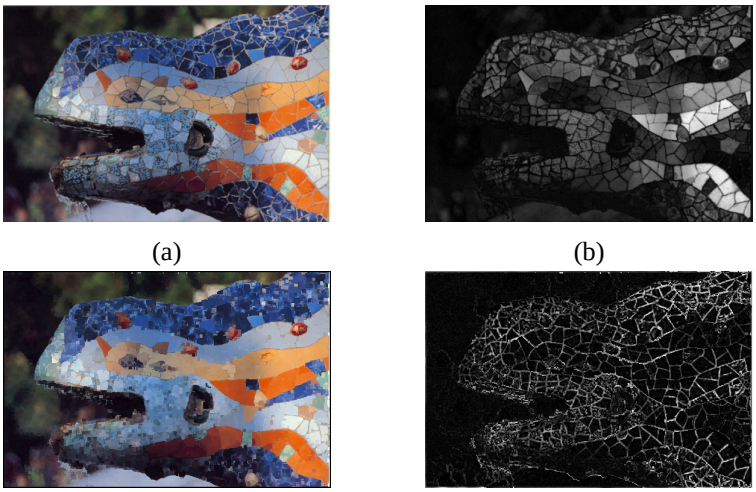
We look lastly at the saturation and chroma, for which one hundred bin histograms are shown in figure 4f. The saturation distribution is regular and symmetric around 0.5 due to its normalisation coefficient. The chroma distribution, on the other hand, is very irregular because of the distances calculated in a digital space, and descends to zero rapidly at the upper end due to the the hexagonal form of the planar cross-sections of the space.

The choice between the use of the approximate hue or trigonometric hue, and between chroma or saturation depends on the computing power available[2]. Given the computing power on our desktops, there is no excuse for not using the accurate versions in normal image analysis tasks. The only area in which one could consider using approximate hue or chroma are in very high speed industrial inspection tasks where the use of trigonometric hue and saturation might require the use of an extra DSP processor. However, a better approximation of the trigonometric hue can be obtained by the use of look-up tables for the trigonometric functions.

5 An example

We give a simple example of the use of the suggested hue, saturation and luminance coordinates. Figure 5a is a colour image in which we wish to extract the greyish lines between the mosaic tiles. The saturation of this image is shown in figure 5b, in which it is visible that the lines to be extracted have, in general, a lower saturation than the tiles. A morphological closing operation

[2]The approximate hue was introduced during the 1970's to speed up the interactive choice of colours in computer graphics programs.



(a) (b) (c) (d)

Figure 5: (a) The lizard image (size 544×360 pixels). (b) The saturation of the lizard image. (c) A morphological closing of the lizard image using a lexicographical order with saturation at the first level. (d) The top-hat — the Euclidean distance between images a and c.

with a square structuring element of size 5×5 pixels was applied to the initial colour image to give figure 5c. The colour vectors were completely ordered by using a lexicographical order with saturation at the first level, luminance at the second level and hue at the third level (the angular nature of the hues were taken into account). This closing succeeds in expanding the tiles to cover the grey lines. Finally, a form of top-hat was calculated by taking the Euclidean distance (in cylindrical coordinates) between figures 5a and 5c to give the greyscale image in figure 5d, in which the pixels of highest grey level correspond to the features we wish to extract.

This representation of the RGB space in cylindrical coordinates can be used in any application in which one of the HLS, HSI, etc. spaces are traditionally used, ensuring that the algorithms are not hampered by a poor representation of the data. Nevertheless, one should remember to take into account the angular nature of the hue component [5].

6 Discussion and conclusion

A critical evaluation of the hue, saturation and brightness or luminance colour spaces is presented, spaces which are essentially representations of the RGB space in cylindrical coordinates. These spaces are often used in computer vision, even though many of the suggested transformations found in the literature are optimised for the numerical specification of colours, and are badly suited to direct application to image processing. Two of the undesirable properties discussed are the artificial expansion of the conic or bi-conic spaces into cylinders, and the resulting dependence of the saturation on the brightness function used.

We have presented a formulation of the saturation which is independent of the brightness function, allowing an unconstrained choice of any brightness function (which has parallel iso-brightness surfaces), including a psycho-visual measure of the luminance. Comparisons of the

distributions of the cylindrical coordinates are presented, as well as a simple example which uses the suggested cylindrical colour coordinates. This example makes use of a Euclidean distance in the suggested colour space to approximate a morphological top-hat. It would be more correct to calculate this difference in the L*a*b* space for which the Euclidean metric is defined, but in which it is less easy to calculate a saturation-based morphological closing. Further applications using the suggested colour space are given in [6], including a real-time wood colour matching application.

References

[1] T. Carron. *Segmentations d'images couleur dans la base Teinte-Luminance-Saturation: approche numérique et symbolique*. PhD thesis, Université de Savoie, 1995.

[2] C-H. Demarty. *Segmentation et Structuration d'un Document Vidéo pour la Caractérisation et l'Indexation de son Contenu Sémantique*. PhD thesis, CMM, Ecole des Mines de Paris, 2000.

[3] C-H. Demarty and S. Beucher. Color segmentation algorithm using an HLS transformation. In *Proceedings of the International Symposium on Mathematical Morphology (ISMM '98)*, pages 231–238, 1998.

[4] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 1992.

[5] A. G. Hanbury and J. Serra. Morphological Operators on the Unit Circle. *IEEE Transactions on Image Processing*, 10(12):1842–1850, 2001.

[6] A. G. Hanbury. *Morphologie Mathématique sur le Cercle Unité: avec applications aux teintes et aux textures orientées*. PhD thesis, CMM, Ecole des Mines de Paris, 2002.

[7] H. Levkowitz and G. T. Herman. GLHS: A generalised lightness, hue and saturation color model. *CVGIP: Graphical Models and Image Processing*, 55(4):271–285, 1993.

[8] C. Poynton. Frequently asked questions about gamma. URL: http://www.inforamp.net/˜poynton/PDFs/GammaFAQ.pdf, 1999.

[9] T-Y. Shih. The reversibility of six geometric color spaces. *Photogrammetric Engineering and Remote Sensing*, 61(10):1223–1232, October 1995.

[10] A. R. Smith. Color gamut transform pairs. *Computer Graphics*, 12(3):12–19, 1978.

[11] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. PhD thesis, Columbia University, 1997.

Colour-Based Pruning of Model Hypotheses For Efficient ARG Object Recognition

A.R.Ahmadyfard, J.Kittler and D.Koubaroulis

Center for Vision, Speech and Signal Processing, University of Surrey

Guildford GU2 7XH,UK

tel:(441483) 689294 fax:(441483)686031

e-mail:(A.Ahmadyfard,J.Kittler,D.Koubaroulis)@eim.surrey.ac.uk

Abstract

In this paper we address the problem of object recognition from 2D views. A new object recognition approach which combines the MNS and ARG methods is proposed. In the new system we use the MNS method as a pre-matching stage to prune the list of model candidates. The ARG method then identifies the best model among the remaining hypotheses through the relaxation labelling process. The results of experiments show considerable gain in the ARG matching speed. Interestingly, as a result of model pruning, the entropy of labelling is reduced which improves the recognition rate for extreme object views as well.

1 Introduction

The recognition of objects from their 2D image is one of the crucial tasks in computer vision. Notwithstanding its classical applications such as robot vision and remote sensing, nowadays there is a lot of interest in object recognition in the context of image retrieval from large databases. Among the many methods which address the problem of 3D object recognition from 2D views[8], some represent the image of an object using geometric features[12]. During matching, the correspondence between the features from the test image and those of object model is sought by exploiting the geometric constraints. The object model which best matches the test image defines the identity of the object in the scene.

The other end of the methodological spectrum is occupied by appearance-based approaches in which the image of an object is represented using the raw pixel information[6]. In contrast with the former methods, the matching between two image of an object is accomplished by comparing the image descriptors in a feature space[1]. As a result, the matching in these methods is much faster than establishing feature correspondences. This is why the appearance-based approach has recently received a lot of attention for image retrieval. Unfortunately, the matching in feature space may fail to interpret a scene image unambiguously. This is likely to happen,

in situations where different objects have similar appearance, or the appearance is distorted by occlusion and the background clutter is complex.In such situations, these methods can be used only to identify a number of candidate models which may match the test image.
In this paper we propose to take advantage of the two approaches to construct a multistage recognition system. In this system we use the appearance-based approach to prune the list of object models in the model database. The resulting candidate models are then involved in the matching based on feature correspondences to determine the identity of the object in the scene. In[3] we proposed a recognition method in which each object is represented in terms of its image regions. The regions extracted are normalised in an affine invariant manner. The normalised regions of the image are represented by an Attributed Relational Graph (ARG) where each node and link between a pair of nodes are described using unary and binary features respectively[3]. The regions are characterised using shape and region appearance properties. Object recognition is achieved by comparing the scene ARG to the graph of object models using relaxation labelling. The results obtained in several experiments involving images of objects taken from different viewing angles in a cluttered environment were very promising. However for objects imaged from extreme viewing angles, and also under sever scaling, our method tends to fail (like most of the existing methods), as in these situations the shape of a segmented region, instead of being informative, is rather unreliable.
In a recent work[2] we modified our previous method by replacing the shape features extracted from the region boundary with more robust and reliable invariant features. The additional benefit of this representation is that we do not need to normalise the regions. We also modified the probabilistic relaxation labelling method used in[3] in order to minimise its sensitivity to the number of spurious nodes in the contextual neighbourhood.
For model pruning, we note that colour has been used as a cue for object recognition very successfully. Among the early efforts, Swain and Ballard[11] introduced a method based on the colour histogram.The sensitivity of this histogram approach to illumination changes was later reduced by Funt and Finlayson[5]. They advocated the use of relative colour rather than absolute colour for indexing. But as histogram matching is, in essence, a global approach, it cannot entirely overcome the sensitivity to changing background clutter. In order to improve robustness to background changes, the use of local colour invariant features has recently been receiving increasing attention. For instance Matas et al[10] proposed a method based on the matching of invariant colour features computed from multimodal neighbourhoods. The method is called Multimodal Neighbourhood Signature (MNS). It has been tested in image retrieval and object recognition applications with promising results[10]. Although colour-based methods in general are remarkably fast and for this reason they are popular in image retrieval, in the object recognition context they are not very reliable. The reason is that these methods match the features of object images in the colour space regardless of any spatial correspondence between them.Clearly, colour features alone cannot capture the structure of an object in the scene. Nevertheless the speed of colour based recognition motivated us to utilise this approach in combination with the ARG method to speed up the ARG matching. Accordingly, in this paper we use the MNS method as a pre-matching stage for the ARG method. In the proposed composite method, MNS prunes the list of model candidates for any given test image. The ARG method is then applied to identify, from among the remaining candidates, the models which match the

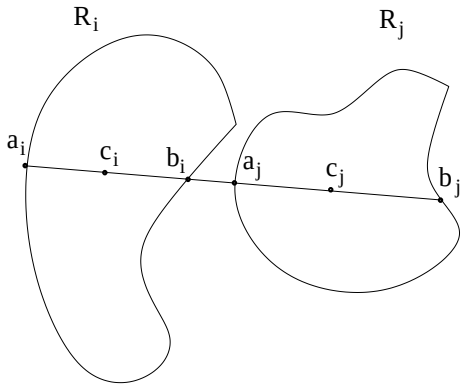


Figure 1: Binary measurements associated with pair of regions

objects in the scene image. We refer to the new system as the MNS-ARG method. The results of experiments carried out with the new method show that a considerable speed-up is achieved as a consequence of the model pruning. We also demonstrate that in addition to the speed gain, the recognition rate of the MNS-ARG system for extreme object views is better than for the stand alone ARG system.
The paper is organised as follows. In the next section we overview the ARG method[2]. In Section 3 we briefly describe the MNS method[10]. The experimental results are reported in Section 4. In the last section we draw the paper to conclusion.

2 ARG object recognition

In this method an object, or more specifically an image of the object is represented in terms of its segmented regions. The extracted regions are described individually and in pairs using their geometric and colour features. The entire image is then represented in the form of an Attributed Relational Graph(ARG)where each node corresponds to one of the regions and the edges between the nodes capture the region adjacency information.
The segmentation of an image into regions is based on colour homogeneity of the pixels. For this purpose we use the region growing method proposed in [7]. Each extracted region R_i is characterised individually using its (RGB) colour vector and we refer to this description as unary measurement vector $\overline{\mathbf{X}}_\mathbf{i}$. The relationship between a pair of regions R_i, R_j is described using geometric and colour measurements which constitute a so called binary measurement vector, $\overline{\mathbf{A}}_\mathbf{ij}$, defined as follows: Let us consider a pair of regions R_i and R_j in Fig 1. The line which connects the centroid points c_i and c_j intersects with the regions boundaries at a_i, b_i, a_j and b_j. Under affine transformation assumed here, the ratio of segments on a line remains invariant. Using this property, we define $m_1 = \frac{a_i a_j}{c_i c_j}$ and $m_2 = \frac{b_i b_j}{c_i c_j}$ as two elements of the binary measurement vector. In addition, the area ratio $AreaRatio = A_i/A_j$ and the distance between colour vectors $\overline{ColourDis} = \overline{C_i} - \overline{C_j}$ are used as complementary components of the binary measurement vector $\overline{\mathbf{A}}_\mathbf{ij}$. All the elements used in the binary measurement vector are affine

invariant.

Using the extracted regions and the associated measurement vectors we construct a Relational Attributed Graph in which a graph node O_i represents region R_i. The measurement vector, $\overline{\mathbf{X}}_\mathbf{i}$, embodies the node unaryattributes. The binary measurement vector $\overline{\mathbf{A}}_\mathbf{ij}$ describes the link between the pair of nodes O_i, O_j.

Using this approach an object is modelled in the recognition system by an attributed relational graph constructed from its representative image. The graphs of all objects in the model database are collected in a single graph referred to as the composite model graph. The content of an imaged scene is interpreted by constructing an ARG for the scene image. The resulting representation is referred to as the scene graph. Scene objects are then identified by matching the composite model and scene graphs.

The matching is accomplished using the relaxation labelling technique[13] which has been modified for the object recognition application[2]. In order to recognise objects in the scene image, the scene graph is matched against the composite model graph. This is in contrast with the methods in which the scene graph is matched against one object model at a time. By this matching strategy, we provide a unique interpretation for each part of the scene[2].

Before describing the algorithm for matching two ARGs let us introduce the necessary notation and the definitions required. We allocate to each node of the scene graph a label. Set $\theta = \{\theta_1, \theta_2, \cdots, \theta_N\}$ denotes the scene labels where θ_i is the label for node O_i. Similarly we use $\Omega = \{\omega_0, \omega_1, \cdots, \omega_M\}$ as the label set for the nodes of the composite model graph. In this label set, ω_0 is the null label which does not refer to any real node. It is added to be assigned to the scene nodes for which no other label in Ω is appropriate[13]. The contextual information in a graph is conveyed to a node from a small neighbourhood. In this regard, node O_j is a neighbour of O_i if the Euclidean distance between the associated regions is below a predefined threshold. We use set $\mathcal{N}_i$ to refer to the nodes in a neighbourhood of O_i. Similarly the labels in the neighbourhood of ω_α are referred to by set Ω_α.

By labelling we mean the assignment of a proper label from set Ω to each node of the scene graph. In this regard, $P(\theta_i = \omega_\alpha)$ denotes the probability that node O_i in the scene graph takes label ω_α. Obviously the majority of labels in Ω are not admissible for O_i. Therefore in the first stage of matching we compile a list of admissible labels for any scene node O_i denoted by Ω^i. This list is constructed by measuring the mean square error between the unary measurement vector for scene node O_i and the vectors of unary relation for all nodes in the model graph. Note that we include the null label in the label list of all the scene nodes, as it can potentially be assigned to any node in the scene. In the second stage of matching the modified labelling probability updating formula is applied[2]:

$$P^{(n+1)}(\theta_i = \omega_\alpha) = \frac{P^{(n)}(\theta_i = \omega_\alpha)Q^{(n)}(\theta_i = \omega_\alpha)}{\sum_{\omega_\lambda \in \Omega} P^{(n)}(\theta_i = \omega_\lambda)Q^{(n)}(\theta_i = \omega_\lambda)} \tag{1}$$

$$\begin{aligned} Q^{(n)}(\theta_i = \omega_\alpha) = \prod_{j \in \mathcal{N}_i} \{ & \sum_{\omega_\beta \in \{\Omega^j \cap \Omega_\alpha\}} P^{(n)}(\theta_j = \omega_\beta) P(A_{ij} | \theta_i = \omega_\alpha, \theta_j = \omega_\beta) \\ & + \sum_{\omega_\beta \in \Omega^j - \{\Omega^j \cap \Omega_\alpha\}} P^{(n)}(\theta_j = \omega_\beta)\eta \} \end{aligned} \tag{2}$$

The relaxation labelling technique updates the labelling probabilities in an iterative manner using the contextual information provided by the nodes of the graph. In this formula-

tion $Q(\theta_i = \omega_\alpha)$ is the support function which measures the consistency of the label assignments to the scene nodes in the neighbourhood of O_i assuming O_i takes label ω_α. The labelling consistency is expressed as a function of the binary measurement vectors associated with the centre node O_i and its neighbours. The support function consists of two parts: the first part measures the contribution from Ω_α neighbours (the main support) and the second part is added to balance the number of contributing terms via the other labels in Ω[2]. η is a parameter which plays the role of the binary relation distribution function $P(A_{ij}|\theta_i = \omega_\alpha, \theta_j = \omega_\beta)$ when the model nodes ω_α and ω_β are not neighbours.It takes a fixed value which is determined experimentally.
Upon termination of the relaxation labelling process, we have a list of correspondences between the nodes of the scene and model graphs. We count the number of scene nodes matched to the nodes of each object model and this measure is used as an object matching score.

3 MNS method

In the MNS method proposed by Matas et al[10] an image is described using a number of local invariant colour features computed on multimodal neighbourhoods detected in the image. In the first step of the MNS representation, the image plane is covered by a set of overlapping windows. For every neighbourhood defined in this manner, the modes of the colour distribution are computed with the mean shift algorithm[4]. The neighbourhoods are then categorised according to their modality as unimodal, bimodal, trimodal, etc. The invariant features are only computed from the multimodal neighbourhoods. For every pair of mode colours m_i and m_j in a multimodal neighbourhood, a 6-dimensional vector $v = (m_i, m_j)$(in RGB^2 domain) is constructed.The computed vectors are then clustered in RGB^2 space using the mean shift algorithm[4]. As an output of this process, for each detected cluster its representative vector is stored. The collection of all cluster representatives constitutes the image signature.
During recognition, the signature of a test image is matched to each model signature separately. As the outcome of this process, each model is given a score according to the dissimilarity between its signature and the test image signature. The models are then rank ordered according to their scores.
The details of the matching process between a test signature D and a model signature Q are as follow: Consider the test and model signatures as sets of features $D = \{f^i{}_D : i = 1..m\}$ and $Q = \{f^j{}_Q : j = 1..n\}$. Recall that each feature in these sets is a 6-dimensional vector in the RGB^2 space. For every pair f^i_D,f^j_Q the distance $d(f^i_D, f^j_Q) \equiv d_{ij}$ is used as the similarity measure between the two features. Now the test and model signatures D and Q are considered as a bipartite graph where the edge between pair of nodes i and j is described by the distance d_{ij} ($d_{ij} = d_{ji}$). A match association function $u(i) : Q \rightarrow 0 \bigcup D$ is defined as a mapping of each model feature i to a proper test feature or to 0 (in case none of the test features matches). In the same manner a test association function $v(j) : D \rightarrow 0 \bigcup Q$ maps each test feature in D to a feature in Q or to 0. A threshold T_h is used to define the maximum allowed distance between two matched features. The algorithm can be summarised as follows:

Algorithm 1: **MNS Matching**

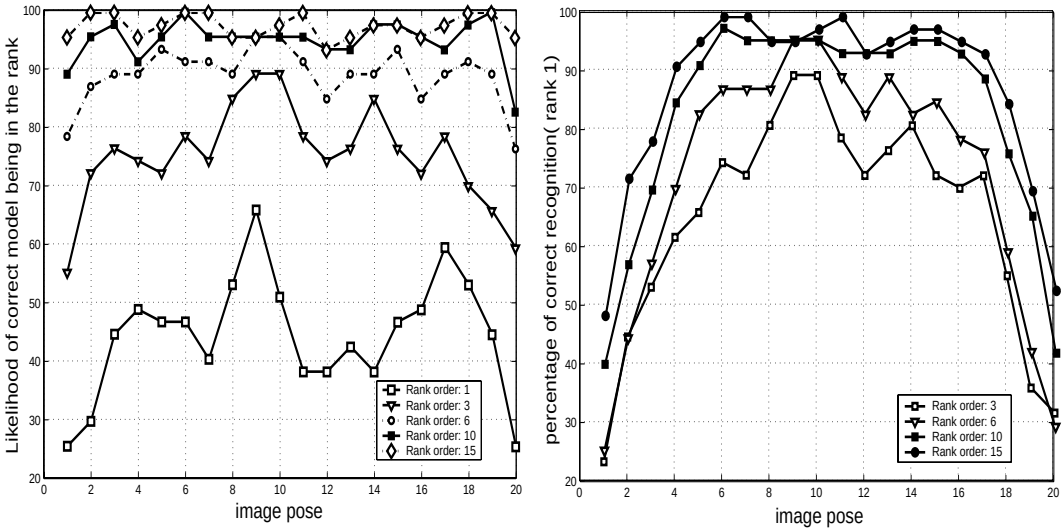
1. Set $u(i) = 0 \quad \text{and} \quad v(j) = 0 \quad \forall\, i, j.$
2. From each signature s compute the invariant features f_D^i, f_Q^j according to the colour change model dictated by the application.
3. Compute all pairwise distances $d_{ij} = d(f_D^i, f_Q^j)$ between the test and model features.
4. Set $u(i) = j,\ v(j) = i$ if $d_{ij} < d_{kl}$ and $d_{ij} < T_h \quad \forall\, k, l$ with $u(k) = 0$ and $v(l) = 0$.
5. Compute signature dissimilarity as $\Delta(D, Q) = \sum_{(\forall\, i\,:u(i)\neq 0)} d_{ij} + \sum_{(\forall\, i\,:u(i)=0)} T_h$

In summary function Δ measures the dissimilarity between the test image and model signatures. This measurement consists of two parts. The first term represents the goodness of fit between the features of the candidate model and the test image features. The second term penalises any unmatched model features. The models are then ranked in the increasing order of their signature dissimilarities.

4 MNS-ARG matching

As the complexity of labelling in the ARG method directly depends on the size of the graph involved in a match, any effort to reduce the graph complexity would speed up the matching process. It is worth recalling that during ARG matching, before labelling and at the end of each iteration, we prune inadmissible labels from the candidate list of labels for each node in the test graph, which speeds up computation.We expect to achieve a further gain in the matching speed by pruning the model candidates using the MNS method. The most important benefit of the model pruning is the reduction in the number of nodes (labels) in the model graph which directly speeds up the labelling process.Accordingly, for a given test image first we use the MNS method to provide a list of model candidates. We then apply the ARG method to identify the model which best matches the test image. We refer to this new system as the MNS-ARG method.

We designed an experiment to demonstrate the effect of model pruning on the performance of the ARG method. We compared ARG with the MNS-ARG method from the recognition rate and the recognition speed points of view. The experiment was conducted on the SOIL-47 (Surrey Object Image Library) database which contains 47 objects each of which has been imaged from 21 viewing angles spanning a range of up to ± 90 degrees. The database is available online[9]. In this experiment we model each object using its frontal image while the other 20 views of the objects are used as test images.The size of images used in this experiment is 288×360 pixels.



(a) The likelihood of the correct model being in the rank

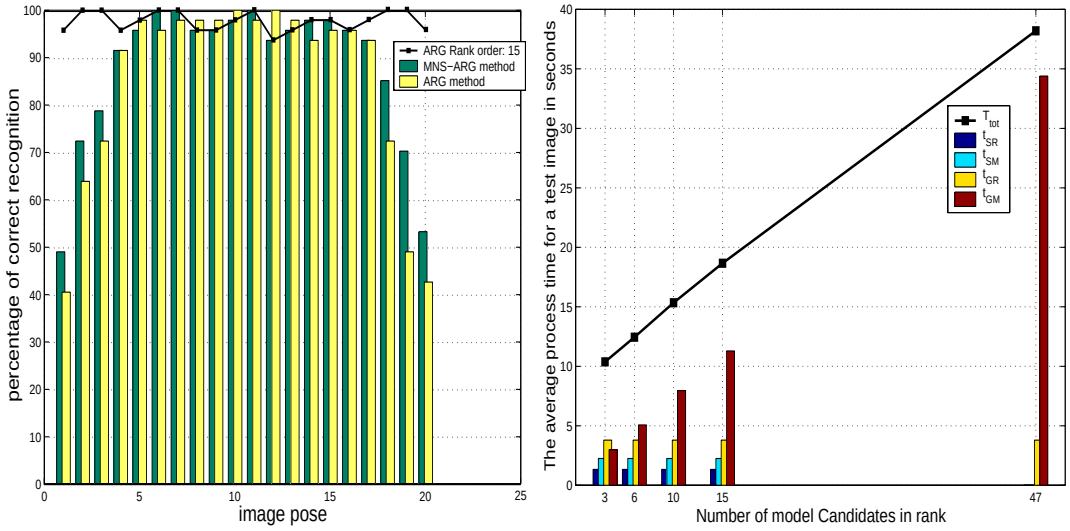
(b) The percentage of correct recognition (rank 1 after the ARG matching)

Figure 2:

For each test image we applied the MNS method to rank the candidate models matched to it. In this regard, different rank orders were selected to evaluate the MNS matching capability. In fig 2(a) we plot the percentage of cases in which the candidate rank order includes the correct model as a function of object pose. This measure is reported for different rank orders (3,6,10,15). The results illustrate that the recognition rate (rank 1) for the MNS method is not very high. As expected, when the rank order increases, the list of the top ranking models is more likely to include the correct model. For instance for rank order 15, in more than 95% of cases we have the correct model among the candidates.

The ARG method was then applied to identify the object model based on the rank order list selected by the MNS method. This recognition procedure was applied to all test images in the database. In fig 2(b) the recognition rate for the MNS-ARG method is plotted as a function of object pose for different rank orders. The results show a good recognition performance for the case when the rank order is more than 10. The results in figs 2(a) and 2(b) show that, apart from the extreme object views, the recognition rate is limited by the MNS performance. For extreme object views, as expected, the MNS-ARG method fails to recognise the object, but notably all candidates are rejected(the miss-classification rate is as low as 10%). This rate is remarkable in comparison with the miss-classification rate of MNS which is up to 75% (fig 2(a)). The failure to recognise objects from their extreme views is due to the significant distortion of the segmented regions. In these situations ARG is not able to establish correspondence between the test image and the correct model.

To demonstrate the effect of model pruning on the recognition rate of the ARG method the associated rates for ARG (without model pruning) and the MNS-ARG method for rank 15 are



(a) The percentage of correct recognition for the ARG and the MNS-ARG methods (rank size 15)

(b) The average process times for recognition of a test image in the ARG and MNS-ARG methods

Figure 3:

plotted in fig 3(a). As a base line we added the matching performance of the MNS method for rank order 15 to this graph. The results show that the model pruning improves the recognition rate for extreme object views. For such views the hypotheses at a node of the test graph do not receive a good support from its neighbours (problem of distortion in image regions). Moreover a large number of labels involved in the matching increases the entropy of labelling. When the number of candidate labels for a test node declines by virtue of model pruning the entropy of labelling diminishes. Consequently it is more likely for a test node to take its proper label (instead of the null label).

Referring to the results in fig 3(a) the recognition rate of MNS-ARG for some object views is occasionally slightly lower than that of the ARG method . The failure is due to the absence of the correct model among the candidates in the rank order list.

We now consider the computational advantage of the model pruning. As the model images in both the ARG and MNS methods are represented off-line we do not consider the cost of model construction in the recognition system processing. In the ARG method the recognition task consists of two stages: the representation of the test image in an ARG form and the graph matching. We refer to the associated process times as t_{GR} and t_{GM} respectively. By analogy, MNS matching also involves two stages: the extraction of the image MNS signature and the signature matching. The corresponding process times are referred as t_{SR} and t_{SM} respectively.

The total recognition time for the ARG matching is $T_{ARG} = t_{GR} + t_{GM}$. When we deploy MNS for model pruning, the total MNS-ARG process time is $T_{MNS-ARG} = t_{SR} + t_{SM} + t_{GR} + t_{GM}$. Among the terms in $T_{MNS-ARG}$ only the graph matching time t_{GM} varies with the size

of the list of candidates. In fact this process time depends on the number of nodes in the model graph which is a function of the number of models and their complexity. In fig 3(b) we plot the average process time which the ARG and MNS-ARG methods take to recognise the object in a test image. The experiment was run on a PC with a Pentium3 800MHZ processor and the CPU time is given in seconds. As expected the total recognition time is a linear function of the number of models in the rank order list. The results demonstrate that the speed gain obtained by pruning the model list is significant. For instance considering MNS-ARG with the rank order 15, the recognition time is about 18 seconds which is less than half of the recognition time for the ARG method. Note that this gain in speed is achieved without any loss in recognition performance.

5 Conclusion

The problem of object recognition from 2D views was addressed. A new object recognition system which combines the Attributed Relational Graph(ARG) and Multimodal Neighbourhood Signature (MNS) methods was proposed. In the proposed system first we perform non-contextual matching using MNS to prune the number of candidate models.In the next stage ARG matching is applied to identify the correct model for each object in a test image. The results of experiments showed a considerable gain in matching speed. As another benefit of model pruning, the results showed improvement in the recognition rate for extreme object views.

References

[1] Leonardis A. and Bischof H. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.

[2] A. Ahmadyfard and J. Kittler. submitted to EUSIPCO.

[3] Ahamdyfard A.R and Kittler J. Region-based object recognition: Pruning multiple representations and hypotheses. In *Proceedings of BMVC*, pages 745–754, 2000.

[4] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proceedings of ICCV*, pages 1197–1203, 1999.

[5] Funt B. Finlayson G. and Barnard J. Color constant color indexing. *IEEE Transaction on PAMI*, 17((5):522–529, 1995.

[6] Murase H. and Nayar S. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, pages 5–24, 1995.

[7] R. Haralick and L. Shapiro. Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, pages 100–132, 1985.

[8] Wolfson H.J. Model-based object recognition by geometric hashing. In *Proceedings of ICCV*, pages 526–536, 1990.

[9] http://www.ee.surrey.ac.uk/EE/VSSP/demos/colour/soil47/.

[10] J. Matas, D. Koubaroulis, and J. Kittler. Colour image retrieval and object recognition using the multimodal neighbourhood signature. In *Proceedings of ECCV*, pages 48–64, 2000.

[11] Swain M.J. and Ballard D.H. Colour indexing. *Intl. Journal of Computer Vision*, 7(1):11–32, 1991.

[12] Pope R. Model-based object recognition a survey of recent research. Technical report, 1994.

[13] Christmas W.J., Kittler J., and Petrou M. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on PAMI*, pages 749–764, 1995.

A general algorithm for finding transitions along lines in colored images

Felix v. Hundelshausen, Raúl Rojas
Free University of Berlin, Department of Computer Science
Takustr. 9, 14195 Berlin, Germany
Tel. ++49-(0)30-838-75-170, Fax. ++49-(0)30-838-75-109
e-mail: {hundelsh|rojas}@inf.fu-berlin.de

Abstract

One primary issue in computer vision is to find edges in images. Edges are important because they yield object boundaries, whose determination is necessary for object recognition, object tracking and localization.
In this paper we propose the "Dual-Window-Transition-Search Algorithm" (DWTS), a new method to detect edges in colored images that restricts image analysis to a higher level knowledge line segment onto the image. The algorithm yields a so called transition function which specifies a probability-like value for each position on the line that indicates whether a transition (edge) is present at the specified position. Detected transitions can be discontinuities in luminance, color and even low-level texture.
The benefits of the algorithm are that it does not require any color class definitions from the user. Furthermore the scale space on which the algorithm works can be configured, allowing the algorithm to be selective only to specific ranges of frequency.
This algorithm can be used as a general building block for several visual high-level algorithms, which address problems as automatic color adjustment, segmentation, object detection, localization of a mobile robot, tracking objects, optical flow and stereo vision.

1 Introduction

It is well known, that in biological vision systems some form of data encoding occurs at a very early stage in image processing. This became evident when Hubel and Wiesel discovered in 1979 that neurons in the primary visual cortex might fire only when the receptive field of the cell is exposed to a bright line at particular location and angle [9].
Not only because of this biological hint but also intuitively, it it clear that edges or transitions (luminance discontinuities) play an important role for vision, since they indicate object boundaries.
Marr and Hildreth [12] proposed to detect edges by finding local maxima of the derivative of the image signal. This idea was advanced by Canny [4], whose edge detector is the current

standard scheme. The idea is based on finding the maxima of the partial derivative of the image function in the direction orthogonal to the edge and smoothing the signal along the edge direction. Typically, the Canny edge operator is implemented by first convolving the image with a Gaussian and then looking for the maxima of the first derivative.
There are many other edge detector schemes such as the Laplacian of Gaussians (LOG), the Prewitt and Roberts edge detectors which all rely on convolving the whole image or parts of it with some filter mask.
The above edge detectors have the main disadvantages that they are not designed for color images and that they are suited only to find edges between different luminosities, but not to find edges between different textures. But this is necessary for most real-world applications.

Thus, this paper proposes a new algorithm, that is able to find edges in colored images. It does not need any previous color specification from the user and is able to even find transitions between regions of different low-level texture. With ”low-level-texture” we mean textures that differ by different fractions of noisy colors. Thus the algorithm is not designed to find transitions between textures that have equal fractions of colors but different shapes within the structure of the texture.
The algorithm is based on calculating color histograms for each position on the line and defining a difference measure to introduce a metric in the histogram space. We call the algorithm the ”Dual Window Transition Search Algorithm”. It is a general algorithm because it can not only be applied for image signals but for any signal that might even consist of different components (as the colors in vision). The remainder of this paper is organized as follows: Section 2 describes the main principle of the approach. Section 3 describes the Dual-Window-Transition-Search Algorithm (DWTS). Section 4 describes the results from simulated and real image data. Finally, Section 5 concludes the paper by summarizing the paper and describing fields of application.

2 Spatial Selectivity

Rather than searching edges in the whole image the algorithm is based on the following principle:
The algorithm just searches for transitions along a user-specified line through the image. This is reasonable because prior knowledge about the possible location of edges can be used to concentrate the computation power on these regions. For instance, when tracking a moving object, the object’s boundary edges will only move a little in two consecutive images (provided that the video rate is high enough). Thus, we often have a prior knowledge of the approximate locations where edges might occur.
When comparing the algorithm with processes in primate vision, one can think of selecting a neuron in the V1 (primary visual cortex), going back to the retina, and just evaluating those retinal receptors which are connected to the cell in V1. In this way computing power is only applied where it is needed.

3 Dual-Window-Transition-Search Algorithm (DWTS)

In the following section we describe the new algorithm. We first give a short overview, followed by a detailed description of the single steps respectively.

3.1 Method Overview

Our approach to find the transitions along a line is based on building and comparing color histograms, that describe the low-level texture of a set of pixels.
Each point on the line divides the line in a left and a right part (thought in the direction of the line). For each point on the line we calculate two color histograms corresponding to a left and right region. By defining an adequate distance measure between the histograms we calculate a value that lays between 0 and 1, that indicates the discrepancy of the two regions. Applying this method to all points on the line yields a function (the transition function) that has maxima where transitions are most likely. The frequency sensitivity of the transition function can be tuned by the predefined size of the regions. Let w be the length of the region to be inspected. Then the steps of the algorithm are summarized as the following:

1. Get the colors of all pixels on a imaginary line from point A to point B.
2. Extract the most important colors by using a clustering algorithm.
3. Begin with the w'th pixel on the line.
4. Build histograms for the left and the right side of the current point.
5. Calculate the distance between the histograms by defining a distance measure which is based on a predefined distance function between two colors.
6. Repeat steps 4-6 for all the other points on the line, yielding the transition function T.
7. Find the local maxima of T and determine the corresponding image locations of the found transitions.

3.2 Color distances

The later definition of the distance measure between histograms is based on a predefined color distance function

$$\begin{array}{rccc} d_{Color}: & ColorSpace \times ColorSpace & \longrightarrow & [0,1] \\ & (c_0, c_1) & \longmapsto & d_{Color}(c_0, c_1) \end{array}$$

where $ColorSpace$ is an arbitrary color space. The distance function must be positive definite, symmetric and must fulfill the triangle inequality. We emphasize that any color space can be used, provided the user can specify any appropriate metric.

3.3 Color Clustering

To reduce the number of colors that will be used for the histograms we cluster the colors that appear along the line. There are two possible approaches to influence the clustering process. The first is to specify a color distance threshold, which yields an undefined number of color clusters having a distance to each other greater than the threshold. The second possibility is not to specify any threshold but to predefine the maximum number of color clusters. Then the algorithm can iteratively adjust an internally maintained color threshold until the specified number of color clusters are obtained. Thus, for lines which pass through regions with very different colors the internal threshold will be high, but for lines which pass through regions with only small color deviations the color clustering will be very sensitive.

We first describe the clustering algorithm based on the predefined threshold since the other one relies on it.

3.3.1 Using a predefined threshold

Consider a set $C = \{c_0, c_1, ...c_{n-1}\}$, $c_i \in ColorSpace$, of colors, a color distance function d_{Color} and the color threshold $t_{col} \in [0, 1]$. The algorithm yields a set of k color clusters $Z = \{clust_0, clust_1, ..., clust_{k-1}\}$ where $clust_i := (m_i, q_i) \in ColorSpace \times [0, 1]$.
The element m_i is the center of all colors that correspond to cluster i and q_i is the fraction of all colors in C that belong to the cluster. Hence, $\sum_{i=0}^{k-1} q_i = 1$.
The algorithm starts with an empty set Z and consecutively adds clusters, if necessary.
The basic intention is the following:
For all colors c_i in C, if there is a cluster $clust_s$ in Z whose center m_s has a distance $d_{Color}(c_i, clust_s) \leq t_{col}$, then attach the color to the cluster by updating m_s and q_s. Otherwise start a new cluster.

3.3.2 Specifying the maximal number of clusters

When specifying the maximal number n_{max} of clusters the algorithm starts with an internal threshold $t_{col} = 0.5$ and seeks an appropriate value through binary search. In each step the above algorithm yields the number of clusters and thereby decides on the adjustment of t_{col}. Before starting the search there must be a test to ensure that it is possible to obtain n_{max} clusters. For instance, if all colors in C are equal, then it is not possible. In such a case, less than n_{max} clusters will be obtained. We note that this simple clustering algorithm can be replaced by more sophisticated methods [5].

3.4 Color Histograms

Since the heart of the algorithm is based on evaluating a distance function between two color histograms we first provide their definition.

Definition 1 *A **color vector** $\vec{c}$ of dimension k over a color space $ColorSpace$ is a vector $\vec{c} \in ColorSpace^k$.*

Definition 2 *A **color histogram** H of size k over a color space $ColorSpace$ is a tupel $H = (\vec{c} \in ColorSpace^k, \vec{q} \in [0,1]^k)$ that fulfills the condition: $\sum_{i=0}^{k-1} q_i = 1$*

Definition 3 *The **size** of a color histogram $H = (\vec{c}, \vec{q})$ is the dimension of $\vec{c}$.*

Next, we want to derive a distance function between two arbitrary color histograms $H_A = (\vec{c_A}, \vec{q_A})$ and $H_B = (\vec{c_B}, \vec{q_B})$. Let s_A, s_B be the sizes of H_A and H_B, respectively.

$$\begin{array}{rccc} d_{Hist}: & HistogramSpace_{s_A} \times HistogramSpace_{s_B} & \longrightarrow & [0,1] \\ & (H_A, H_B) & \longmapsto & d_{Hist}(H_A, H_B) \end{array}$$

Here $HistoSpace_k := ColorSpace^k \times [0,1]^k$.

3.5 Properties

The distance function should have the following properties:

- The distance function should introduce a metric to the $HistogramSpace$. Thus it must be positive definite, symmetric and should obey the triangle inequality.
- Assume that H_A and H_B have a color vector of dimension 1.
  Thus $q_{A0} = q_{B0} = 1$. Then the distance between H_A and H_B should be the distance of their only colors, $d_{Hist}(H_A, H_B) = d_{Color}(c_{A0}, c_{B0})$.
- The distance should be zero, if the two histograms are equal, which means that for each color of c_A there exists an exact corresponding color in c_B and the fractions of the colors are equal.
- A difference in the histogram fractions should yield a large distance, if and only if the corresponding colors differ much (in the sense of d_{Color}.

To find an appropriate metric, we first have to make a further definition:

Definition 4 *The **color similarity matrix** $S_{\vec{c_A}\vec{c_B}} \in M(m_A \times n_B, [0,1])$ of the color vectors $\vec{c_A}$ and $\vec{c_B}$, with $m_A = dim(\vec{c_A})$ and $n_B = dim(\vec{c_B})$, is defined by:*
$S_{\vec{c_A}\vec{c_B}} = (s_{ij}), \quad i = 0, 1, ..., m_A - 1; \quad j = 0, 1, ..., n_B - 1$ *with*
$s_{ij} = 1 - d_{Color}(c_{Ai}, c_{Bj})$.

We note, that due to to symmetry of d_{Color} the color similarity matrix is symmetric, if the color vectors have the same dimension.

Definition 5 *The **color histogram difference** d_{Hist} between two color histograms $H_A = (\vec{c_A}, \vec{q_A})$ and $H_B = (\vec{c_B}, \vec{q_B})$ with $dim(\vec{c_A}) = dim(\vec{c_B})$ is defined by:*

$$d_{Hist}(H_A, H_B) := \frac{1}{2} \vec{\Delta q}^{\,T} S \vec{\Delta q} \quad \textit{where } \vec{\Delta q} = \vec{q_B} - \vec{q_A} \tag{1}$$

3.6 Transition Function

To obtain the desired transition function, we first scan the colors along the selected line (i.e. by using Bresenham's algorithm [3]). Let c_i be the i'th color along this line. Let the total number of colors be n.
Next, we apply the color clustering algorithm to the scanned colors and obtain a set $Z = \{clust_0, clust_1, ..., clust_{k-1}\}$ of color clusters, with $clust_i = (m_i, q_i)$. Here m_i is the color center of cluster i.
We define a color vector $\vec{v} \in ColorSpace^k$ by:

$$\vec{v} := (m_0, m_1, ...m_{k-1})^T \tag{2}$$

We perform a sequence of steps for all colors from c_w to c_{n-w}. Remember, that w is the size (number of colors) of the left and right window[1] that will be used to build the color histograms. For the first and last positions on the given line, we cannot define regions that lie completely within the line. Thus, we can not reliably determine the color histograms for that points. Hence, we exclude them from analysis. If they should be included, the user has to specify a longer line. The steps for each position i on the line are:

1. Let $w_{left} := \{0, 1, ..., n-1\} \cap \{i-w, i-w+1.., i-1\}$
   and $w_{right} := \{0, 1, ..., n-1\} \cap \{i, i+1, ..., i+w-1\}$ be the index sets of the left and right region of position i, respectively. We build the histograms $H_{left} := (\vec{v}, \vec{q_{left}})$ and $H_{right} := (\vec{v}, \vec{q_{right}})$ by scanning the colors of the corresponding region and updating the respective q-values by the color distance of the actual color to the color of the respective bin of the histogram. The updates are performed in the way that first the distances are summed up for each bin and finally, all values are scaled down to sum up to 1.

2. Calculate the difference $t_i := d_{Hist}(H_{left}, H_{right})$ between the two histograms by applying definition 5(page 5). We can use the reduced version $d_{Hist}(H_{left}, H_{right}) = \frac{1}{2}\vec{\Delta q}^T S \vec{\Delta q}$ since the color vectors of both histograms are equal.

We obtain a sequence of values $t_w, t_w + 1, ...t_n - w \in [0, 1]$ that indicate the existence of a transition.

3.7 Optimization

Instead of recalculating the histograms and the histogram difference from scratch while passing through the colors of the line, considerable speed enhancements can be achieved by just updating the histograms and the difference measure from step to step.

For the convenience of derivation we reformulate the histogram difference measure from equation 1 by using values to specify the histogram fractions that are scaled up in the way that the value 1 is mapped to the window size w:

$$d_{Hist}(H_A, H_B) = \frac{1}{2w^2}\vec{\Delta z}^T S \vec{\Delta z} \tag{3}$$

[1]The window is meant just along the line segment, not as a 2-D image point neighborhood.

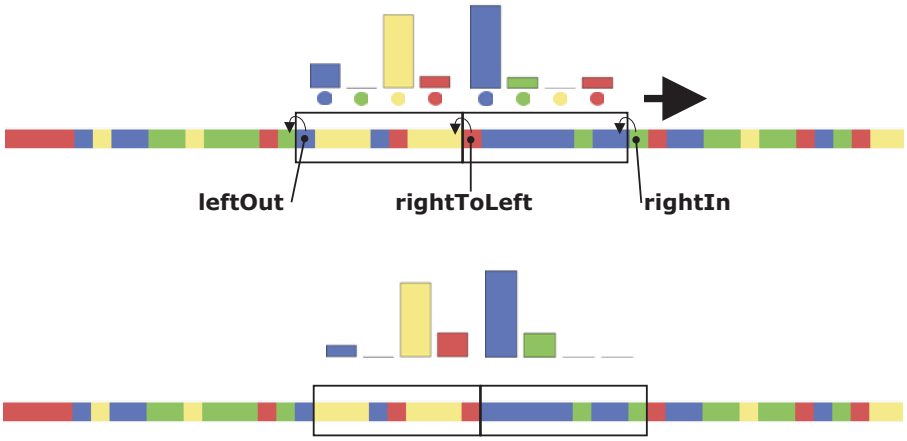


Figure 1: The left and right window are displaced one step to the right. The corresponding color histogram is shown above each window. To update the histograms only three pixels have to be dealt with.

where $\vec{\Delta z} = \vec{z_{right}} - \vec{z_{left}}$. Here $\vec{z_{left}}, \vec{z_{right}} \in I\!R^k$ are vectors whose components specify the contribution of the colors in the respective window to the respective color cluster of the histogram. With w being the size of each window, we have:

$$\sum_{i=0}^{k-1} z_{right\ i} = \sum_{i=0}^{k-1} z_{left\ i} = w \tag{4}$$

Suppose, that at step s we have calculated $d_{Histo}(H_{left}, H_{right})$. Next, we want to determine $d_{Histo}(H'_{left}, H'_{right})$ at the next step s', where H'_{left}, H'_{right} are the color histograms of one position later in the direction of the line (see figure 1):

$$d_{Hist}(H'_A, H'_B) = \frac{1}{2w^2} \vec{\Delta z'}^T S \vec{\Delta z'} \tag{5}$$

where $\vec{\Delta z'} = \vec{z'_{right}} - \vec{z'_{left}}$. Here $\vec{z_{left}}, \vec{z_{right}}$ specify the new histogram fractions.
For an arbitrary color $color \in ColorSpace$ we define $\vec{y_{color}} \in [0,1]^k$ by:

$$\vec{y_{color}} := \frac{1}{\sum_{i=0}^{k-1} (1 - d_{Color}(color, v_i))} \begin{pmatrix} 1 - d_{Color}(color, v_0) \\ 1 - d_{Color}(color, v_1) \\ \vdots \\ 1 - d_{Color}(color, v_{k-1}) \end{pmatrix} \tag{6}$$

where the elements specify the similarity of the color to the respective histogram color. Thus,

if $leftOut$, $rightToLeft$ and $rightIn$ specify the colors according to figure 1, we have:

$$\vec{z'_{right}} = \vec{z_{right}} + \vec{y_{rightIn}} - \vec{y_{rightToLeft}} \quad (7)$$

$$\vec{z'_{left}} = \vec{z_{left}} + \vec{y_{rightToLeft}} - \vec{y_{leftOut}} \quad (8)$$

$$\vec{\Delta z'} = \vec{z'_{right}} - \vec{z'_{left}} = \vec{\Delta z} + \vec{\Delta y} \text{ with } \vec{\Delta y} := \vec{y_{rightIn}} - 2\vec{y_{rightToLeft}} + \vec{y_{leftOut}} \quad (9)$$

Thus,

$$\begin{aligned} d_{Histo}(H'_{left}, H'_{right}) &= \frac{1}{2w^2}\vec{\Delta z'}^T S \vec{\Delta z'} \\ &= \frac{1}{2w^2}(\vec{\Delta z}^T S \vec{\Delta z} + \vec{\Delta z}^T S \vec{y} + \vec{y}^T S \vec{\Delta z} + \vec{y}^T S \vec{y}) \text{ (since } S \text{ is symmetric)} \\ &= \frac{1}{2w^2}(\vec{\Delta z}^T S \vec{\Delta z} + 2\vec{y}^T S \vec{\Delta z} + \vec{y}^T S \vec{y}) \\ &= d_{Histo}(H_{left}, H_{right}) + \Delta u \quad (10) \\ &\quad \text{with } \Delta u := \frac{1}{2w^2}\vec{\Delta y}^T S(2\vec{\Delta z} + \vec{\Delta y}) \quad (11) \end{aligned}$$

The following steps summarize the optimized algorithm:

1. Calculate the initial transition value $t_w := d_{Histo}(H_{left}, H_{right})$ at position $i = w$ on the line by applying equation 1.
2. Determine the three colors $leftOut_i$, $rightToLeft_i$ and $rightIn_i$ at the current position i.
3. Calculate Δu and $\vec{\Delta z'}$.
4. Update the actual transition value: $t_{i+1} = t_i + \Delta u$
5. Repeat steps 2 to 5 for all remaining positions on the line.

The total running time of the algorithm is:

- Scanning the colors along a line: $O(n)$
- Color clustering: $O(nk)$, where n is the number of colors along the line and k is the number of color clusters.
- Calculation of the similarity matrix: $O(k^2)$
- Calculation of the transition values: $O(nk^2)$

Thus, the total running time is $O(nk^2)$. If the user restricts the number of emerging color clusters, which can be done in all practical cases (i.e. k=3 yet yields good results), the running time is $O(n)$.

4 Results

4.1 Real image data

Figure 2 shows a 24 Bit RGB image of a mandrilla mokey. The diagrams on the bottom show the results of calculating the transition function for the line shown in the figure, when using two different window sizes (2, 32). Each diagram displays the colors along the imaginary line on a horizontal beam on the bottom of the respective diagram. At local (and global) maxima a small circle is drawn at the curve of the function.

4.2 Generated image data

To illustrate, that the algorithm is able to detect transitions between different low-level textures we have generated test images with regions of different noisy textures (figure 3).
In case A where the textures have partially different colors the transitions can clearly be found. In case B the textures do not differ in the colors, but in the fractions of the occurrences of colors. Here the transitions can also be determined. In case C the textures neither differ in colors, nor in the fractions of their occurrences. Instead the textures differ by the size of grain. Thus, case C is not a low-level texture. As expected, the algorithm can not find this transition.

4.3 Real Running Times

We have implemented the algorithm with Visual C++, running on a Pentium III, 1 GHz, under Windows 2000. In a non-optimized implementation, the calculation of the transition function along a line with a length of 150 pixels needs 1.7 milliseconds.

5 Conclusion

We have developed a new algorithm to detect transitions (edges) along an imaginary line within a colored image. It has the following desired properties:

1. It detects transitions that evolve from intensity, color or low-level texture discontinuities.
2. It is robust against noise.
3. It does not require the user to specify any color classes.
4. It can operate on different scale spaces.

We have tested the algorithm thoroughly with real and simulated data.
The algorithm can be used as a general building block for higher algorithms. We have already applied the algorithm for the following purposes:

1. **Edge Detection** Within the RoboCup project [1] we have successfully used a specialized version of this algorithm to detect the walls, goals and opponents.

2. **Learning Color Classes** Since the algorithm does not require the user to specify any kind of color classes, it can be used to learn them. For instance, in RoboCup the objects (walls, goals, playing field, ball) are color coded. Calibrating the color classes has been one of the greatest difficulties. With our algorithm the objects can be first recognized by their shape, and next, the colors and even the low-level textures of the objects can be learned automatically.

3. **Higher Feature Detection** The algorithm can be used to build a contour following algorithm. It is then possible to find higher feature as corners and T-junctions which are very important for the detection of part boundaries[8] in human vision.

4. **Segmentation** The algorithm can be used for several segmentation algorithms. For instance the algorithms can be applied for region growing algorithms by detecting for each step to a specific direction, whether a transition is present or not [14][7].

5. **Tracking** By using groups of parallel lines which are installed orthogonal to an object edge the algorithm can be easily used to track moving objects. Combined with a Kalman-filter[13] very accurate results can be achieved.

6. **Snakes, Active Contours and Active Appearance Models** Snakes[10], Active Contours [2] and Active Appearance Models [6] rely on the detection of local features at a priori given locations. Thus, our algorithm can ideally be applied.

7. **Optical Flow** By determining the correlation of the transition functions in consecutive images, optical flow can be determined.

8. **Stereo Vision** The algorithm can be applied for stereo vision by defining equally posed lines in the images of both cameras. By determining the correlation between the transition functions the disparity can be calculated.

Our future goal is to construct a complete vision system that is based on this algorithm. Here, one important issue is how to use scale-space theory[11] and higher feed-back loops to automatically configure the open parameter ”window size” of our algorithm.

References

[1] S. Tadokoro A. Birk, S. Coradeschi. *RoboCup-01: Robot Soccer World Cup V*. Springer, 2001.

[2] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[3] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Syst. J.*, 4(1):25–30, 1965.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8:679–698, 1986.

[5] R. Duda, P. Hart, and D. Stork. Pattern classification, 2000.

[6] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models, 1998.

[7] R. Rojas F. v. Hundelshausen, S. Behnke. An omnidirectional vision system that finds and tracks color edges and blobs. *Birk, A., Coradeschi, S., Tadokoro, S. (eds): RoboCup-01: Robot Soccer World Cup V, Springer, 2001.*, 2001.

[8] Donald D. Hoffman. *Visual Intelligence*. W. W. Norton & Company, Inc., October 1998.

[9] D. H. Hubel and T. N. Wiesel. Brain mechanisms of vision. *Scientific American*, 241(3):130–139, 41–44, 1979.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. of IEEE Conference on Computer Vision*, pages 259–268, London, England, 8-11 1987.

[11] T. Lindeberg. *Scale-Space Theory In Computer Vision*. Kluwer Academic Publishers, 1994.

[12] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings Royal Society of London Bulletin*, 204:301–328, 1979.

[13] P. Maybeck. The Kalman filter: An introduction to concepts. In Autonomous Robot Vehicles. Springer, 1990.

[14] F. v. Hundelshausen. An omnidirectional vision system for soccer robots. Master's thesis, Department of Computer Science, Free University of Berlin, Takustr. 9, D-14195 Berlin, April 2001.

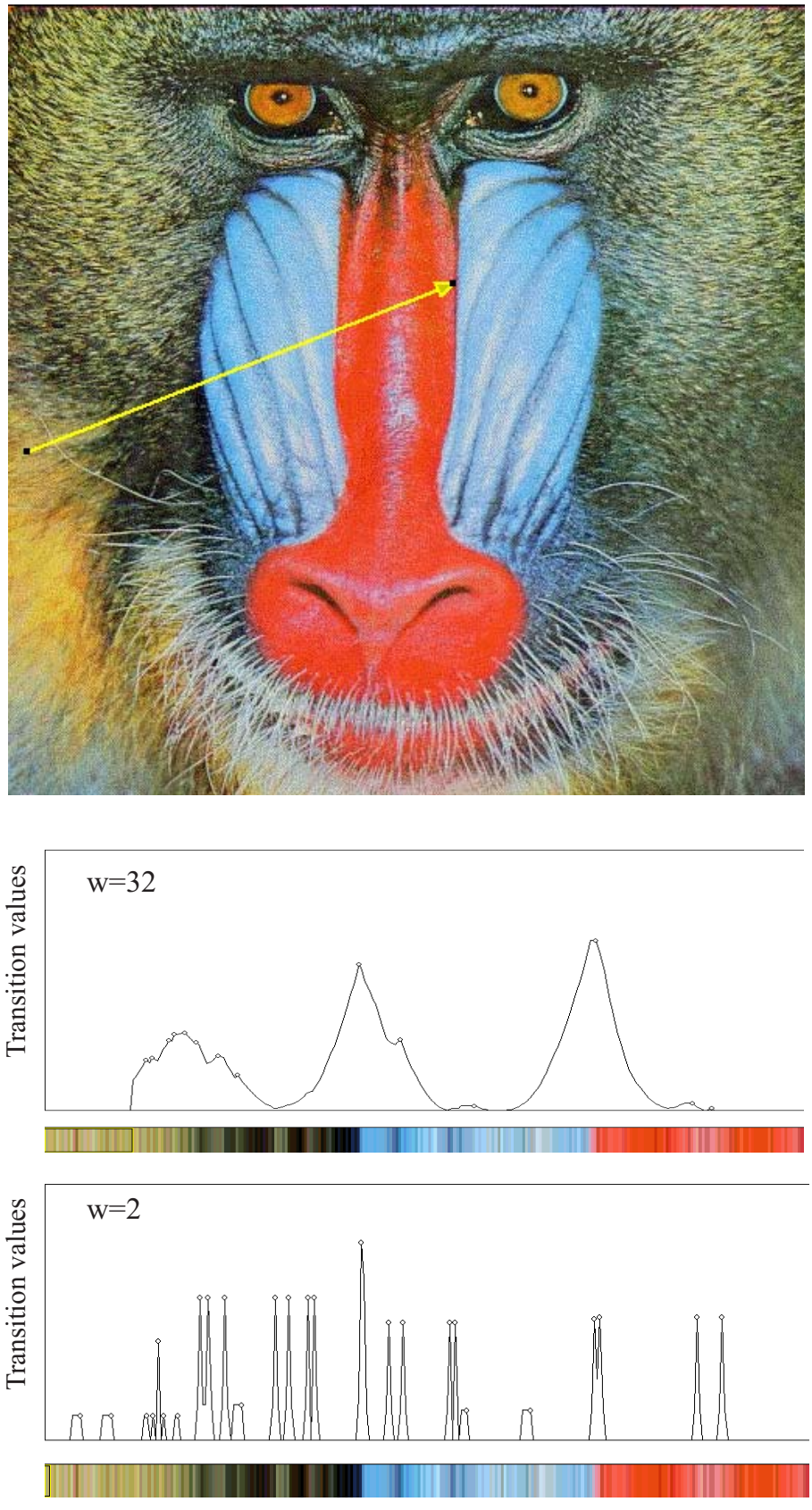


Figure 2: The image is a 24-Bit color photo of a mandrilla. The diagrams on the bottom show two transition functions with different window sizes along the line.

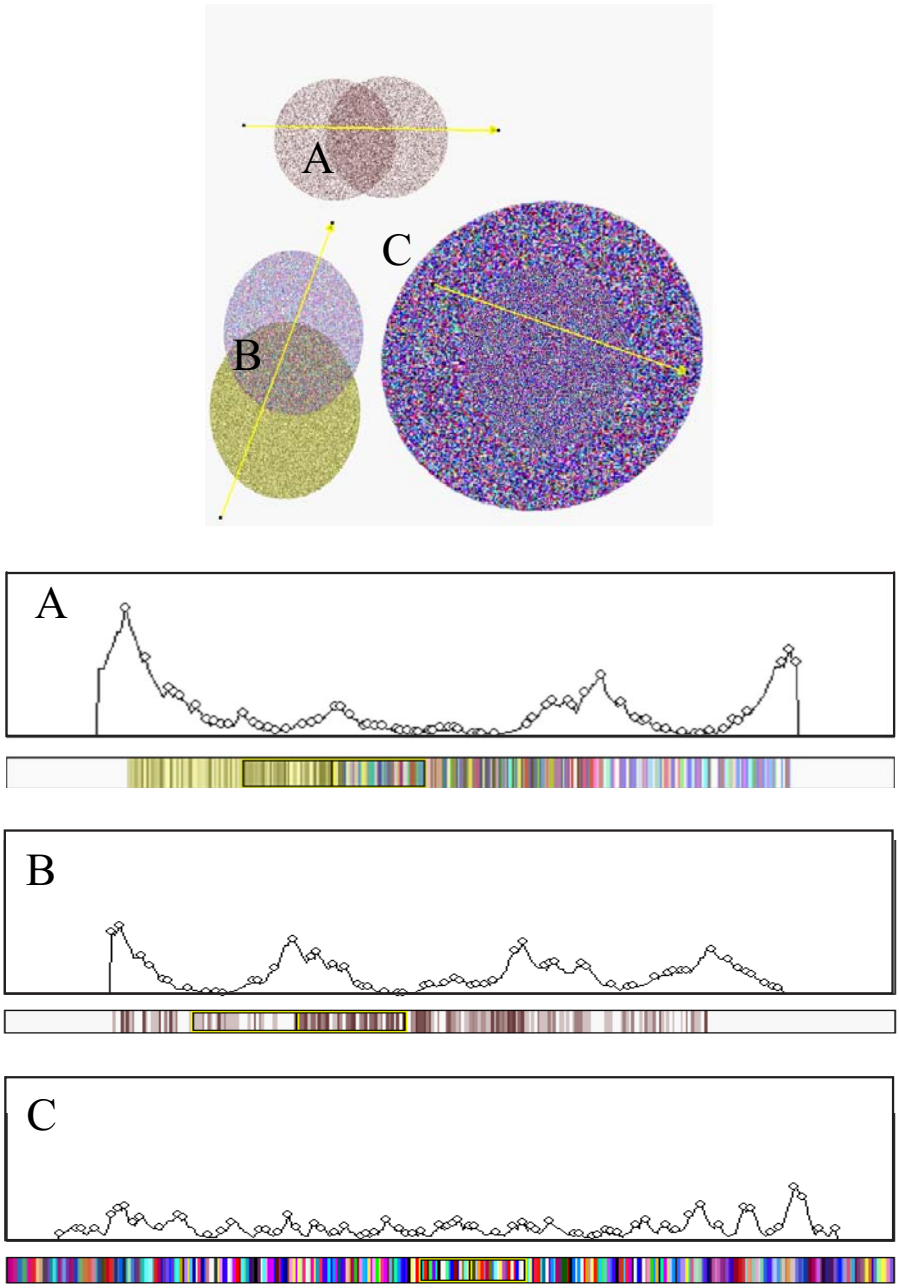


Figure 3: The algorithm is able to find transitions between different low-level textures.

360 x 360 Mosaic with Partially Calibrated 1D Omnidirectional Camera *

Hynek Bakstein and Tomáš Pajdla
Center for Machine Perception, Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague
+420 2 2435 7637, +420 2 2435 7385
e-mail: {bakstein,pajdla}@cmp.felk.cvut.cz

Abstract

We focus on composition of mosaic images using a 360 x 360 mosaic camera. Such a camera is obtained by moving a 1D omnidirectional camera on a circular path. This 1D camera captures a set of light rays in one plane, which should be tangent to the circular path of the camera center and perpendicular to the plane in which this circle lies. However, in practical realizations, this condition does not need to be fulfilled exactly. There are three possible rotations of the 1D omnidirectional camera in space. In this paper we investigate one of them, the in-plane rotation. We show that this rotation breaks the epipolar rectification of the mosaic images and we show that only one point correspondence is required to rectify the images. We also examine the effect of non-square pixels of the CCD cameras used for a practical realization of the 360 x 360 mosaic camera.

Keywords: Omnidirectional cameras, 360 x 360 mosaic

1 Introduction

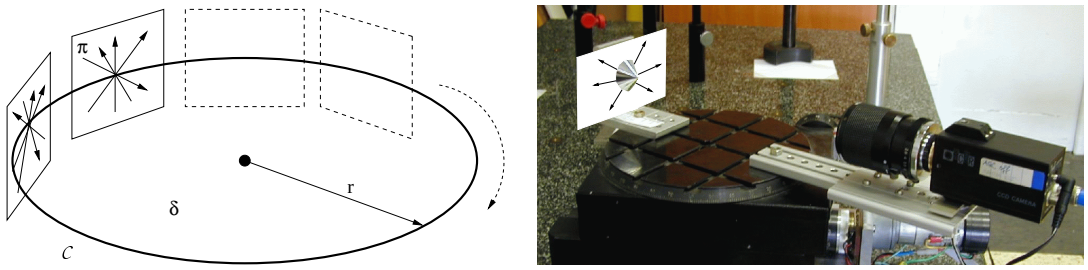
Noncentral cameras, where the light rays do not meet in one common point, are recognized as a new field of research in computer vision. Unlike in case of central cameras, where a consistent theory of geometrical models and relations exists [4], noncentral cameras provide a lot of open problems. For example, the epipolar geometry of some noncentral cameras is described in [7, 5, 8].

We focus on a 3D reconstruction from one particular noncentral camera, the 360 x 360 mosaic [6, 2]. This camera has two interesting properties, the first is that it captures the whole panorama not only in the horizontal direction but also in the vertical direction. The second

*This work was supported by the following grants: MSM 212300013, GAČR 102/01/0971, MŠMT KONTAKT 2001/09.

property is that corresponding points lie on the same image row. Moreover, this camera has a very simple model with only one intrinsic parameter [2].

We can imagine the 360 x 360 mosaic camera as a planar pencil of rays π moved on a circular path $\mathcal{C}$, see Figure 1(a), while this plane is tangent to this path $\mathcal{C}$ and perpendicular to the plane δ in which the circle $\mathcal{C}$ lies. We call this set of rays π a *1D omnidirectional camera*. When all these conditions are fulfilled, the 360 x 360 mosaic camera sees all points outside the cylinder, generated by the circle $\mathcal{C}$, exactly twice in two different rotation positions. This ensures, that there is a disparity between two images of one scene point. The size of the disparity is influenced by the radius of the circle $\mathcal{C}$.



(a) (b)

Figure 1: (a) The camera model and (b), one of its practical realization, showing a plane of light rays reflected by the mirror.

Very important question is, how to actually build such a camera? The circle $\mathcal{C}$ is realized by a turntable. The key issue is how to create the 1D omnidirectional camera. Originally [6], two realizations were proposed. One employed a pinhole camera and a hat-like shaped mirror, the other one used a telecentric lens and a conical mirror with the apex angle equal $90°$. We decided to test the latter realization, shown in Figure 1(b). We also proposed another realization employing a fish eye lens with field of view larger than $180°$ [1] instead of the mirror. This realization provides simpler setup at almost the same cost. It should be noted that the fish eye lens has to be calibrated in order to select the planar pencil of light rays π.

It is somehow confusing that the word *camera* has several meanings in this text. There is the *360 x 360 mosaic camera* or simply *mosaic camera* which produces a mosaic image pair, the *1D omnidirectional camera*, which captures a planar pencil of light rays π and is rotated on a circular path $\mathcal{C}$ to produce the 360 x 360 mosaic camera, and finally the word *camera* alone means ordinary CCD camera, which is employed in the practical realization of the 1D omnidirectional camera. The first two meaning describe abstract cameras. The emphasized term should distinguish which device is discussed in the text. The same applies for the word *image*. By a *mosaic image* we understand the output from the mosaic camera while *image* stands for output from the camera realizing the 1D omnidirectional camera and it is represented by 2D matrix. The image from the 1D omnidirectional camera is generally represented as an ellipse (in ideal case as a circle) in this image.

The camera observes a scene reflected in the mirror, see Figure 1(b). When selecting a circle in the image centered on the mirror tip, one plane π of pixels is determined and this circle represents an image from the 1D omnidirectional camera. In case of a fish eye lens, the light

rays from π are directly perspectively imaged to a circle in the image. As noted before, the mosaic camera observes each point in the scene exactly twice, therefore it is natural to split the plane π, with respect to a line parallel to vertical image axis and passing through the image of the tip of the cone (the image center in case of the fish eye), into two parts which are used to compose the two mosaic images, see Figure 2.

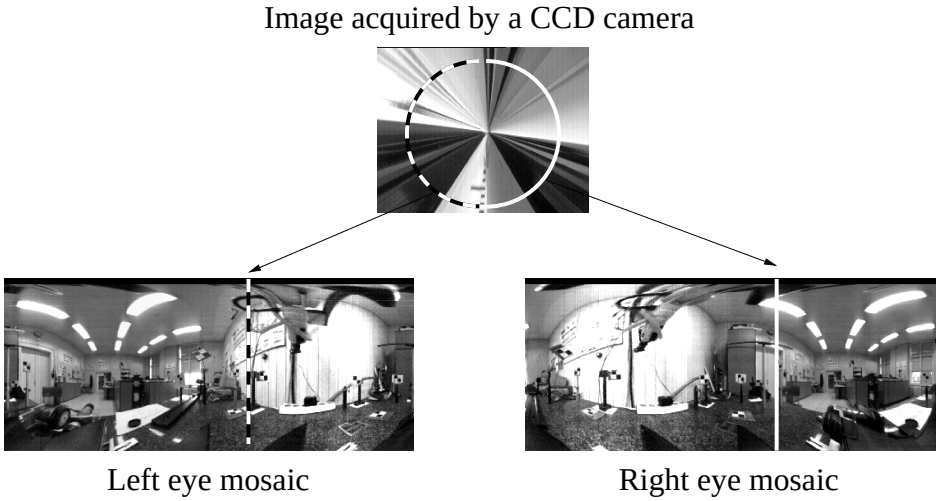


Figure 2: Mosaic composition from slices.

However, the above observations are valid only if the plane π is actually perpendicular to the plane δ and tangent to the circle $\mathcal{C}$. Moreover, the procedure explained in Figure 2 is valid only for a camera with square pixels. In case of non square pixels we have to select an ellipse instead of a circle. The vertical image axis must also be perpendicular to the plane δ, otherwise we have to use a different line to divide the ellipse into two parts. All the previous conditions do not have to be satisfied in the real setup due to some imperfections.

When we say that the 1D omnidirectional camera is *partially calibrated*, we mean that we know how to select the proper pixels in the 2D image which correspond to the planar pencil of light rays, but we do not know the coordinate system in this pencil. This means that we do not know which pixel represents a light ray perpendicular to the plane δ. It also means that we do not know whether the vertical image axis is perpendicular to the plane δ.

In this paper, we focus on two particular issues, the problem of non square pixels, discussed in Section 2, and the case when the vertical image axis is not perpendicular to the plane δ, see Section 3. We call this situation *in-plane rotation*. The other two issues are left for future work. Experimental results are presented in Section 4.

2 Non-square pixels

The term *non-square pixels* can have two meanings. Either it can mean that there is a difference in scale of the image axes, or the angle between the axes is not equal to $90°$. General calibration models [10] assume that both these problems can occur simultaneously. However, modern CCD

cameras have hardly the angle between the image axes different from 90°. Therefore we focus only on the difference in the scale of the image axes.

This difference causes that circles in an ideal (undistorted) image become ellipses in the observed (distorted) image, see Figure 3. The transformation between the distorted and the undistorted image can be expressed in a matrix form as:

$$\mathbf{K^{-1}} = \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \beta & -\beta v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \tag{1}$$

This matrix $\mathbf{K}$ is a simplified intrinsic calibration matrix of a pinhole camera [4]. The displacement of the center of the image is expressed by terms u_0 and v_0, the skewness of the image axes is neglected here. Provided with this matrix, we can select the proper ellipse in the image

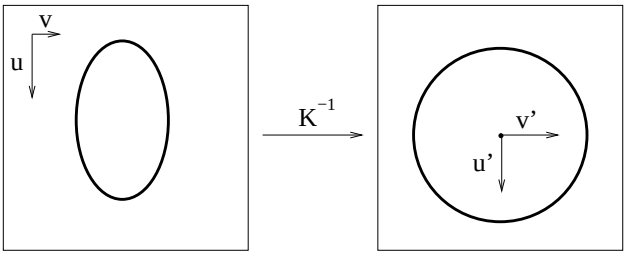


Figure 3: A circle in the image plane is distorted due to a different length of the axes. Therefore we observe an ellipse instead of a circle in the image.

which correspond to the planar pencil of light rays π, representing the image from the 1D omnidirectional camera. Without this knowledge, we cannot select the proper rays and we cannot construct the mosaic camera.

3 In-plane rotation

In-plane rotation occurs when the vertical image axis is not perpendicular to the plane δ. This can happen when the CCD chip of the camera or even the camera itself is slightly tilted. It should be noted that the plane π is still perpendicular to the plane δ and tangent to the circle $\mathcal{C}$ in this case. It only the image axes are rotated in the plane π, thus the term *in-plane rotation*. Therefore, if we choose the line l,which is used for splitting the ellipse corresponding to the light rays in plane π, to be perpendicular to the vertical image axis, it will be not perpendicular to the plane δ. The ellipse has to be splitted with respect to some other line l', as it is depicted in Figure 4(a). The angle between these two lines is the same as the angle of the in-plane rotation.

What is the effect of the in-plane rotation? The splitting of the ellipse means that from one acquisition step we get two mosaic images. Each point in the scene is visible twice, once in each of the images. When an in-plane rotation occurs, two images of one point can be observed in one mosaic image, see Figure 4(b). This is due to the fact, that 360 x 360 mosaic images have a topology of a torus [9].

It can be observed in Figure 4 that the left eye mosaic is in the correct case composed by segments 4 and 3 and the right eye mosaic consists of segments 1 and 2. In case of the in-plane rotation, the segment 3 is moved from the bottom of the left eye mosaic to the bottom of the right eye mosaic. The order of rows in this segment is reversed. Similar situation occurs with the segment 1. The angle of the in-plane rotation determines the size of these two segments. Since we know that the row coordinates of two corresponding images of one points has to be the same, we can compute the number of rows which have to be shifted back.

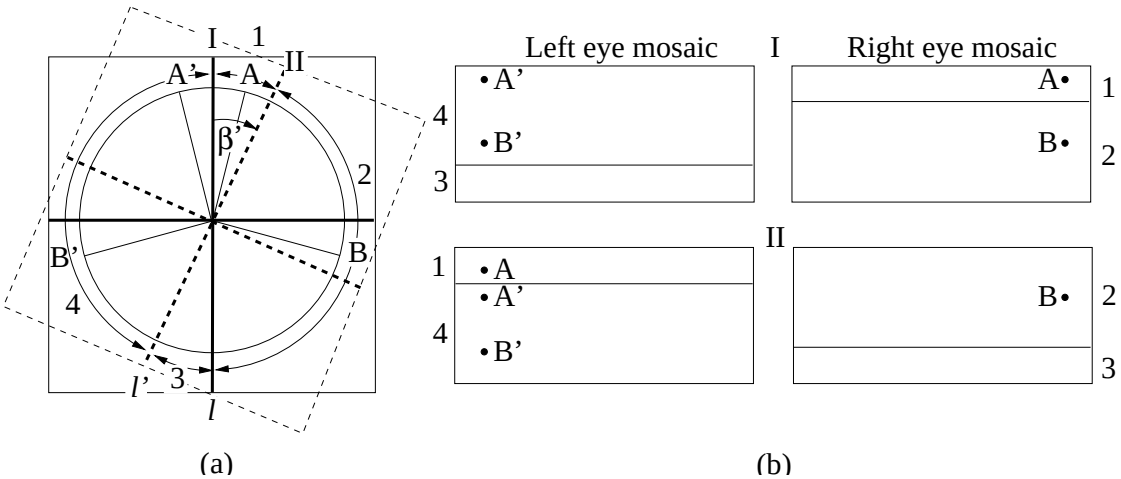


Figure 4: Effect of an in-plane rotation, see text for a description.

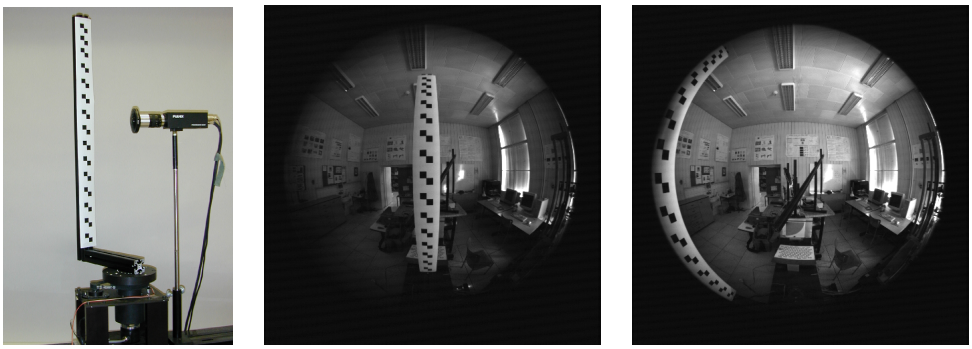
4 Experimental results

During our attempts to build a 360 x 360 mosaic camera, we found that corresponding points in the two mosaic images do not lie on the same image row. Instead, their coordinates exhibited almost the same difference, which corresponds to some in-plane rotation. We performed the following experiment, where the 1D omnidirectional camera was realized by a fish eye lens. Similar procedure is valid also when the mirror is used to create the 1D omnidirectional camera.

A Pulnix high resolution camera was equipped with a standard $12.5mm$ lens and a Nikon FC-E8 fish eye converter. In the initial position, the camera was pointing with its optical axis at one of the calibration points, see Figure 5(a). Then we rotated the calibration target from -90° to 90° with a 10° step. In each step, we captured an image of the calibration target and extracted coordinates of the selected calibration point. Figure 5(b) show an image from the camera corresponding to no rotation of the target and Figure 5(c) shows the target rotated by 90°. Note the significant distortion which does not allow to use automatic detection of points. Another shape of the markers should probably be used.

Figure 6 shows coordinates of the detected images of this point moving in space. It can be noticed that the point is not moving on a straight horizontal line nor on a curve, which would correspond to ideal case or an incorrectly estimated image center respectively. Instead,

the point moves on a rotated straight line. This indicates, that the CCD chip of our camera is slightly tilted, which results in an in-plane rotation of our 1D omnidirectional camera.



(a) (b) (c)

Figure 5: (a) Experimental setup. (b) One of the images. (c) The calibration target in the leftmost position, note the significant distortion.

We verified this with a standard calibration procedure [3] and found that there is really a tilt of 0.288° of the CCD chip present, which results in the change in the v coordinate up to almost 5 pixels in a 1000×1000 image. Although the rotation seems really small, the higher the resolution, the bigger the influence of the tilt. Also in the resulting mosaic, the corresponding images of one scene point were not on the same image row, see Figure 7. After the epipolar rectification compensating the in-plane rotation, the points moved to the same image row, as it is depicted in Figure 7.

Figure 8 shows this difference for selected corresponding images of scene points in case of rectified mosaics. The points were detected manually to ensure that also points from the top and bottom part of the mosaic image will be compared, see Figure 9(a) and Figure 9(b) for images of these points in the left and right mosaic image respectively. Due to a high distortion in the top and bottom parts of the mosaic images, automatic matching algorithms based on correlation would fail on these points. Note that the differences in the row coordinate are smaller than one pixel.

5 Conclusion

We have demonstrated the effects of two sources of distortion which can arise in realization of the 360 x 360 mosaic camera. We have shown that the effect of non-square pixels can be compensated by selecting an appropriate ellipse instead of a circle in the image. We discussed the effect of the in-plane rotation on the resulting mosaic images. It shows up that only one point correspondence is needed to rectify the mosaic images. Other two rotations of the 1D omnidirectional camera with respect to the plane constraining the circular motion path are left for future work.

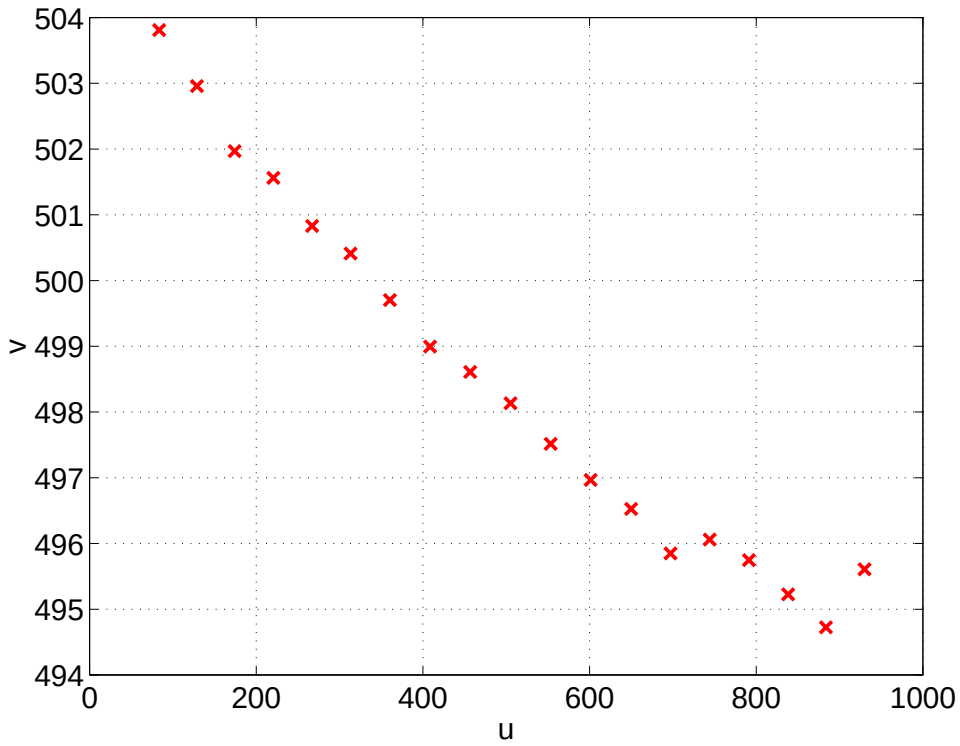


Figure 6: Image coordinates of the calibration point for rotations of the calibration chart from $-90°$ to $90°$.

References

[1] H. Bakstein and T. Pajdla. Omnivergent stereo-panoramas with fish-eye lens. Research Report CTU–CMP–2001–22, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, August 2001.

[2] Hynek Bakstein and Tomáš Pajdla. 3D reconstruction from 360 x 360 mosaics. In A. Jacobs and T. Baldwin, editors, *Proceedings of the CVPR'01 conference*, volume 2, pages 72–77, Loas Alamitos, CA, December 2001. IEEE Computer Society.

[3] O. Faugeras. *Three-dimensional computer vision — A geometric viewpoint*. MIT Press, 1993.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] F. Huang, S. K. Wei, and R. Klette. Epipolar geometry in polycentric panoramas. In R. Klette, T. Huang, and G. Gimel'farb, editors, *Multi-Image Analysis : Proceedings of the 10th International Workshop on Theoretical Foundations of Computer Vision*, Lecture Notes in Computer Science, pages 39–50, Berlin, Germany, March 2000. Springer.

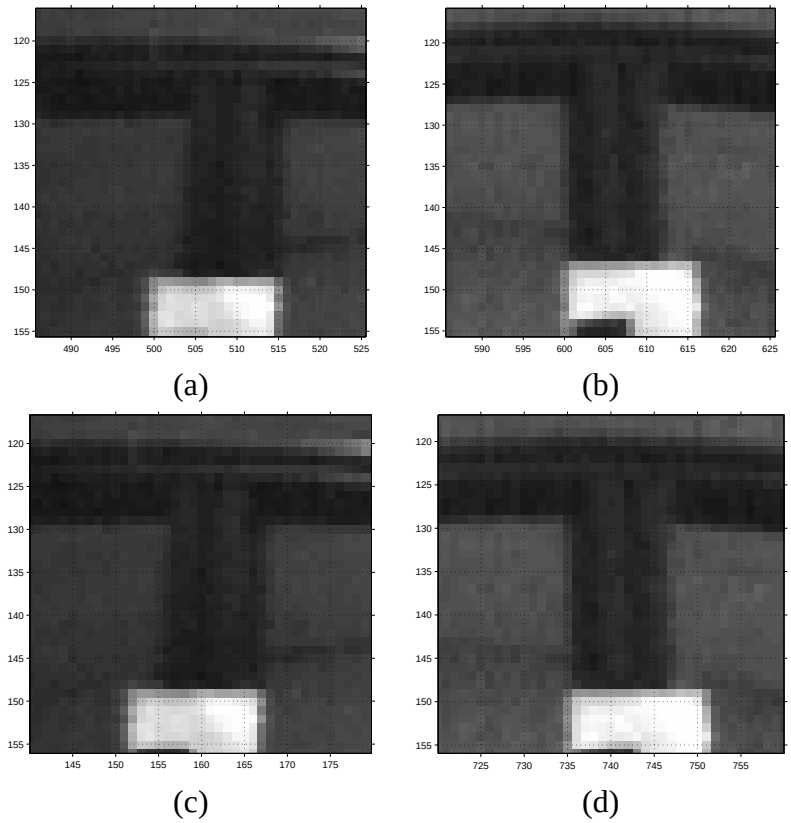


Figure 7: Detail of a corresponding pair of points (a) and (c) in the right mosaic and (b) and (d) in the left mosaic representing the difference from the ideal case, where the corresponding points lie on the same image row. The upper row shows point position before epipolar rectification, note the difference in the row. The lower row shows point position after the rectification Note that the points lie on the same image row.

[6] S. K. Nayar and A. Karmarkar. 360 x 360 mosaics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), Hilton Head, South Carolina*, volume 2, pages 388–395, June 2000.

[7] T. Pajdla. Epipolar geometry of some non-classical cameras. In B Likar, editor, *Proceedings of Computer Vision Winter Workshop*, pages 223–233, Ljubljana, Slovenia, February 2001. Slovenian Pattern Recorgnition Society.

[8] S. Seitz. The space of all stereo images. In J. Little and D. Lowe, editors, *ICCV'01: Proc. Eighth IEEE International Conference on Computer Vision*, volume 1, pages 26–35, Los Alamitos, CA, USA, July 2001. IEEE Computer Society.

[9] H.-Y. Shum, A. Kalai, and S. M. Seitz. Omnivergent stereo. In *Proc. of the International Conference on Computer Vision (ICCV'99), Kerkyra, Greece*, volume 1, pages 22–29, September 1999.

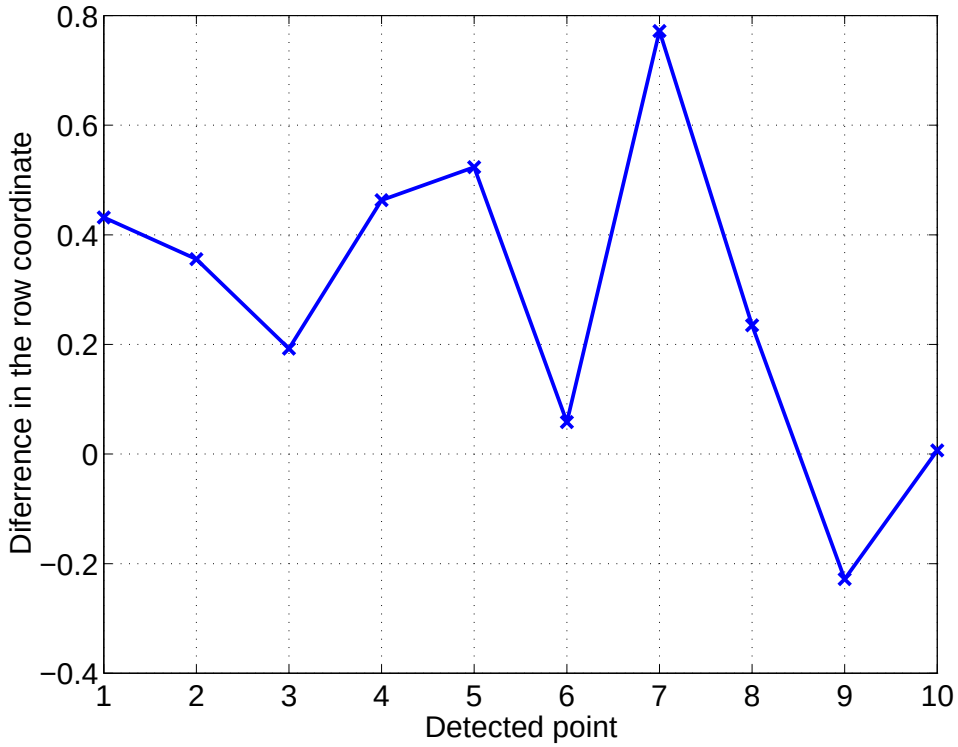


Figure 8: Difference in the row coordinates of 10 manually selected points.

(a) (b)

Figure 9: (a) Right eye and (b) left eye mosaics. Manually detected points are marked by 'x'.

[10] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine metrology using off-the-shelf cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

Calibration of a fish eye lens with field of view larger than 180° *

Hynek Bakstein and Tomáš Pajdla
Center for Machine Perception, Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague
+420 2 2435 7637, +420 2 2435 7385
e-mail: {bakstein,pajdla}@cmp.felk.cvut.cz

Abstract

We present a complete step-by-step approach to calibration of ultra wide angle fish-eye lenses with the angle of view larger than $180°$. Such a large field of view is necessary for some applications such as the 360 x 360 mosaicing. Recently, a Nikon FC-E8 fish eye lens converter with the field of view equal $183°$ become available. In this paper, we propose its model, suggest a calibration procedure, and demonstrate its use in a mosaicing application. First of all we propose a general model of a camera with field of view larger than $180°$. Then, we identify the structure and the parameters of the mapping between the incoming light rays and pixels for the Nikon FC-E8 converter. Finally, we present a complete camera calibration method from a known calibration target.

Keywords: Camera calibration, wide angle lenses

1 Introduction

Large field of view (FOV) is useful for some computer vision applications such as selfcalibration where it provides better conditioned views and less degenerate situations [11, 10]. Several ways to enlarge the FOV exist. Mirrors, lenses, moving parts, or a combination of the previous can be employed for this purpose. In this paper we focus on the use of a special lens, the Nikon FC-E8 fish eye converter [3], which provides FOV of $183°$. This FOV allows us to employ this lens in building of a 360 x 360 mosaic [7]. We mounted this lens on a Pulnix digital camera equipped with a standard 12.5mm lens as it is depicted in Figure 1. Our experiments also show that such a lens provides better results than mirrors, which were often used to build 360 x 360 mosaics [7]. Focusing of the lens is easier than focusing on the mirror and also the setup of the mosaicing camera is simpler.

*This work was supported by the following grants: MSM 212300013, GAČR 102/01/0971, MŠMT KONTAKT 2001/09.

Figure 1: Nikon FC-E8 fish eye converter mounted on a Pulnix digital camera with a standard 12.5mm lens.

For many computer vision tasks, the relationship between the light rays entering the camera and pixels in the image has to be known. In order to find this relationship, the camera has to be calibrated. A suitable camera model has to be chosen for this task. It turns out that the pinhole camera model with a planar retina, see Figure 2, is not sufficient for sensors with large FOV [5]. Previous approaches used planar retina and pinhole model [1, 2, 12, 13]. In [9], a stereographic projection was employed but the experiments were evaluated on lenses with FOV smaller than $180°$. We introduce a method for calibration from a single image of one known 3D calibration target with iterative refinement of parameters of our camera model with a spherical retina, depicted in Figure 2.

In the next section, we introduce a camera model with a spherical retina. Then we discuss various models describing the relationship between the light rays and pixels in Section 3. Section 4 is devoted to the determination of this model for the case of Nikon FC-E8 converter. A summary of the presented method is given in Section 5. Experimental results are presented in Section 6.

2 Camera Model

The camera model describes how a 3D scene is transformed into a 2D image. It has to incorporate the orientation of the camera with respect to some scene coordinate system and also the way how the light rays in the camera centered coordinate system are projected into the image. The orientation is expressed by extrinsic camera parameters while the latter relationship is determined by intrinsic parameters of the camera.

Intrinsic parameters can be divided into two groups. The first one includes the parameters of the mapping between the rays and ideal orthogonal square pixels. We will discuss these parameters in the next section. The second group contains the parameters describing the relationship between ideal orthogonal square pixels and the real pixels of image sensors.

Let (u, v) denote coordinates of a point in the image measured in an orthogonal basis as shown in Figure 3. CCD chips often have a different spacing between pixels in the vertical and the horizontal direction. This results in images unequally scaled in the horizontal and vertical direction. This distortion causes circles to appear as ellipses in the image, as shown in

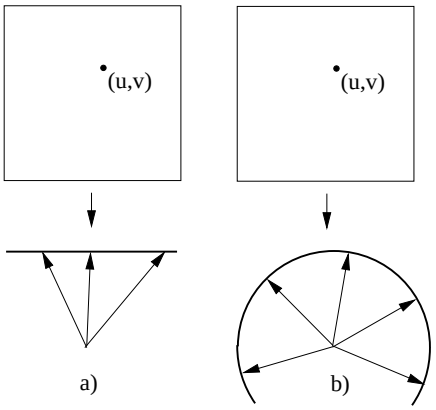


Figure 2: From image coordinates to light rays: (a) a directional and (b) an omnidirectional camera.

Figure 3. Therefore, we introduce a parameter β representing the ratio between the scales of the horizontal respectively the vertical axis. A matrix expression of the distortion can be written in the following form:

$$\mathbf{K^{-1}} = \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \beta & -\beta v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \tag{1}$$

This matrix is a simplified intrinsic calibration matrix of a pinhole camera [4]. The displacement of the center of the image is expressed by terms u_0 and v_0, the skewness of the image axes is neglected in our case, because cameras usually have orthogonal pixels.

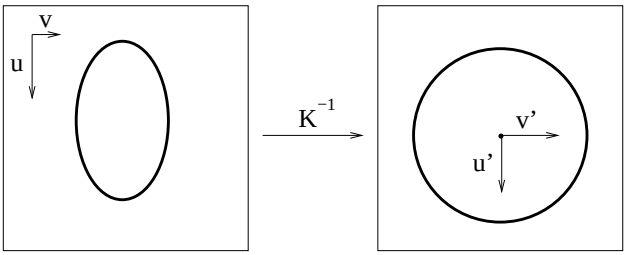


Figure 3: A circle in the image plane is distorted due to a different length of the axes. Therefore we observe an ellipse instead of a circle in the image.

3 Projection Models

Models of the projection between the light rays and the pixels are discussed in this section. Most commonly used approach is that these models are described by a radially symmetric function that maps the angle θ between the incoming light ray and the optical axis to some distance r from the image center, see Figures 7 and 7(b). This function typically has one parameter k. As

it was stated before, the perspective projection, which can be expressed as $r = k \tan \theta$, is not suitable for modeling cameras with large FOV. Several other projection models exist [5]:

- stereographic projection $r = k \tan \frac{\theta}{2}$,
- equidistant projection $r = k\theta$,
- equisolid angle projection $r = k \sin \frac{\theta}{2}$, and
- sine law projection $r = k \sin \theta$.

Figure 4 shows graphs of the above projection functions for angle θ varying from 0 to 180 degrees. It can be noticed that perspective projection cannot cope with angles θ near $90°$. It can also be noticed that most of the models can be approximated with an equidistant projection for a smaller angle θ. However, when the FOV of the lens increases, the models differ significantly. In the next section we describe a procedure for selecting the appropriate model for Nikon FC-E8 converter.

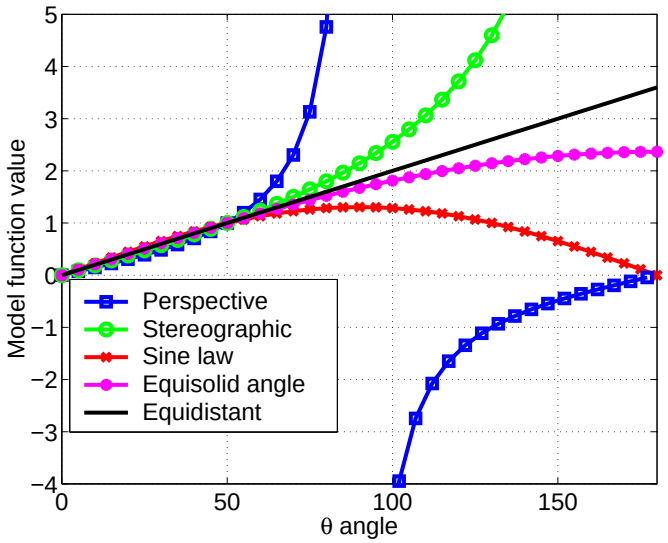
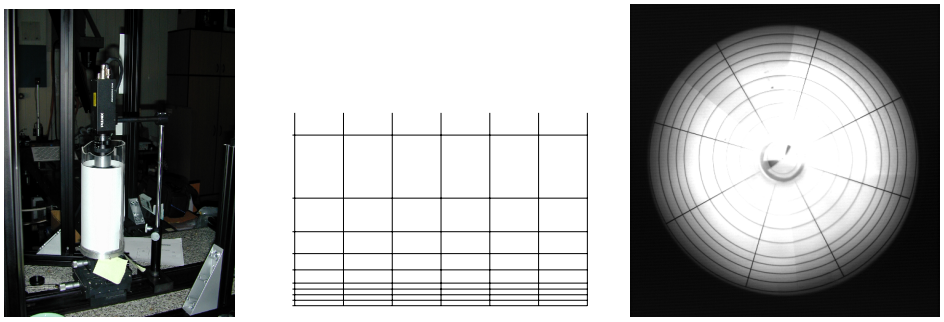


Figure 4: Values of projection models function for angle θ in range of 0 and 180 degrees.

4 Model Determination

In order to derive the projection model for Nikon FC-E8, we have investigated how light rays with constant increment in the angle θ are imaged on the image plane. We performed the following experiment. The camera was observing a cylinder with circles seen by light rays with known angle θ, as it is depicted in Figure 5(a). These circles correspond to an increment in the angle θ set to $5°$ for rays imaged to the peripheral parts of the image ($\theta = 90°..70°$) and to $10°$ for the rays imaged to the central part of the image. Figure 5(b) show the grid which after

wrapping around a cylinder produced the circles. Figure 5(c) shows an image of this cylinder. It can be seen that circles are imaged to approximate circles and that constant increment in angles results in slowly increasing increment in radii of the circles in the image. Note that the circles at the border have angular distance $5°$, while the distance near the center is $10°$.



(a) (b) (c)

Figure 5: (a) Camera observing a cylinder with a calibration pattern (b) wrapped around the cylinder. Note that the lines corresponds to light rays with an increment in the angle θ set to $5°$ (the bottom 4 intervals) and $10°$ (the 5 upper intervals). (c) Image of circles with radii set to a tangent of a constantly incremented angle results in concentric circles with almost constant increment in radii in the image.

We fitted all of the models mentioned in the previous section to detected projections of the light rays into the image. The stereographic projection with two parameters: $r = a \tan \frac{\theta}{b}$ provided the best fit but there was still a systematic error, see Figure 6. Therefore, we extended the model which resulted in a combination of the stereographic projection with the equisolid angle projection. This improved model is identified by four parameters, see Equation 3, and provides the best fit with no systematic error, as it is depicted in Figure 6. An initial fit of the parameters is discussed in the following section.

5 Complete Camera Model

Under the above observations, we can formulate the model of the camera. Provided with a scene point $\mathbf{X} = (x, y, z)^T$, we are able to compute its coordinates $\tilde{\mathbf{X}} = (x, y, z)^T$ in the camera centered coordinate system:

$$\tilde{\mathbf{X}} = \mathbf{R}\mathbf{X} + \mathbf{T} \ , \tag{2}$$

where $\mathbf{R}$ represents a rotation and $\mathbf{T}$ stands for a translation. The standard rotation matrix $\mathbf{R}$ has three degrees of freedom and $\mathbf{T}$ is expressed by the vector $\mathbf{T} = (t_1, t_2, t_3)^T$.

Then, the angle θ, see Figure 7(a), between the light ray through the point $\tilde{\mathbf{X}}$ and the optical axis can be computed. This angle determines the distance r of the pixel from the center of the image:

$$r = a \tan \frac{\theta}{b} + c \sin \frac{\theta}{d} \ , \tag{3}$$

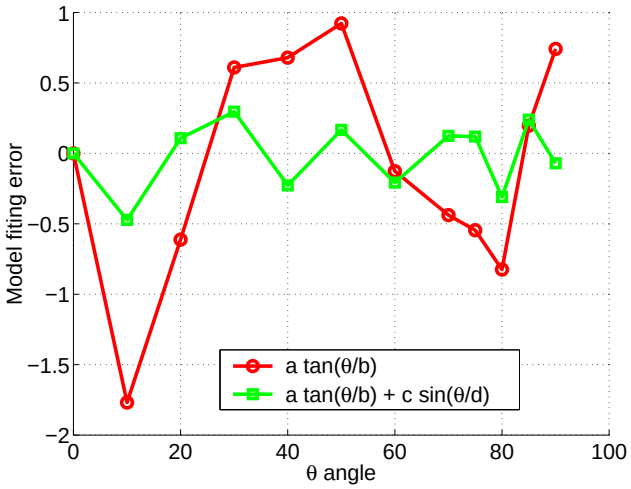
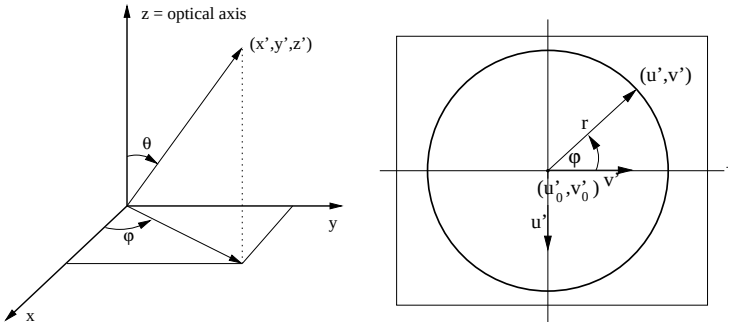


Figure 6: Model fit error for stereographic and combined stereographic and equisolid angle projection. Parameters a, b, c, and d were estimated in an optimization procedure.

where a, b, c, and d are parameters of the projection model.



(a) (b)

Figure 7: (a) Camera coordinate system and its relationship to the angles θ and φ (b) From polar coordinates (r, φ) to orthogonal coordinates (u', v').

Together with the angle φ between the light ray reprojected to xy plane and the x axis of the camera centered coordinate system, the distance r is sufficient to calculate the pixel coordinates $\mathbf{u}' = (u', v', 1)$ in some orthogonal image coordinate system, see Figure 7(b), as

$$u' = r \cos \varphi \quad (4)$$

$$v' = r \sin \varphi \ . \quad (5)$$

In this case the vector $\mathbf{u}'$ does not represent a light ray from the camera center like in a pinhole camera model, instead it is just a vector augmented by 1 so that we can write an affine transform of the image points compactly by one matrix multiplication (6).

Real pixel coordinates $\mathbf{u} = (u, v, 1)$ can then be obtained as

$$\mathbf{u} = \mathbf{K}\mathbf{u}' \ . \quad (6)$$

The complete camera model parameters including extrinsic and intrinsic parameters can be recovered from measured coordinates of calibration points by minimizing an objective function

$$J = \sum_{i=1}^{N} \|\tilde{\mathbf{u}} - \mathbf{u}\| \ , \tag{7}$$

where $\|...\|$ denotes the Euclidean norm, N is the number of points, $\tilde{\mathbf{u}}$ are coordinates of points measured in the image, and $\mathbf{u}$ are their coordinates reprojected by the camera model. A MATLAB implementation of the Levenberg-Marquardt [6] minimization was employed in order to minimize the objective function (7).

The rotational matrix $\mathbf{R}$ has three degrees of freedom, as well as the vector of translation $\mathbf{T}$, see (2). The image center, scale ratio of the image axes β, and the four parameters of the mapping between the light rays and pixels (3) give 7 intrinsic parameters. Therefore, our model is identified by 13 parameters.

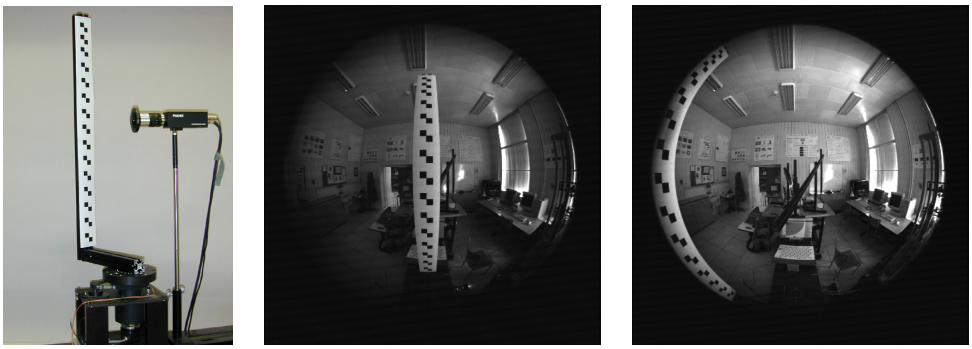
When minimizing the objective function (7), we set the image center to the center of the circle (ellipsis) surrounding the image, see Figure 5. This is possible because the Nikon FC-E8 lens is so called circular fish eye, where this circle is visible. Assuming that the mapping between the light rays and pixels (3) is radially symmetric, this center of the circle should be the image center. Parameters of the model were set to an ideal stereographic projection, which means that $b = 2$, $c = 0$, $d = 1$, and a was set using the ratio between the coordinates of points corresponding to the light rays with the angle θ equal to 0 and 180 degrees. The value of the β parameters was set to 1. The camera position was set to be in the center of the scene coordinate system with the z axis coincident with the optical axis of the camera.

6 Experimental Results

We performed two calibration experiments. In the first experiment, the calibration points were located on a cylinder around the optical axis and the camera was looking down into that cylinder, see Figure 5(a). The points had the same depth for the same value of θ. The second experiment employed a 3D calibration object with points located on a half cylinder. The object was realized such that a line of calibration points was rotated on a turntable, as it is depicted in Figure 8. Here, the points with the same angle θ had different depths.

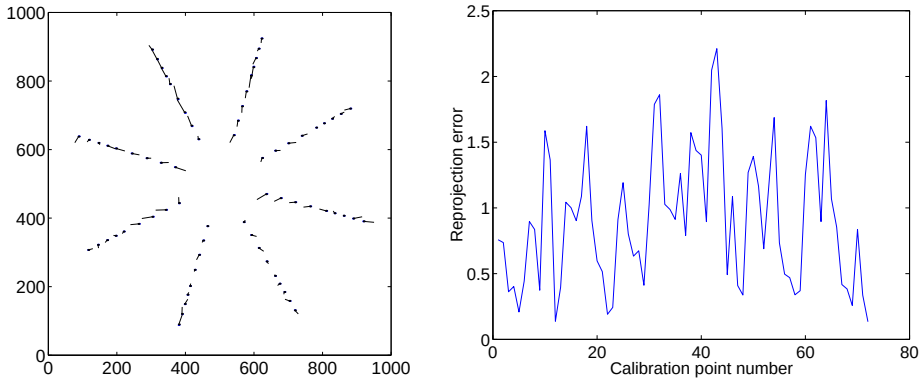
The first experimental setup was also used to determine the projection model, as it is described in Section 4. The total number of 72 points was manually detected. One half of them was used for the estimation of the parameters while the second half was used for the verification. The same approach was also used in the second experiment, where the number of calibration points was 285. Again, all points were detected manually.

Figure 9(a) shows the reprojection of points, computed with parameters estimated during the calibration, compared with their coordinates detected in the image. The lines represent the errors between the respective points scaled 20 times to make the distances clearly visible. The same error is shown in Figure 9(b) for all the points. It can be noticed that the error does not show any significant systematic dependence.



(a) (b) (c)

Figure 8: (a) Experimental setup for the half cylinder experiment. (b) One of the images with the calibration target in the middle of the image. (c) The calibration target is located 90° left from the camera, note the significant distortion.



(a) (b)

Figure 9: (a) Reprojection of points for the cylinder experiment. The distances between the reprojected and the detected points are scaled 20 times. (b) Reprojection error for each point.

Similar graphs illustrate the results from the second experiment. Figure 10 shows the comparison between the reprojected points and their coordinates detected in the image. Again, the lines representing the distance between these two sets of points are scaled 20 times.

Figure 11(a) depicts this reprojection error for each calibration point. Note that the error is bigger for points in the corners of the image, which is natural, since the resolution here is higher and therefore one pixel corresponds to a smaller change in the angle θ. However, it can be seen in Figure 10 that the reprojection error is nearly random.

To verify the randomness of the error, we performed the following test. Because the points in the image were detected manually, we suppose that the detection error has normal distribution in both image axes. Therefore, a sum of squares of these errors, normalized to unit variance, should be described by a χ^2 distribution [8]. Figure 11(b) shows a histogram of detection errors together with a graph of a χ^2 density. Note that χ^2 distribution describes well the calibration error distribution.

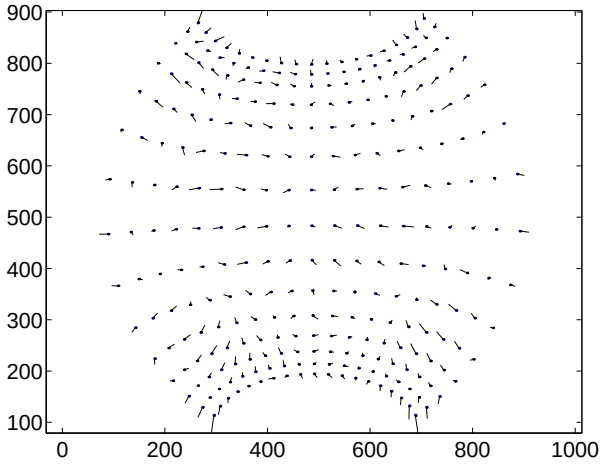
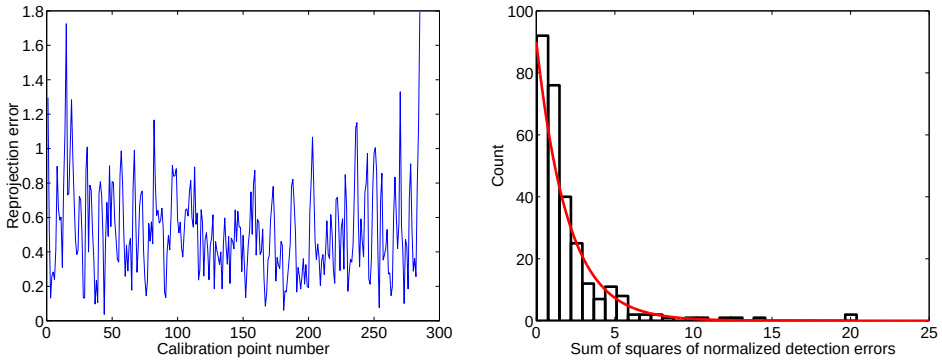


Figure 10: Reprojection of points for the half cylinder experiment. The distances between the reprojected and the detected points are scaled 20 times.



(a) (b)

Figure 11: (a) Reprojection error for each point for the half cylinder experiment. (b) Histogram of sum of squares of normalized detection errors together with a χ^2 density marked by the curve.

7 Conclusion

We have proposed a camera model for lenses with FOV larger than $180°$. The models is based on employment of a spherical retina and a radially symmetrical mapping between the incoming light rays and pixels in the image. We proposed a method for identification of the mapping function, which led to a combination of two mapping functions. A complete calibration procedure, involving a single image of a 3D calibration target, is then presented. Finally, we demonstrate the theory in two experiments, all using the Nikon FC-E8 fish eye converter. We believe that the ability to correctly describe and calibrate the Nikon FC-E8 fish eye lens converter opens a way to many new application of very wide angle lenses.

References

[1] A. Basu and S. Licardie. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16(4):433–441, 1995.

[2] S. S. Beauchemin, R. Bajcsy, and Givaty G. A unified procedure for calibrating intrinsic parameters of fish-eye lenses. In *Vision Interface (VI 99)*, pages 272–279, May 1999.

[3] Nikon Corp. Nikon www pages: http://www.nikon.com, 2000.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] Fleck M. M. Perspective projection: the wrong imaging model. Technical Report TR 95-01, Comp. Sci., U. Iowa, 1995.

[6] J.J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis, Lecture Notes in Mathematics 630*, pages 105–116. Springer Verlag, 1977.

[7] S. K. Nayar and A. Karmarkar. 360 x 360 mosaics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), Hilton Head, South Carolina*, volume 2, pages 388–395, June 2000.

[8] A. Papoulis. *Probability and Statistics*. Prentice-Hall, 1990.

[9] D. E. Stevenson and M. M. Fleck. Robot aerobics: Four easy steps to a more flexible calibration. In *International Conference on Computer Vision*, pages 34–39, 1995.

[10] T. Svoboda, T. Pajdla, and V. Hlaváč. Epipolar geometry for panoramic cameras. In Hans Burkhardt and Neumann Bernd, editors, *the fifth European Conference on Computer Vision, Freiburg, Germany*, number 1406 in Lecture Notes in Computer Science, pages 218–232, Berlin, Germany, June 1998.

[11] Tomáš Svoboda, Tomáš Pajdla, and Václav Hlaváč. Motion estimation using central panoramic cameras. In Stefan Hahn, editor, *IEEE International Conference on Intelligent Vehicles*, pages 335–340, Stuttgart, Germany, October 1998. Causal Productions.

[12] R. Swaminathan and S.K. Nayar. Non-metric calibration of wide-angle lenses. In *DARPA Image Understanding Workshop*, pages 1079–1084, 1998.

[13] Y. Xiong and K. Turkowski. Creating image based vr using a self-calibrating fisheye lens. In *IEEE Computer Vision and Pattern Recognition (CVPR97)*, pages 237–243, 1997.

Nonparametric, Model-Based Radial Lens Distortion Correction Using Tilted Camera Assumption *

Janez Perš, Stanislav Kovačič
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Slovenia
Tel: +386 1 4768 876 Fax: +386 1 4768 279
e-mail: {janez.pers},{stanislav.kovacic}@fe.uni-lj.si

Abstract

Radial lens distortion prohibits use of simple pinhole camera models in computer vision applications, especially when using wide-angle lenses, which result in barrel-type distortion. Usual approach to radial distortion is by the means of polynomial approximation, which introduces distortion-specific parameters into the camera model and requires iterative methods for their calculation. Based on the properties of distorted images, an alternative approach is proposed in this paper. The basic assumption is that distortion occurs due to transformation of the observed differential of radius and is locally dependent of the angle of principal rays. The geometric relations which result from this assumption are complemented with the equations of the perspective radial lens projection function to derive model of radial distortion with single parameter - focal length. Experiments were conducted to illustrate the validity and performance of this approach.
Key words: lens distortions, radial distortion, camera calibration

1 Introduction

Lens *distortions* are long-known phenomena that prohibit use of simple pinhole camera models in the most of the computer vision applications. Being the most stubborn type of lens *aberrations*, they do not influence quality of the image, but have significant impact on image geometry [4]. Several types of lens distortions exist, however, *radial* distortion is usually the most severe part of the total lens distortion, especially when inexpensive

*This work was supported by Ministry of Education, Science and Sport of Republic of Slovenia (Research program 1538-517). Significant amount of work was performed during the first author's stay at Centre for Machine Perception at Czech Technical University in Prague, Czech Republic, where he was supported by the Slovenian-Czech project "Omnidirectional vision".

wide-angle lenses are used. Effect of radial distortion on image geometry is illustrated in Figure 1.

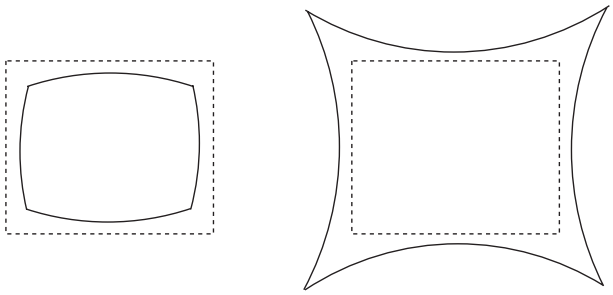


Figure 1: Effect of radial distortion on image geometry. Dashed line represents the rectangular object as it would appear in absence of radial distortion. Solid line shows the object shape in the presence of (a) barrel and (b) pinchusion distortion.

There are two major types of radial distortion [4]. When image points get displaced from its desired location to the position closer to the optical axis (negative displacement), *barrel* distortion occurs. Alternatively, image points can get displaced to the position further away from the optical axis (positive displacement), in this case *pinchusion* distortion occurs. Barrel distortion is common in wide angle lenses and it therefore dominates the distortion-related research, as far as computer vision is considered.

1.1 Related work

The science of precise measurement using optical instruments has developed long time before first computer vision-based measuring systems became available [4]. Major part of photogrammetric work was performed manually and high-quality optical equipment was prerequisite for accurate measurements. These instruments are today referred to as "metric" equipment, in contrast to "non-metric" or consumer equipment, which now dominates the field of computer vision. Expensive metric cameras usually incorporated complex optics, which included correcting elements aimed at correcting lens distortions. Radial distortions of these cameras were small (in the range of micrometers), but were nevertheless fully documented in camera documentation [4]. Calibration of these high-precision cameras was performed using highly specialized equipment.

Advent of computer vision brought off-the-shelf cameras and lenses into the field of visual inspection and measurement. This required different calibration procedures, which could be carried out on inexpensive, but computer-supported equipment. Radial distortions of these lenses were much higher (several percent at the image boundary, see [8]). Polynomial model for radial distortion, which originated in photogrammetry was adopted, as demonstrated by Tsai [8].

In the following years, many authors tried to compensate for radial lens distortion. Some of them used wide-angle lens for image acquisition, which resulted in radial distor-

tions evidently exceeding 20% [7]. This called for some kind of radial distortion correction even when no precise measurements were performed. Most approaches used polynomial approximation model for radial distortion, with rare exceptions [1], such as FET (Fish-Eye Transform) model by Basu and Licardie and FOV (Field-Of-View) model by Devernay and Faugeras. The FET model is based on the observation that fish-eye have a high resolution at the fovea, and a non-linearly decreasing resolution towards the periphery. FOV model is based on simple optical model of fish-eye lens and introduces field of view ω as distortion parameter. However, neither FET nor FOV model provides relations between the distortion parameters and the *physically measurable* lens parameters.

Most of the radial distortion-focused research is still based on polynomial models and their variations, for example [2].

1.2 Our approach

Several authors label radial distortion as an *error* of the lens design and manufacturing. However, it is inherent property of any lens [3] and has to be compensated for, either mathematically or optically. In this paper we derive a mathematical model of radial distortion which is based on the camera and lens projection geometry and does not introduce *any* distortion-specific parameters into the camera model.

This paper is structured as follows: first, we define ideal (linear) camera model and expand it with the polynomial-based (classical) radial distortion model. Next, we present a concept of radial projection function, which is used in lens design and mathematics to study lens properties. In the next step, we propose new approach for modeling lens distortion, which is subsequently used to derive alternative model of radial distortion, based on the most widely used, *perspective* projection function. Next, we present some results of tests on the real images, that demonstrate the effectiveness of this approach, and finally we conclude the paper with comments on properties of this alternative radial distortion model.

2 Linear camera and polynomial distortion model

Pinhole camera can be represented by the following linear model [3]:

$$\begin{bmatrix} \mathbf{u}_l \\ 1 \end{bmatrix} \simeq \mathbf{K}[\mathbf{R} \mid -\mathbf{R}\mathbf{t}] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{u}_l = [u_l, v_l]^\top$ are the coordinates of the image point, which is a projection of a scene point with coordinates $\mathbf{X} = [X, Y, Z]^\top$. $\mathbf{K}$ is the calibration matrix, containing all intrinsic camera parameters. $\mathbf{R}$ and $\mathbf{t}$ are rotation matrix and translation vector, the extrinsic camera parameters. This is an idealized camera model - $\mathbf{u}_l$ cannot be directly measured due to distortions.

Therefore, we can extend the linear camera model with radial distortion, which could be generally represented as [3]:

$$\mathbf{u} = d(\mathbf{u}_l, \mathbf{p}), \tag{2}$$

where $\mathbf{u}_l$ are the coordinates of the undistorted image point and $\mathbf{u} = [u, v]^\top$ are the coordinates of the distorted image point. d is the distortion function and $\mathbf{p}$ is the vector of distortion parameters. $\mathbf{u}$ is directly measurable, for example as distances on the surface of the photographic film, or as coordinates of a pixel in an image from the CCD camera.

Polynomial approximation has been thus far the preferred method of modeling radial distortion function d. Polynomial model of radial distortion can be then expressed by the following equations:

$$\frac{r}{r_l} = \frac{\|\mathbf{u} - \mathbf{u_0}\|}{\|\mathbf{u_l} - \mathbf{u_0}\|} = \frac{u - u_0}{u_l - u_0} = \frac{v - v_0}{v_l - v_0} = D(r_l, \mathbf{k}), \tag{3}$$

where the polynomial on the right-hand side is given by

$$D(r_l, \mathbf{k}) = 1 + k_1 r_l^2 + k_2 r_l^4 + \ldots + k_n r_l^{2n}. \tag{4}$$

The point $\mathbf{u}_0 = [u_0, v_0]^\top$ is the image center, $\mathbf{k} = [k_1, k_2 \ldots k_n]^\top$ are the distortion coefficients and $r_l = \|\mathbf{u}_l - \mathbf{u}_0\| = [(u_l - u_0)^2 + (v_l - v_0)^2]^{(1/2)}$ is the radius of $\mathbf{u}_l$, or the distance from $\mathbf{u}_l$ to $\mathbf{u}_0$. The length of the vector $\mathbf{k}$ is denoted by n, and $2n$ is the order of the distortion polynomial D. Most of the authors, including [8], concluded that for most of the practical tasks, second order ($n = 1$) or fourth order ($n = 2$) polynomial is sufficient. However, this may not be case for wide-angle lenses, as illustrated in [6]. In addition, independently of the number of the coefficients used, this model of radial distortion *always* requires iterative approach to obtain the coefficients, which is its major drawback. Furthermore, if the inverse of function $d(\mathbf{u}_l, \mathbf{p})$ is needed it has to be computed iteratively as well [3].

3 Camera model-based radial distortion function

Polynomial approximation of radial distortion function, as defined in Equations (3) and (4) is based on the assumption that the underlying distortion function is not known and that it cannot be obtained by analytical means. This is probably true for high-quality lens with distortion-correcting elements, where polynomial function approximates the inaccuracies in the lens manufacturing. However, this may not be true for simple, widely-used lenses which have significant distortions that result from the lens geometry itself.

3.1 Camera models

The projection geometry of most cameras can be modeled as perspective projection of the 3D world onto a sphere (the *viewing sphere*), followed by a projection of the sphere onto a plane [5]). Five ideal *radial projection functions* are used in lens design and mathematics to map an angular distance α from the optical axis onto a distance $r(\alpha)$ from the image center $\mathbf{u}_0$, [5]: perspective, $r(\alpha) = k \tan \alpha$, stereographic, $r(\alpha) = k \tan(\alpha/2)$, equidistant,

$r(\alpha) = k\alpha$, equi-solid angle, $r(\alpha) = k\sin(\alpha/2)$ and sine law, $r(\alpha) = k\sin(\alpha/2)$. The coefficient k corresponds to the focal length f of the lens used for image acquisition, [4].

3.2 Correcting the distortion

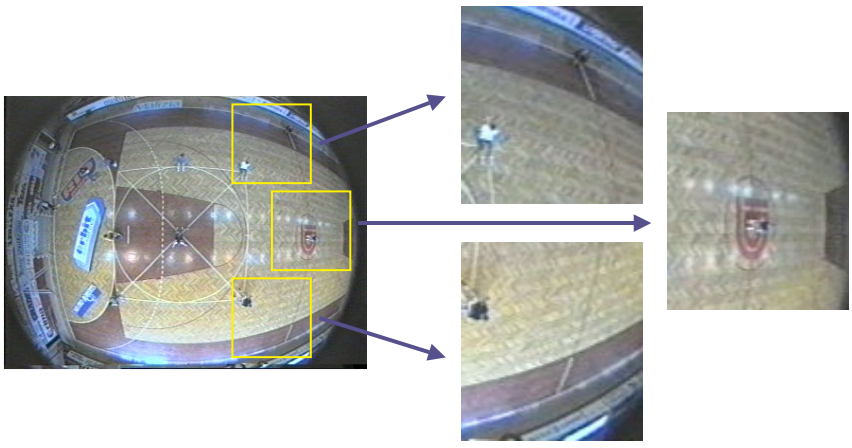
We can formulate our problem as follows: we are looking for the radial distortion function d, as defined in Equation (2). To stress the radially symmetric nature of d, we can rewrite it as

$$r' = d(r'_l, \mathbf{p}), \tag{5}$$

where $r'_l = \|\mathbf{u}'_l - \mathbf{u}'_0\| = [(u'_l - u'_0)^2 + (v'_l - v'_0)^2]^{(1/2)}$. The prime signs denote the variables which are defined on the image plane, not in the object space.

Although polynomial approximation is sometimes thought of as being the only way to model the unknown distortion function, this is not the case. Model of ideal camera can be changed to incorporate the radial distortion, even if such model does not correspond closely to the actual physics of the real camera. Example of such approach is the principle of *variable focal length* [4], which assumes that focal length of the camera changes with respect to the the radial distance r_l, which causes radial distortion. This principle was successfully employed in certain types of photogrammetric instruments [4].

Similarly, we propose another model of radial lens distortion, which can be constructed by observing the effects of radial distortion on images, acquired using wide-angle lens. Let us look at the typical, barrel-distorted image, shown in Figure 2a. Significant distortion manifests itself through intense bending of otherwise straight lines of the planar pattern.



(a) (b)

Figure 2: (a) Image of the planar pattern (handball/basketball court), acquired using wide-angle lens. (b) Three enlarged sections of the original image look similar as they were acquired with the tilted camera, using lens with the smaller viewing angle.

Closer look at the three enlarged sections of the original image, shown in Figure 2b reveals similar appearance, as if these images were acquired separately, with the help of

tilted camera, using lens with the smaller viewing angle. If tilted camera would be used, distances on these images would appear shorter than they are on the observed plane due to tilt. Then, the following assumption can be formulated:

Assumption. *Barrel distortion of wide-angle lens occurs due to the transformation of radius on the observed plane to the radius on the image plane under the influence of the viewing angle* α*.*

Certain geometric relations can be established on the basis of this assumption, as illustrated in Figure 3.

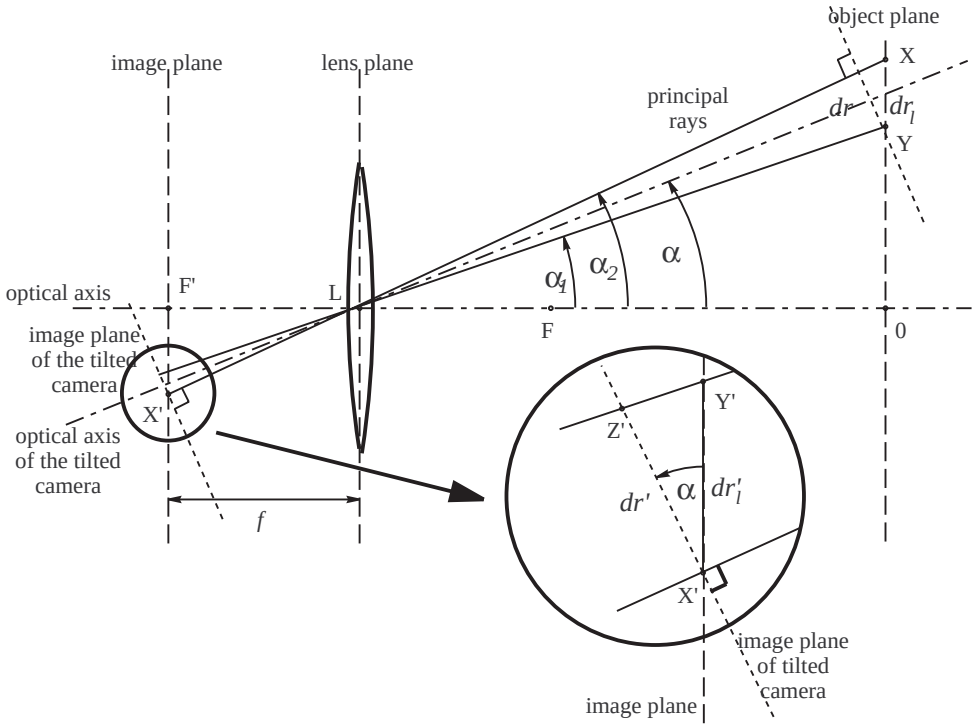


Figure 3: Geometry related to tilted camera assumption.

Position of the observed point on the image plane is marked by X, and its image on the image plane is marked X'. We can then define the following quantities: $\overline{F'L} = \overline{FL} = f$, the focal length of the lens. $\overline{OX} = r_l$ is the true radial distance of the point X in object plane. $\overline{XY} = dr_l$ is the length of differential of the radius in the object plane. Similarly, $\overline{X'Y'} = dr'_l$, is the length of the image of the radius differential dr_l in the image plane. α_1 and α_2 are angles of principal rays originating at the opposite ends of dr_l and ending at the opposite ends of dr'_l. Let us additionally assume the position of observed object in infinity[1], which causes the image plane to appear exactly at the focal point F'. We also assume that the observed dr_l part of radius r_l in the object plane is infinitely small. As a consequence, principal rays are parallel, therefore $\alpha_1 = \alpha_2 = \alpha$.

[1]In practice, this means that object distance is much larger than focal length f of the lens used, $\overline{LO} \gg f$.

To account for radial distortion with accordance to the tilted camera assumption, we introduce the concept of imaginary *tilted camera*, which is tilted for angle α from optical axis of our (real) camera, observing the point X on the image plane. Optical axis of tilted camera is intersecting with the object plane near the point X - it is intersecting the plane *exactly* at the point X if dr_l is infinitely small.[2] Similarly, let us also assume that the image plane of tilted camera intersects with the image plane of real camera exactly in point X'. The radial distance r of point X' on the image to the image plane center F' is equal to $\overline{F'X'}$. Due to camera tilt α, the differential dr'_l is then projected to the image plane of our tilted camera according to the following formula:

$$dr' = \cos(\alpha) \cdot dr'_l, \tag{6}$$

as shown in the enlarged part of the Figure 3. By changing the angle α, we can obtain distortion for every differential $dr'(r')$, along the radius r'. Relation between the angle α and radial distance r' along the image plane can be obtained from camera models, described in Section 3.1. For the most frequently used perspective model, we can write

$$r'_l = f \tan \alpha, \tag{7}$$

$$\alpha = \arctan \frac{r'_l}{f}. \tag{8}$$

By combining Equation (6) and Equation (8) we get:

$$dr' = \cos \left[\arctan \frac{r'_l}{f} \right] dr'_l. \tag{9}$$

Total radial distance r' from the point X' on the image plane to the image center F' can be obtained by integration of the Equation (9),

$$\int_0^{r'} dr' = \int_0^{r'_l} \cos \left[\arctan \frac{r'_l}{f} \right] dr'_l, \tag{10}$$

which yields[3]

$$r' = f \cdot ln \left(\frac{r_l}{f} + \sqrt{1 + \frac{r_l^2}{f^2}} \right). \tag{11}$$

This is essentially the distortion function $r' = d(r'_l, \mathbf{p})$, as defined in Equation (5), derived for perspective camera model with accordance to the tilted camera assumption. It is obvious that its parameter vector $\mathbf{p}$ contains only one parameter - focal length f. By solving the Equation (11) for r_l, we can obtain the inverse formula, which defines the

[2]Then, our tilted camera has zero viewing angle, and is distortion free.

[3]Symbolic integration and function inversion were performed using Matlab 5 and its Symbolic Math Toolbox.

transformation of distorted radial distance r' to the undistorted radial distance r'_l in the image plane,

$$r'_l = -\frac{f}{2}\frac{(e^{-\frac{2r'}{f}}) - 1}{e^{-\frac{r'}{f}}}. \tag{12}$$

4 Experiments

We tested the performance of the derived distortion model by using two lenses with focal lengths of 6.5 and 8.5 mm. Several images of calibration pattern were acquired with each of the lens and standard, linear 3D calibration was performed for each lens. Additionally, image of the checkerboard pattern was taken through each of the lens, resulting in grayscale image of 768×576 pixels. Positions of square corners in image pixel coordinates were obtained by convolving the image with the checkerboard operator; several (not more than six) missed points were added manually. Obtained points were grouped into the array of vertical and horizontal lines, which are shown in Figure 4. Four lines (two verical and two horizontal) were chosen for radial distortion evaluation. Two lines pass near the image center and serve as reference, since they are not heavily distorted. Two lines are located at image border and measure the actual improvement in grid linearity. For each of the four lines, marked with asterisks, residual error before and after RMS line fit was measured. Array of pixels was compensated for radial distortion using the formula (12) and measurements were repeated. Tables and graph in Figure 4 show the results.

3D calibration provided us with two focal lengths for each lens (for vertical and horizontal direction) since pixels are not square. The average of those two values in pixels was used for distortion correction as parameter f. Center of distortion was set to the center of image.

Results clearly show that derived distortion model closely resembles radial distortion of both tested lenses. Diagrams in Figure 4e and Figure 4f confirm that radial distortion for border lines decreased significantly. On the other hand, only marginal increase in distortion of center lines can be observed.

5 Conclusion

Derived distortion functions (11) and (12) have built-in implication that the lens radial projection function is close to the perspective projection. Projection function of particular lens can be closer to some other model [5], however, similar derivation could be done for any of the projection functions, provided that the corresponding integral (10) exists.

The distortion functions (11) and (12) need focal length f to model the radial distortion of particular lens. However, unlike the distortion parameters $k_1, k_2 \ldots k_n$, the focal length is closely related to the camera geometry and is as such part of the parameter set of every 3D calibration. Therefore, our distortion correction function introduces *no*

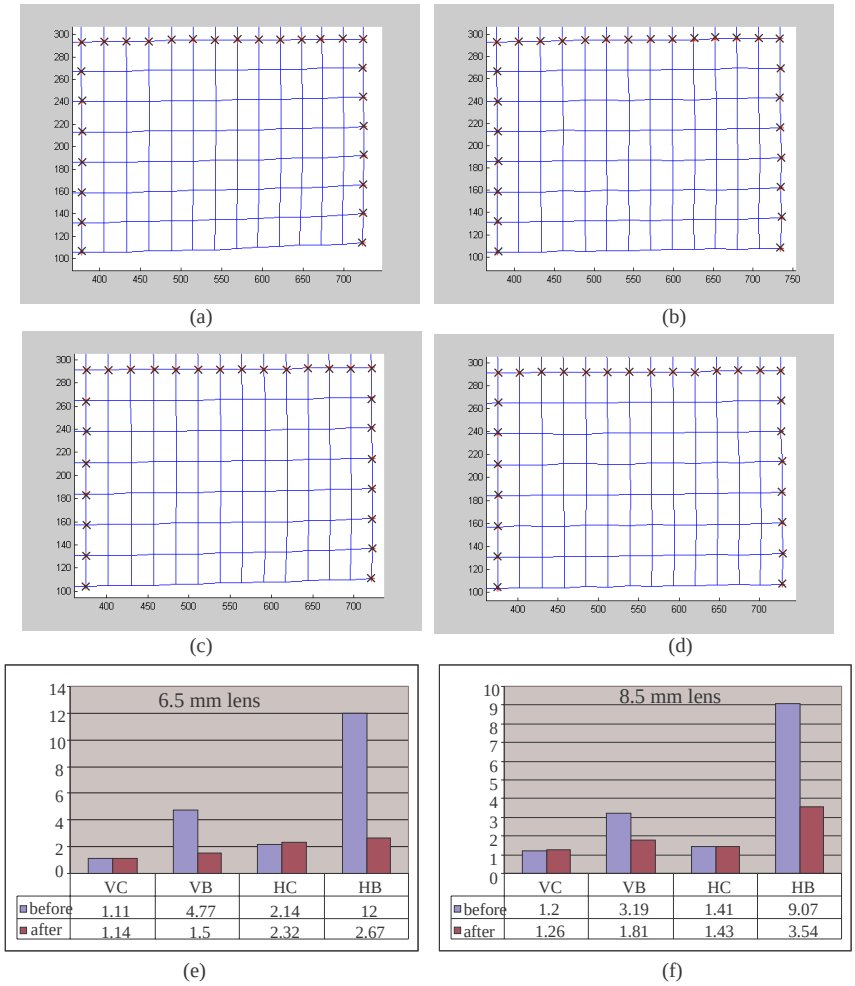


Figure 4: Experiment results. Only lower-right quadrant of the grid is shown. First row: 6.5 mm lens, $f_{average}$=818 pixels. (a) before, (b) after the correction. Second row: 8.5 mm lens, $f_{average}$=1067 pixels. (c) before, (d) after the correction. Third row: numerical results for (e) 6.5 mm lens and (f) 8.5 mm lens. VC - vertical centerline, VB - vertical borderline, HC - horizontal centerline, HB - horizontal borderline. Parts of VC, VB and HC are visible in a) through d) and marked with crosses.

distortion-specific parameters to the camera model. This has important implications. The calibration of lens which have moderate radial distortion can be simplified by not including the radial distortion into the original model. The camera parameters can be then obtained using a closed-form algorithm, for example DLT, and radial distortion can be removed afterwards, with a help of focal length, calculated during the first calibration phase. For wide-angle lenses, the linear camera model can be extended to incorporate radial distortion, which would require iterative nonlinear parameter search, however the dimension of the search space is reduced for at least two parameters of the radial distortion polynomial. Many advanced cameras (for example digital photographic cameras) can measure focal length used for each exposition, and therefore this measurement can

be used to reduce radial distortion. In the case that radial distortion correction is desired from purely cosmetical reasons, approximate focal length in pixels can be calculated from the nominal focal length of the lens used and from the dimensions of the image sensor. We successfully employed this technique for some wide-angle images from our lab, however, due to lack of space, results are not presented here.

From the viewpoint of computer vision field, lenses have two important properties: radial projection function and focal length. Both of these properties were taken into the account in the derivation of radial distortion functions, which emphasizes the view that the radial distortion really *is* an inherent property of any lens, *not* an error in manufacturing process or lens design. It is most likely that for some applications the derived radial distortion functions do not provide sufficient accuracy; in this case the need for additional polynomial model remains. However, such polynomial model would probably model the true *errors* of the lens, not the camera and lens *geometry*.

Acknowledgment

We wish to thank **Hynek Bakstein** from Centre for Machine Perception, Czech Technical University in Prague for performing the camera calibration for the purpose of here mentioned experiments. He and the rest of the CMP staff are also credited for supplying us with numerous helpful hints and suggestions related to this work.

References

[1] Devernay, F., Faugeras, O., "Straight lines have to be straight", *MVA(13)*, No. 1, 2001, pp. 14-24.

[2] Fitzgibbon, A.W., "Simultaneous linear estimation of multiple view geometry and lens distortion", *CVPR2001*.

[3] Pajdla, T., Werner, T. and Hlaváč, V., "Correcting Radial Lens Distortion without Knowledge of 3-D Structure *Technical report TR97-138*, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic.

[4] Slama C. (Ed.), *Manual of Photogrammetry, Fourth Edition*, American Society of Photogrammetry, 1980.

[5] Stevenson D. E. and Fleck M.M., "Nonparametric correction of distortion", *Technical report TR 95-07*, University of Iowa, Computer Science.

[6] Stevenson D. E. and Fleck M.M., "Robot Aerobics: Four Easy Steps to a More Flexible Calibration", *Intern. Conf. on Computer Vision*, pp. 34–39, 1995.

[7] Swaminathan R. and Nayar S.K., "Non-metric calibration of wide-angle lenses and polycameras", *IEEE Computer Society Conference CVPR 99*, Vol. 2, pp. 413-419, 1999.

[8] Tsai R.Y., "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses.", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, pp. 323-344, August 1987.

Rotational Invariants for Wide-baseline Stereo *

Jiří Matas, Petr Bílek, Ondřej Chum

Centre for Machine Perception

Czech Technical University, Department of Cybernetics

Karlovo namesti 13, Prague, Czech Republic

tel: +420 2 2435 7637, fax: +420 2 2435 7385

e-mail: [matas,bilek,chum]@cmp.felk.cvut.cz

Abstract

The problem of establishing correspondences between a pair of images taken from different viewpoints, i.e. the "wide-baseline stereo" problem, is studied in the paper. To handle the problem of affine distortion between two corresponding regions a method based on rotational invariants computed on normalised measurement regions is applied.

A robust similarity measure for establishing tentative correspondences is used. The robustness ensures that invariants from multiple measurement regions, some that are significantly larger (and hence discriminative) than the distinguished region, may be used to establish tentative correspondences.

1 Introduction

Given two images of a scene taken from arbitrary viewpoints, the problem of establishing reliable correspondences is fundamental in many computer vision tasks. Applications include 3D scene reconstruction, motion recovery, image mosaicing, content-based image retrieval, mobile robot navigation and many more. In the wide-baseline set-up, local image deformation cannot be realistically approximated by translation or translation with rotation, and a full affine model is required. Correspondence cannot be therefore established by comparing regions of a fixed shape, like rectangles or circles, since their shape is not preserved under the group of transformations that occur between the images.

In the literature, correspondences have been traditionally sought by matching features computed on local neighborhoods of detected interest points [18, 10, 1, 12]. To cope with different viewpoint, both the local regions and the descriptors of such regions have to be defined in affine

*This research was supported by the Czech Ministry of Education under Research Programme MSM 210000012 Transdisciplinary Biomedical Engineering Research and Grant Agency of the Czech Republic GACR 102/01/0971.

invariant way. The fully affine-invariant regions were introduced recently, exploiting local texture characteristics [1], or local configuration of multiple image edges or interest points [5, 16]. Schaffalitzky and Zisserman [7] presented a method for automatic determination of local neighborhood shape, but only for image areas where stationary texture occurs.

In this paper, we rely on the so called Maximum Stable Extremal Regions and Separated Elementary Cycles of the Edge Graph introduced in [4], which were shown to define highly repeatable local frames over a wide range of image formation conditions. Using measurements on these frames, we are able to successfully solve non-trivial instances of the problem of establishing correspondences between two images. We experimentally show that the measurements are sufficiently stable.

The main contribution of the paper is the utilization of processes for determination of fully affine-invariant descriptors of local regions. The approach is based on moment invariants. However, instead of using full affine invariants [15, 2], we first normalise local region up to rotation and then only the rotational invariants are computed.

The paper is organised as follows: The structure of the class of wide-baseline and recognition algorithms and two types of distinguished regions (originally proposed by Matas et al. [4]) are discussed in Section 2 and Section 3. Section 4 aims to main contribution of this paper, i.e. identifying measurement regions and extracting their affine invariant characterisations.

In Section 5 details of a matching algorithm (from the above-mentioned class) are given. A *robust* approach proposed by Matas et al. [4] is used for tentative correspondence computation.

Experimental results on images taken with an uncalibrated camera are presented in Section 6. Epipolar geometry is established using combination of multiple types of distinguished regions. Presented experiments are summarised and the contributions of the paper are reviewed in Section 7.

2 Correspondence from Distinguished Regions

Algorithms for wide-baseline stereo described in the literature have adopted strategies with a similar structure whose core can summarised by concept based on distinguished regions (introduced by Matas et al. [4]):

Algorithm 1: Wide-baseline Stereo from Distinguished Regions - The Framework

1. Detect **distinguished regions**.
2. Describe DRs with invariants computed on **measurement regions**.
3. Establish **tentative correspondences** of DRs.
4. **Estimate epipolar geometry** in a hypothesise-verify loop.

Distinguished Regions. To identify correspondences between two images, simply detectable and stable regions have to be present in the images. We will call such regions **distinguished regions** (DR). Matas et al. [4] have defined DR in more formal way:

> Let image I be a mapping $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$. Let $\mathcal{P} \subset 2^{\mathcal{D}}$, i.e. $\mathcal{P}$ is a subset of the power set (set of all subsets) of $\mathcal{D}$. Let $\mathcal{A} \subset \mathcal{P} \times \mathcal{P}$ be an adjacency relation on $\mathcal{P}$ and let $f : \mathcal{P} \rightarrow \mathcal{T}$ be any function defined on $\mathcal{P}$ with a totally ordered range $\mathcal{T}$. A **region** $\mathcal{Q} \in \mathcal{P}$ **is distinguished** with respect to function f iff $f(\mathcal{Q}) > f(\mathcal{Q}'), \forall(\mathcal{Q}, \mathcal{Q}') \in \mathcal{A}$.

Measurement Regions. Note that we do not require DRs to have any transformation-invariant property that is unique or rare in the image. In other words, DRs need not be discriminative (salient). If a local frame of reference is defined on a DR by a transformation-invariant construction (projective, affine, similarity invariant), a DR may be characterised by invariant measurements computed on any part of an image specified in the local (DR-centric) frame of reference. We used the term **measurement region** for this part of the image.

Invariant Descriptors. The most simple situation arises if a local affine frame is defined on the DR. Photometrically normalised pixel values from a normalised patch characterise the DR invariantly. More commonly, only a point or a point and a scale factor are known, and rotation invariants [9, 8] or affine invariants [15, 2] must be used.

Tentative Correspondences. At this stage, we have a set of DRs for each image and a potentially large number of invariant descriptors associated with each DR. Selecting mutually nearest pairs in Mahalanobis distance is the most common method [8, 15, 9]. Note that the objective of this stage is not to keep the maximum possible number of good correspondences, but rather to maximise the fraction of good correspondences. The fraction determines the speed of epipolar geometry estimation by the RANSAC procedure [13].

Epipolar Geometry estimation is carried out by a robust statistical method, most commonly RANSAC. In RANSAC, randomly selected subsets of tentative correspondences instantiate an epipolar geometry model. The number of correspondences consistent with the model defines its quality. The hypothesise–verify loop is terminated when the likelihood of finding a better model falls below a predefined threshold.

3 Detection of DR

The art is in finding distinguishing properties that can be detected without the obviously prohibitive exhaustive enumeration of all subsets. We employ new types of distinguished regions proposed by Matas et al. [4]. For both types of DR, *Separated Elementary Cycles of the Edge Graph (SECs)* and the *Maximally Stable Extremal Regions (MSERs)*, an efficient (near linear complexity) and practically fast (from fraction of a second to seconds) detection algorithm has been found. Low computational complexity and invariance to photometric and geometric transformation are desirable theoretical properties of the process of distinguished region detection. Stability, robustness and frequency of detection and hence usefulness of a particular type of DR has been tested experimentally and successful wide-baseline experiments on indoor and outdoor datasets was presented.

4 Affine Invariant Description of DR

4.1 Affine Invariant Measurement Region

If we have identified DR, we would like to characterize it by measurements computed on part of an image (measurement region, MR) defined by this DR. In order to cope with different viewing conditions the MRs and descriptors extracted from MR have to be defined in an invariant way. We will assume that only affine transformation is present.

To define measurement region we use first μ_1 and second Σ_1 statistics computed on data positions within distinguished region. The mean μ_1 and covariance matrix Σ_1 define ellipse $E_1(x, y)$:

$$(\mathbf{x} - \mu_1)^T \Sigma_1^{-1} (\mathbf{x} - \mu_1) = 1, \tag{1}$$

where $\mathbf{x} = (x, y)^T$. Without loss of generality we will further assume $\mu_1 = \mu_2 = (0, 0)$. First, we will prove that covariance matrix Σ_1 and covariance matrix Σ_2 computed on original DR and DR after affine transformation A are also related by A. Let:

$$\Sigma_1 = \frac{1}{|\Omega_1|} \int_{\Omega_1} \mathbf{x}\mathbf{x}^T \, d\Omega_1 \quad \text{and} \quad \mathbf{y} = A\mathbf{x}, \tag{2}$$

then

$$\Sigma_2 \;=\; \frac{1}{|\Omega_2|} \int_{\Omega_2} \mathbf{y}\mathbf{y}^T \, d\Omega_2 = \frac{1}{|A||\Omega_1|} \int_{\Omega_1} A\,\mathbf{x}\mathbf{x}^T A^T |A| \, d\Omega_1 = A\,\Sigma_1\,A^T, \tag{3}$$

where Ω_1 and Ω_2 are regions defined by first and second distinguished region.

Next we will prove that if the distinguished region is transformed by affine transform A, then the transformation between ellipses E_1 and ellipse E_2 (defined by Σ_2 computed from transformed region) is known:

$$\mathbf{y}^T \Sigma_2^{-1} \mathbf{y} = (A\mathbf{x})^T (A\Sigma_1 A^T)^{-1} (A\mathbf{x}) = \mathbf{x}^T A^T A^{-T} \Sigma_1^{-1} A^{-1} A\mathbf{x} = \mathbf{x}^T \Sigma_1^{-1} \mathbf{x}. \tag{4}$$

It means, both the original and the transformed ellipses are related by affine transformation A. The E_1 can be transformed to E_2 (and vice versa), nonetheless, this transformation is known up to rotation. Indeed, if we denote $\Sigma_2 = C^T C$, we can write for arbitrary rotation R:

$$\mathbf{y}^T \Sigma_2^{-1} \mathbf{y} = \mathbf{y}^T (C^T C)^{-1} \mathbf{y} = \mathbf{y}^T (C^T \underbrace{R^T R}_{E} C)^{-1} \mathbf{y}. \tag{5}$$

The ellipse defined by covariance matrix will be used for MR normalisation up to rotation in next step.

4.2 Affine Invariant Description

There exist three basic ways how to obtain characterisation of DR that will be invariant to affine transformation: 1. compute affine invariant directly from MR, 2. identify local affine coordinate system and normalise MR or 3. normalise MR up to rotation and compute rotation invariant.

The first two approaches are used in many recent image matching and wide-baseline stereo algorithms [6, 15, 14]. In this paper we are focused on third approach, i.e. we normalise all MRs up to rotation (exploiting results from previous paragraph) and on this region compute characterisation that is invariant to rotation (see Fig. 1). It is obvious that this characterisation has to be also affine invariant.

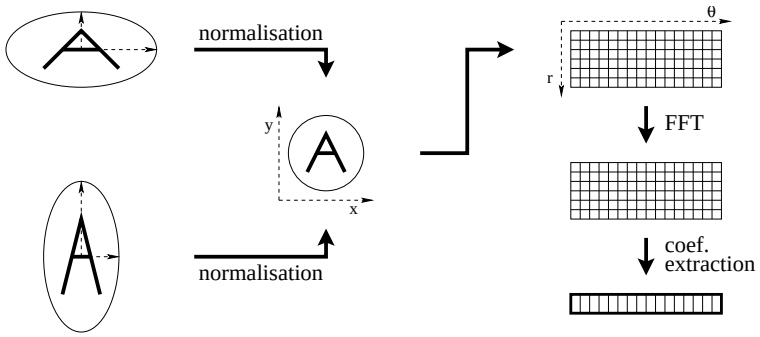


Figure 1: Computing affine invariant descriptors

We have to find a transformation which turns ellipse defined by covariance matrix to unity circle (whitening of covariance matrix). By this transformation we 'normalise' measurement region. Since corresponding MRs are same up to an affine transform, the normalised MR are same up to an unknown rotation. Inn order to match corresponding MR we have to determine this rotation.

The requirement that the rotation have to be known can be relaxed by employing rotation invariants. We exploit invariants based on integral transform (moment invariants). Let $I(x, y)$ be normalised MR, then rotational moment of order $k + l$ is defined as

$$M(k, l) = \iint P_k(r) e^{-i\theta l} I(r, \theta)\, d\theta dr, \tag{6}$$

where $I(r, \theta) = I(r \cos\theta, r \sin\theta)$ and $P_k(r)$ is polynomial with degree k. Rotational invariant descriptor is then computed as magnitude of moment with given order:

$$R_{inv}(k, l) = |M(k, l)|. \tag{7}$$

In our experiments we use $P_k(r) = r^k$, $k = 0, 1, 2$. Note, the other types of moments (Zernike, Fourier-Mellin, Complex etc.) are special cases [17, 11] for particular choice of function $P_k(r)$.

The algorithm for extracting affine invariant characterisation of DR is simply deduced from equations and can be summarised as follows (see also Fig. 1):

Algorithm 2: Extracting affine invariant characterisation of DR

1. compute first and second order statistics of MR
2. transform MR to have unit covariance matrix

3. express data of normalised MR in polar coordinates,
4. apply one-dimensional FFT along θ-axis and keep only magnitude of complex numbers,
5. combine coefficients along r-axis according to polynomial $P_k(r)$.

---

5 Matching

For establishing tentative correspondence we employ robust matching method proposed by Matas et al. [4]. Each DR is described by a measurement vector $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$. In the matching problem there are two sets $\mathcal{L}$ and $\mathcal{R}$ of DR measurement vectors originating from the 'left' and 'right" image respectively. The task is to find tentative matches given the local description. The set of initial correspondences is formed as follows. Two regions with descriptions $\mathbf{x} \in \mathcal{L}$ and $\mathbf{y} \in \mathcal{R}$ are taken as a candidates for a match iff $\mathbf{x}$ is the most similar measurement to $\mathbf{y}$ and *vice-versa*, i.e.

$$\forall \mathbf{x}' \in \mathcal{L} \setminus \mathbf{x} : d(\mathbf{x}, \mathbf{y}) < d(\mathbf{x}', \mathbf{y}) \qquad \text{and} \qquad \forall \mathbf{y}' \in \mathcal{R} \setminus \mathbf{y} : d(\mathbf{y}, \mathbf{x}) < d(\mathbf{y}', \mathbf{x}),$$

where d is the asymmetric similarity measures defined below. In the computation of $d(\mathbf{x}, \mathbf{y})$ each component of the measurement vector is treated independently. The similarity between the i-th component of $\mathbf{x}$ and $\mathbf{y}$ is measured by the number of vectors $\mathbf{y}'$ whose i-th measurement is closer. In other words the similarity in the i-th component is the rank of the measurement from $\mathbf{y}$ among all measurements $\mathbf{y}'$ from $\mathcal{R}$:

$$\text{rank}^i_{\mathbf{x},\mathbf{y}} = |\{A \in \mathcal{L} : |a_i - y_i| \leq |x_i - y_i|\}|.$$

The overall similarity measure is then defined as follows

$$d(\mathbf{x}, \mathbf{y}) = |\{i \in \{1, \ldots, n\} : \text{rank}^i_{\mathbf{x},\mathbf{y}} < t\}|,$$

where n is dimension of the measurements vector and t a predefined ranking threshold. The computation of $d(\mathbf{y}, \mathbf{x})$ is analogous with the roles of $\mathcal{L}$ and $\mathcal{R}$ interchanged. The most important property of d is that the influence of any single measurement is limited to 1. Only the main idea of the probabilistic error model behind the design may be mentioned due to limited space. Under a very broad range of error models, corresponding measurements are more likely to be below the ranking threshold than a mismatch.

6 Experiments

The following parameters of the matching algorithm were used in following experiments. As distinguished regions we have tacitly used combination of **Separated Elementary Cycles of the Edge Graph** (SECs) and **Maximally Stable Extremal Region** (MSERs). These DR was

detected by method proposed by Matas et al. [4]. The measurement regions defined in terms of affine-invariant constructions on the DR boundaries were the following: the DR itself and its convex hull scaled by factors of 1.5, 2 and 3. The MRs were described by affine invariant characterization as proposed in Section 4. Tentative correspondences comprised only those pairs whose characterisation were mutually nearest in the robust similarity measure. Epipolar geometry was estimated by the 7-point algorithm [3]. In all experiments, only a linear algorithm is used [3] to estimate epipolar geometry; no effort was made to improve the precision by known methods such as bundle adjustment, correlation, or homography growing.

6.1 Experiment I: Stability of measurement regions

The stability of assignment of measurement regions to distinguished regions is experimentally validated. In Fig. 2 there are examples of two corresponding DR with MR (fitted ellipse) in the left and right column. The MR are normalised and expressed in polar coordinates (r is vertical and θ horizontal axis). The result is depicted in the middle column. It is obvious, the normalised MR are same up to translation in r (horizontal) axis, which corresponds to rotation in Cartesian coordinates.

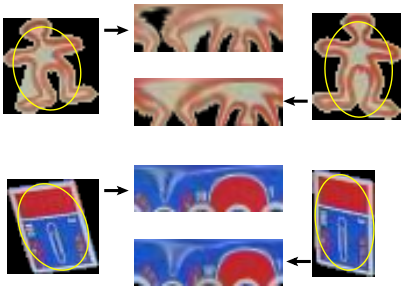


Figure 2: Stability of MR (see text for comments).

6.2 Experiment II: Epipolar geometry estimation

We tried to estimate epipolar geometry on the image from BOOKSHELF. The number of DRs in the left and right images was 1091 and 1118 respectively. The number of DRs with mutually nearest invariant descriptions, i.e. the number of tentative correspondences, was 424 in this test.

The RANSAC procedure found an epipolar geometry consistent with 187 tentative correspondences of which all are correct. The numbers of detected DRs in the left and right images, tentative correspondences (TC), epipolar geometry consistent correspondences (EG) and the number of mismatches (miss) are summarised in the caption of Figure 3. Mismatches are correspondences consistent with the estimated epipolar geometry that are not projections of the same part of the scene. The ratio TC/EG determines the average number of RANSAC hypothesis-verify attempts and hence the speed of epipolar geometry estimation.

The bottom row of Figure 3 shows close-ups of two rectangular regions selected from the left and right images respectively.

In the experiment conducted by Matas et al. [4] (the same dataset, but DR described by affine invariants) the number of tentative correspondences was 52, from which 29 was consistent. It is obvious that both methods provide comparable results, but further experiments on a wide range of scenes are needed to improve understanding of their relative merits.

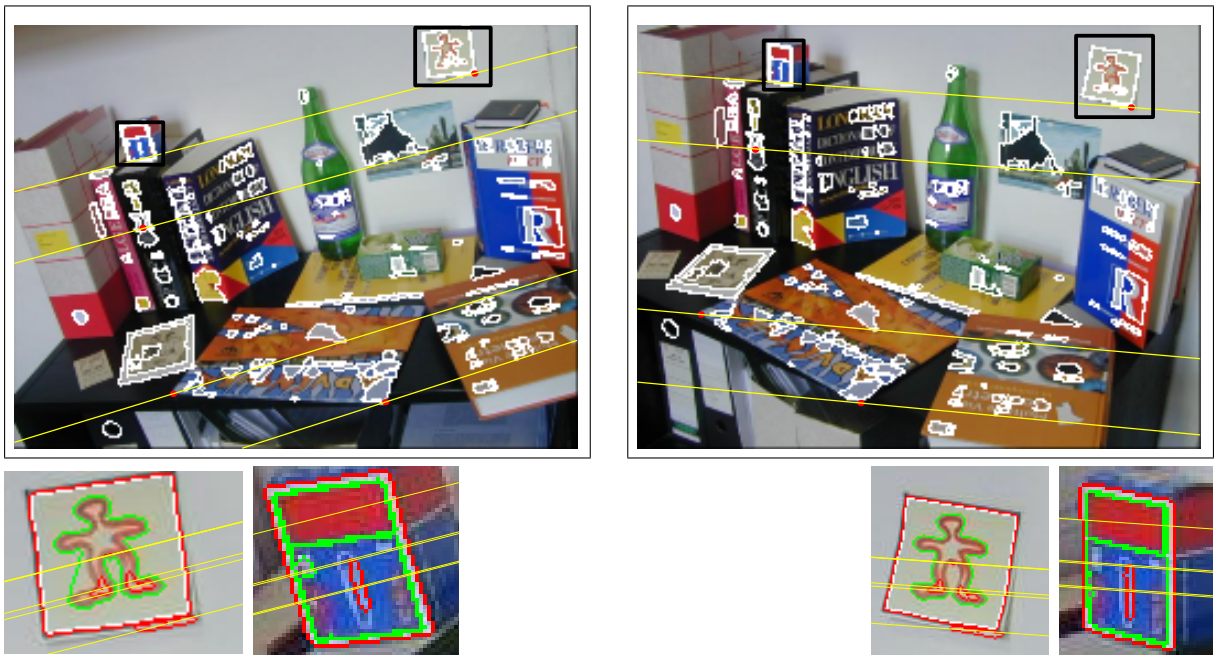


Figure 3: BOOKSHELF: Estimated epipolar geometry (top row) and a close-up of selected areas marked with black rectangles in the originals (bottom row).

	left	right	TC	EG	miss
DRs	1091	1118	424	187	0

7 Conclusions

The main contribution of the paper is the method for defining affine invariant measurement regions and manner how the invariant characterisation of MRs is computed. We establish local affine frame, that is determined up to rotation and such rotation is eliminated by employing rotational invariants. This approach is similar to computation of affine invariants, nonetheless it can provide better stability. Moreover, it can serve as 'another detector of correspondences' in existing application and improve estimation of epipolar geometry.

In a second contribution, a robust similarity measure for establishing tentative correspondences was used. Due to the robustness, we were able to consider invariants from multiple measurement regions, even some that were significantly larger (and hence probably discriminative) than the associated distinguished region.

Good estimates of epipolar geometry were obtained on wide-baseline problems with the robustified matching algorithm operating on the output produced by the proposed detectors of distinguished regions. Fully affine distortions was present in the tests. Nonetheless, further experiments on a wide range of scenes are needed to improve understanding of merits of rotational and affine invariants.

References

[1] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. of Computer Vision and Pattern Recognition*, pages I:774–781, 2000.

[2] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Journal of the Pattern Recognition*, 26(1):167–174, 1993.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[4] Jiří Matas, Ondřej Chum, Urban Martin, and Tomáš Pajdla. Distinguished regions for wide-baseline stereo. Research Report CTU–CMP–2001–33, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, November 2001.

[5] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV'98*, pages 754–760, 1998.

[6] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 754–760, January 1998.

[7] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*, July 2001.

[8] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Eighth Int. Conference on Computer Vision (Vancouver, Canada)*, 2001.

[9] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–535, May 1997.

[10] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[11] C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[12] Dennis Tell and Stefan Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *ECCV (1)*, pages 814–828, 2000.

[13] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In *BMVC96*, page Motion and Active Vision, 1996.

[14] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Proc Third Int'l Conf. on Visual Information Systems*, pages 493–500, 1999.

[15] T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In M. Mirmehdi and B. Thomas, editors, *Proc British Machine Vision Conference BMVC2000*, pages 412–422, London, UK, 2000.

[16] Tinne Tuytelaars and Luc J. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Visual Information and Information Systems*, pages 493–500, 1999.

[17] J. Wood. Invariant patter recognition: A review. *Journal of the Pattern Recognition*, 29(1):1–17, 1996.

[18] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

Estimation of the Temporomandibular Joint Position *

Vladimír Smutný, Jan Čech, Radim Šára, Taťjana Dostálová†

Czech Technical University, Center for Machine Perception

Karlovo nam. 13, 121 35 Prague 2, Czech Republic

+420 2 24357280, fax +420 2 2435 7385

e-mail: smutny@cmp.felk.cvut.cz

†Charles University, Prague

Abstract

The paper describes novel use of computer vision methods in dentistry. The temporomandibular joint head trajectory shall be estimated relative to the skull. The camera looks to two known targets, the first mounted on the upper teeth (that is on the skull), the second is mounted on the lower teeth (on the jawbone). The relative position of the targets is estimated by the camera calibration procedure using the known points on the targets. The position and trajectory of the joint head otherwise hidden for camera view by the skin is calculated from the sequence of images.

1 Medical Motivation

Dental prosthetics is widely applied in the practice and almost everybody will sooner or later be treated. The prosthetics requires to model the matching of the upper and lower teeth arc, particularly the matching of the inserted prosthesis with the opposing teeth on the other arc. This matching is performed in the special apparatus called articulator (see Fig. 1). The articulator models the relative position and the motion of lower jaw relative to the skull (that is upper jaw). The contemporary articulators have several adjustable parameters which model the differences between individual patients.

The mandibular joint is different from the most other joints in the human body. Two joints, left and right, connect the lower jaw with the skull. The joints together allow lower jaw three degrees of freedom, the jaw can be opened (abduction), put forward (propulsion), and put to the side (lateropulsion). The actual shape of the mandibular joint head and skull define the

*The 1st author was supported by the European Union under project IST-2001-33266 and by the Czech Ministry of Education under projects MSM 210000012, MSMT Kontakt ME412 and by the Grant Agency of the Czech Republic under projects GACR 102/01/0971, GACR 102/01/1371 and by the Czech Ministry of Health under project NN6333-3/2000. The 3rd author was supported by the Czech Ministry of Education under project LN00B096 and by the Czech Ministry of Health under project NN6333-3/2000.

Figure 1: The articulator is used to adjust the prosthetic implants. Upper and lower teeth arc shall slide smoothly when patient is chewing. The relative position of the plaster casts of the upper and lower teeth arc are adjusted in the articulator. The articulator could be tuned by several parameters. The scale for adjusting the angle between joint trajectory and occlusion plane is visible on the aluminium ring.

allowed motions. The first approximation of the individual head motion is rotation around the axis passing through the centers of joint heads (opening) and sliding of the heads in the skull either synchronously (propulsion) or of just one head (lateropulsion). The sliding is described in the medical literature as the ”S” shape curve which is very flat and can be approximated by the straight line.

Among the most important parameters which shall be adjusted to individuals are two angles, reflecting the orientation of the mandibular joint relative to the skull. First angle is the angle between approximation of the joint trajectory shape by the straight line and the plane defined by the upper teeth arc (occlusion plane). The second angle is between the sagittal plane and the trajectory curve.

Both angles are stable for the adult individual during the whole life. That means the parameters of the skull shape shall be measured only once before first prosthetic treatment. Then the data e.g. in the form of two angles are recorded in the patient record and later reused if needed.

2 Previous Approaches

There are two basic methods used nowadays by dentists. In the first one the complicated apparatus is bound both to the head and lower jaw. The patient is then asked to perform specific motions of the lower jaw. The motion is recorded by the pencil held in the apparatus on the paper attached to another part of the apparatus. The method has a poor accuracy, it is rather complicated for the user, and time consuming.

The other method is very similar and the main difference is that the pencil and paper record is replaced by the electromagnetic sensor and computer record of the signals. The complicated operation is the common feature of both methods.

3 Proposed Method

We propose the method which simplifies the manipulation. The basic principle is the estimation of the moving target position from its image. For estimating the camera external and internal parameters one needs only one image of the calibration target if the target is three-dimensional [8]. In our case the target takes shape of the convex or concave corner (Fig. 2). The corner is covered by the self identifying pattern [2], [6]. The basic property of the pattern is that position of the vertices of the chessboard-like pattern could be measured with high accuracy even when the pattern plane is significantly slanted.

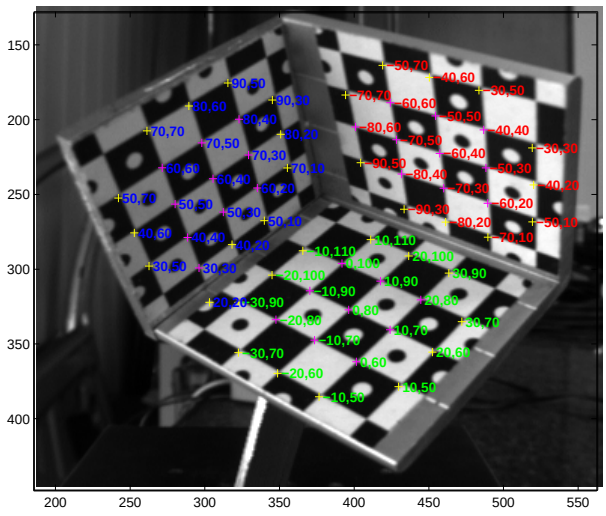


Figure 2: The target used. It is formed by three perpendicular planes, each textured with the selfidentifying pattern. In the image shown are the corners of the chessboard-like pattern found and marked by crosses. Each is accompanied by the number preassigned to it. The whole target is firmly connected to the measured object (e.g. lower jaw).

3.1 Principle

The basic principle of the method is shown in the Fig. 3. The image of the target is captured. The landmarks on the target are identified in the image, their positions in the target coordinate system are known from separate calibration. The data can be used for camera calibration obtaining the projection matrix M [8]. The projection matrix can be decomposed by QR decomposition into motion matrix A and camera projection matrix K. If two targets are present in the scene, then

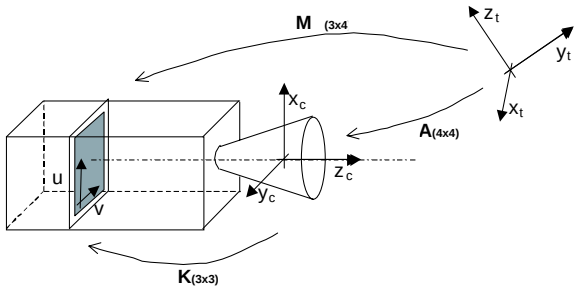


Figure 3: The transformations between the target coordinate system, camera coordinate system, and image coordinate system.

one will obtain two matrices A_1 and A_2. The relative position of the targets is then described as $A_1 A_2^{-1}$. As the patient is shaking the head during measurement, one has to see always two targets in the image, one used to eliminate the patient's motion.

The targets are mounted on the lower as well as upper jaw (Fig. 4). The procedure estimates the position of the camera relative to the upper and lower target. The relative position of the targets is calculated. Another procedure is used to establish the relative position of the joint head and the upper head. Finally the position of the joint head is known in the coordinate system defined by the occlusion plane and line passing through both heads.

3.2 Measured Parameters

In order to fulfill dentists requirements one has to measure following parameters:

1. The orientation of the occlusion plane in the coordinate system of target H.
2. The position of the axis connecting the joint heads in the coordinate system of the target D.
3. The position of the axis connecting the joint heads in the coordinate system of the target H in the jaw-back position.
4. The position of the joint heads during the motion, propulsion and lateropulsion.

3.3 Finding Occlusion Plane Orientation

The occlusion plane is found by placing the target P with the plate to the mouth, Fig. 4. The plate is pushed towards the upper teeth arc. As the plate plane is known in the coordinate system of the target P, by capturing the image of the upper target H and the target P, one can calculate the orientation of the plane P in the coordinate system of the target H:

$$\begin{bmatrix} \mathbf{n_H} \\ 0 \end{bmatrix} = (\mathbf{A_H^C})^{-1} \mathbf{A_P^C} \begin{bmatrix} \mathbf{n_P} \\ 0 \end{bmatrix}. \quad (1)$$

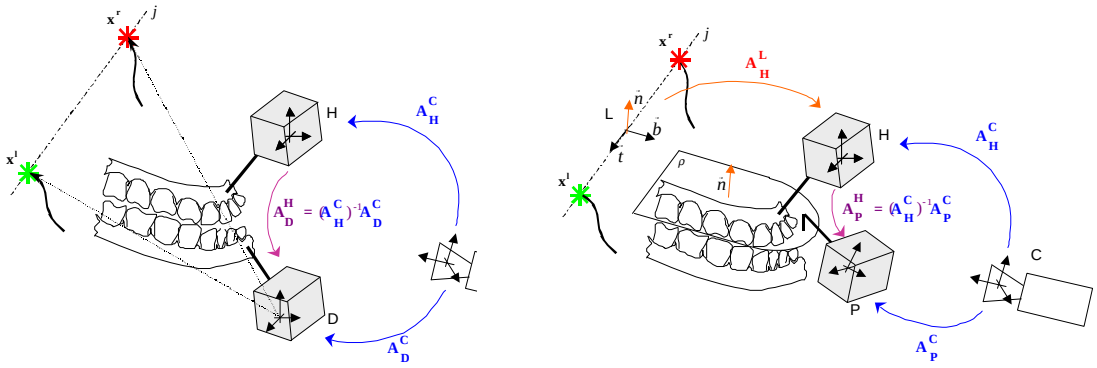


Figure 4: The connection of the targets to the coordinate systems. There are three targets used, H, P, and D. The H target is mounted to the upper teeth arc. Its coordinate system is connected to the skull coordinate system. The P target is mounted to the plate which is placed to the occlusion plane. The target D is mounted to the lower teeth arc and its coordinate system is thus related to the mandibular joint head. The joint heads are marked by stars and their "S" trajectories are shown.

Hence the occlusion plane orientation in the target H coordinate system is established.

3.4 Finding the Joint Heads

It is necessary to find the position of the left $\mathbf{x^l}$ and right $\mathbf{x^r}$ joint head in the lower target D coordinate system. Direct measurement is difficult, so we propose an automatic method which estimates first the joint axis j from mouth opening, Fig. 4.

When a patient slowly opens his or her mouth, we assume that the mandible's motion can be approximated by a pure rotation around the fixed axis j connecting the joint heads. This motion is more complicated, but the approximation is acceptable for small angles of the mouth opening.

The situation is shown in the Fig. 5. The motion matrix $\mathbf{A^H_{D_i}}$ is calculated for all images of the sequence. Motion matrices are transformed to the relative motion matrices of the target:

$$\mathbf{A_{T_i}} = (\mathbf{A^H_{D_{i-1}}})^{-1}\mathbf{A^H_{D_i}}. \tag{2}$$

Then the points of the axis are invariant points of such transformations:

$$\mathbf{A_{T_i}}\tilde{\mathbf{x}} = \tilde{\mathbf{x}}. \tag{3}$$

Let us rearrange previous equation:

$$(\mathbf{A_{T_i}} - \mathbf{E})\tilde{\mathbf{x}} = \mathbf{0}, \tag{4}$$

$$(\mathbf{R_i} - \mathbf{E})\mathbf{x} = -\mathbf{t_i}, \tag{5}$$

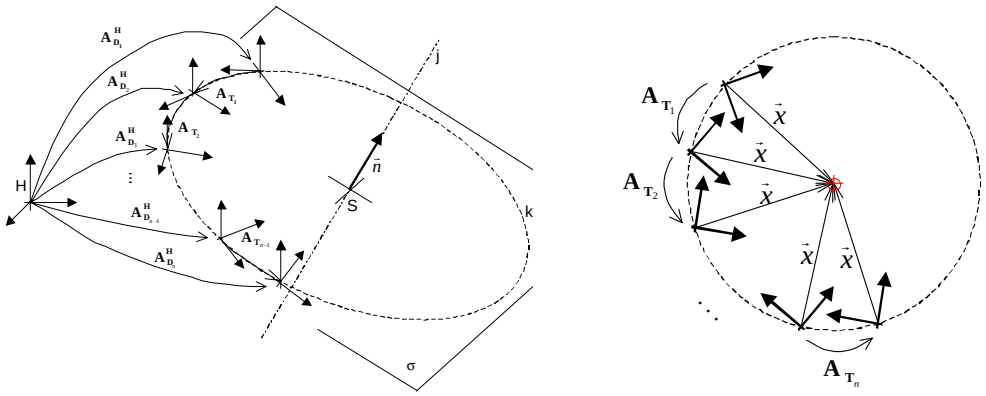


Figure 5: The motion of coordinate system connected to the lower jaw during opening of patient mouth. The motion can be approximated by the rotation around the line connecting the joint heads. The axis of rotation can be estimated from the motion and thus the relationship of the targets to the joint heads can be found.

where

$$\mathbf{A_{T_i}} = \begin{bmatrix} \mathbf{R_i} & \mathbf{t_i} \\ 000 & 1 \end{bmatrix}. \tag{6}$$

The equations (5) representing all $\mathbf{A_{T_i}}$ transformations are stacked:

$$\begin{bmatrix} \mathbf{R}_1 - \mathbf{E} \\ \mathbf{R}_2 - \mathbf{E} \\ \vdots \\ \mathbf{R_n} - \mathbf{E} \end{bmatrix} \mathbf{x} = - \begin{bmatrix} \mathbf{t_1} \\ \mathbf{t_2} \\ \vdots \\ \mathbf{t_3} \end{bmatrix}. \tag{7}$$

This equation is solved using SVD. The solution agrees with the equation of a line in space:

$$j: \ \mathbf{x} = \mathbf{x_0} + t.\mathbf{u}. \tag{8}$$

The coordinates of joint heads are set:

$$\begin{aligned} \mathbf{x}^l &= \mathbf{x_0} + 0.5d\,\mathbf{u}, \\ \mathbf{x}^r &= \mathbf{x_0} - 0.5d\,\mathbf{u}, \end{aligned} \tag{9}$$

where d is a distance between joint heads and it could be measured easily.

3.5 Joint Heads Trajectory

The patient is asked to make propulsion and lateropulsion. The sequence of images is captured and processed. As the relative position of the joint heads to the targets was established in the previous steps, it is easy to calculate the trajectory:

$$\tilde{\mathbf{x}}_{\mathbf{H}} = (\mathbf{A}_{\mathbf{H}}^{\mathbf{C}})^{-1} \mathbf{A}_{\mathbf{D}}^{\mathbf{C}} \, \tilde{\mathbf{x}}_{\mathbf{D}}, \tag{10}$$

where $\tilde{\mathbf{x}}_{\mathbf{D}}$ represents the constant coordinates $\mathbf{x}^{\mathbf{l}}$, $\mathbf{x}^{\mathbf{r}}$ of the joint heads, $\tilde{\mathbf{x}}_{\mathbf{H}}$ is their trajectory in the upper target H coordinate system.

Finally the trajectory is expressed in the standard L coordinate system, Fig. 4:

$$\tilde{\mathbf{x}}_{\mathbf{L}} = \mathbf{A}_{\mathbf{H}}^{\mathbf{L}} \, \tilde{\mathbf{x}}_{\mathbf{H}}. \tag{11}$$

The base vectors of L coordinate system are: the normal of the occlusion plane $\mathbf{n}$, orthonormal projection of the joint axis direction vector $\mathbf{t}$ and $\mathbf{b} = \mathbf{n} \times \mathbf{t}$.

The trajectories are later approximated by a straight line and required parameters are measured.

4 Experimental Results and Discussion

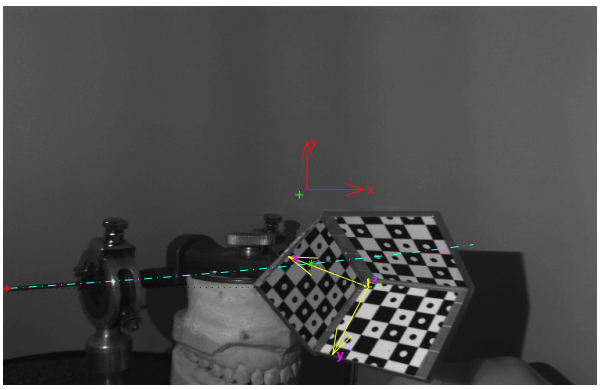


Figure 6: The target mounted on the articulator. The dot and dash line is the estimated axis of rotation during mouth opening, it coincides well with the actual axis of rotation of the upper part of the articulator. The coordinate systems shown are image coordinate system and coordinate system connected to the target.

The experiments till now were performed with the targets mounted on the computer controlled turn and translation table and with the targets mounted on the articulator (Fig. 6). The measured results show that the position of the joint head is known with the accuracy in the range of 0.1 mm (Fig. 7). These results give us motivation for further experiments in vivo. At the time we are preparing experiments with volunteers.

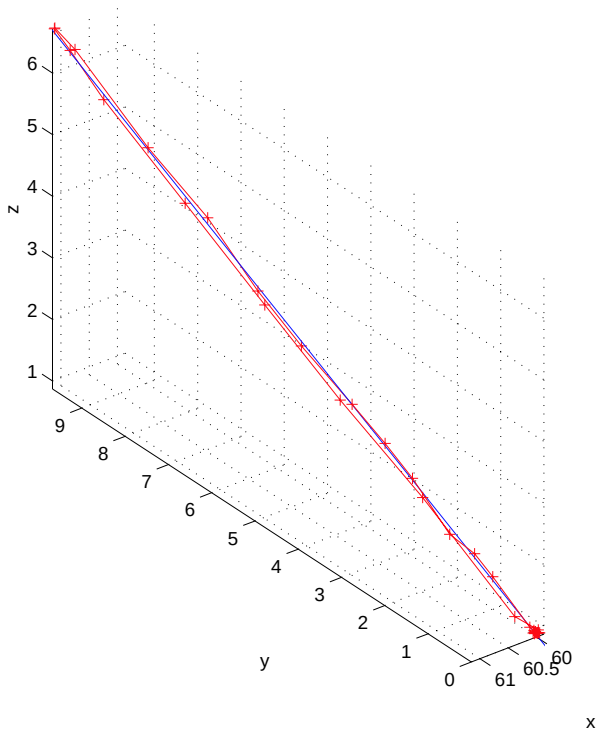


Figure 7: Trajectory of the joint heads in the skull coordinate system. The trajectory is here straight line as the experiment was performed with the articulator in the Fig. 1. The actual shape for patient shall be “S” shaped.

References

[1] Haruhiko Asada and Jean-Jacques E. Slotine. *Robot Analysis and Control*. John Wiley and Son, New York, USA, 1986.

[2] Pauseph-John Farrugia and Radim Šára. Detection of lens calibration targets in noisy and distorted images. Research Report CTU-CMP-1999-8, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, August 1999.

[3] Olivier Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Massachusetts, 1993.

[4] Olivier Faugeras, Quang Tuan Luong, and Théo Papadopoulo. *The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, Cambridge, Massachusetts, 2001.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[6] Csaba Meszaros. Authomatic detection of the calibration pattern of the perspective camera. Technical report, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, 2000.

[7] Tomáš Pajdla, Tomáš Werner, and Václav Hlaváč. Correcting radial lens distortion without knowledge of 3-D structure. Technical Report K335-CMP-1997-138, FEE CTU, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic, June 1997.

[8] Milan Šonka, Václav Hlaváč, and Roger D. Boyle. *Image Processing, Analysis and Machine Vision*. PWS, Boston, USA, second edition, 1998.

Evaluating error of homography *

Ondřej Chum, Tomáš Pajdla
Center for Machine Perception, Faculty of Electrical Engineering
Czech Technical University, Technicka 2, Prague, Czech Republic
tel: +420 2 2435 7637, fax: +420 2 2435 7385
e-mail: [chum,pajdla]@cmp.felk.cvut.cz

Abstract

In this paper, an exact computation of the geometric error for homography is derived. We assume the Gaussian noise model for the perturbation of image coordinates and formulate the problem as a least squares minimization. This paper shows how to compute accurate geometric error through solving a polynomial of degree eight. This approach avoids falling into local minima that may occur when iterative methods are used. An application where this method can improve accuracy is discussed. Experiments comparing the geometric error with its approximation by the Sampson error are presented.

1 Introduction

Estimation of a homography is used in many applications, e.g. in mosaicing [5] or wide baseline stereo matching [6]. In many applications we also need to compute the error (or the distance) of the correspondence with respect to a given homography $\mathtt{H}$. This is necessary for instance in RANSAC [2], a commonly used robust algorithm. Some applications may require not only the distance of correspondence to the model of homography but also points, which are consistent with given homography and are in a small neighborhood of measured noisy points.

We assume that a homography $\mathtt{H}$ and a noisy correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ measured in the images are given. Let the homogeneous coordinates of the corresponding points be $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$.

There are several possible ways how to measure this type of error. We will mention algebraic, geometric, and Sampson's errors. The *algebraic error* is defined as the distance between $\mathtt{H}\mathbf{x}$ and $\mathbf{x}'$ measured in the second image. However, it is well known and demonstrated in experiment 1 that this is not a good error function. Supposing the Gaussian noise model of perturbation of image coordinates, the maximal likelihood estimation of the position of noise free correspondence $\hat{\mathbf{x}} \leftrightarrow \mathtt{H}\hat{\mathbf{x}}$ is obtained by minimizing *geometric error* $d_\perp^2 = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \mathtt{H}\hat{\mathbf{x}})^2$ over all $\hat{\mathbf{x}}$. This error could be thought of as a distance of point $\mathbf{X} = (x, y, x', y') \in R^4$ to two–dimensional variety $\mathcal{V}_\mathtt{H}$ defined as points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ satisfying $\hat{\mathbf{x}}' \times (\mathtt{H}\hat{\mathbf{x}}) = \mathbf{0}$, Figure 2. The first order approximation of this error called the *Sampson's error* was first used by Sampson in [7] for conics. The derivation of Sampson's error for homographies is described in [4].

*This research was supported by the grants MSM 210000012, MSMT Kontakt 2001/09, MSMT Kontakt ME412, GACR 102/01/0971, GACR 102/00/1679

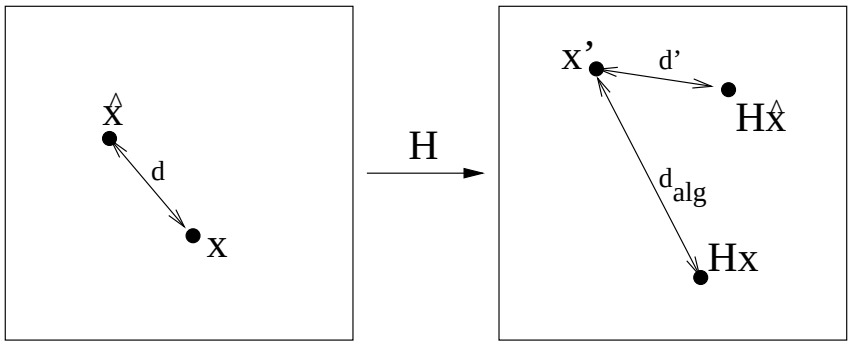


Figure 1: Two images linked with homography H. Points $\mathbf{x}$ and $\mathbf{x}$' are measured points, $\hat{\mathbf{x}}$ is point minimizing $d^2 + d'^2$ where d and d' are the distances $\mathbf{x}$ to $\hat{\mathbf{x}}$ and $\mathbf{x}'$ to $\mathtt{H}\,\hat{\mathbf{x}}$.

Exact computation of the geometric error is equivalent to finding point $\hat{\mathbf{X}} \in R^4$ on the variety $\mathcal{V}_{\mathtt{H}}$, so that the distance to the measured point $\mathbf{X}$ is minimized. This problem has not yet been addressed. A recent monograph [4], which describes current state of the art, states in the section 3.2.6: "This point can not be estimated directly except via iteration, because of the non-linear nature of the variety $\mathcal{V}_{\mathtt{H}}$."

In this paper we show that the geometric error could be exactly obtained by solving the polynomial of degree eight. This idea has first appeared in [1].

The rest of the paper is structured as follows. In section 2 the formulae for geometric error are derived. In 2.1 more details about the implementation are disclosed. Then, some experiments are presented in the section 3. An application of the new method is mentioned in the section 4, and conclusions are given in the section 5.

2 The geometric error

In this section the problem of computing geometric error is transformed to finding roots of polynomial of degree eight.

Let the regular matrix

$$\mathtt{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

represent a homography. The distance of points lying on the variety $\mathcal{V}_{\mathtt{H}}$ to the measured correspondence point $\mathbf{X}$ could be written as a function of the matrix H, measured correspondence points in images $\mathbf{x}$, $\mathbf{x}'$, and a point $\hat{\mathbf{x}}$ in the first image. If we expand the matrix multiplication, we have

$$e(\hat{\mathbf{x}}) = (x - \hat{x})^2 + (y - \hat{y})^2 + (x' - \hat{x}')^2 + (y' - \hat{y}')^2, \tag{1}$$

where

$$\hat{x}' = \frac{h_1\hat{x} + h_2\hat{y} + h_3}{h_7\hat{x} + h_8\hat{y} + h_9} \quad \text{and} \tag{2}$$

$$\hat{y}' = \frac{h_4\hat{x} + h_5\hat{y} + h_6}{h_7\hat{x} + h_8\hat{y} + h_9}. \tag{3}$$

The proof of existence of a global minimum could be found in appendix A.

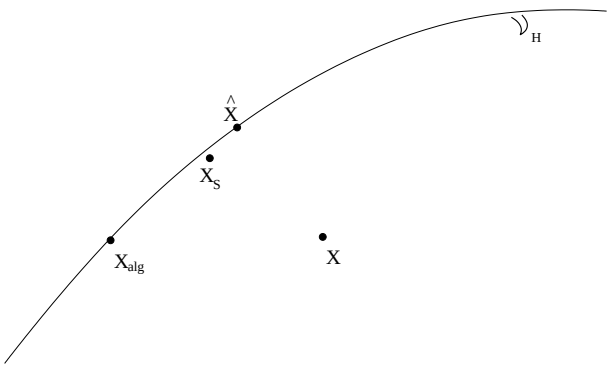


Figure 2: The variety $\mathcal{V}_{\mathtt{H}}$ and points where different errors of the measured noisy point $\mathbf{X}$ with respect to homography $\mathtt{H}$ are reached. The geometric error is reached at $\hat{\mathbf{X}}$, Sampson's error in $\mathbf{X}_S$, and the algebraic error in $\mathbf{X}_{alg}$.

Direct solving of equation $\frac{\partial e}{\partial \hat{y}} = 0$ leads to a polynomial in two variables of order four in $\hat{x}$ and order five in $\hat{y}$. The same happens for the partial derivative of e by $\hat{x}$. Therefore, we first transform the images to lower the degree of the polynomial. Since we will use only Euclidean transformations, which do not change the distances, the solution of our transformed problem will be the transformed solution of the original problem.

At first we shift the points $\mathbf{x}$ and $\mathbf{x}'$ to the origin of the first and the second image respectively. This could be achieved through applying two translations

$$\mathtt{L} = \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathtt{L}' = \begin{pmatrix} 1 & 0 & -x' \\ 0 & 1 & -y' \\ 0 & 0 & 1 \end{pmatrix}.$$

After translating the images we have

$$\mathtt{L}'\hat{\mathbf{x}}' \approx \mathtt{L}'\mathtt{H}\mathtt{L}^{-1}\mathtt{L}\hat{\mathbf{x}}. \tag{4}$$

In this equation $\approx$ stands for 'equal up to scale'. Let $\mathtt{B} = \mathtt{L}'\mathtt{H}\mathtt{L}^{-1}$ and $\bar{\mathbf{x}} = \mathtt{L}\hat{\mathbf{x}}$. We can easily verify that the first two entries in the third row remain unchanged by applying the translations, and so

$$\mathtt{B} = \begin{pmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ h_7 & h_8 & b_9 \end{pmatrix}. \tag{5}$$

Having this $\mathtt{B}$, we can rewrite the error term e as follows

$$\bar{e} = \bar{x}^2 + \bar{y}^2 + \left(\frac{b_1\bar{x} + b_2\bar{y} + b_3}{h_7\bar{x} + h_8\bar{y} + b_9}\right)^2 + \left(\frac{b_4\bar{x} + b_5\bar{y} + b_6}{h_7\bar{x} + h_8\bar{y} + b_9}\right)^2. \tag{6}$$

If $h_8 = 0$, the finding extreme of e would be easy, because $\frac{\partial \bar{e}}{\partial \bar{y}}$ would be linear in $\bar{y}$ ($\bar{e}$ would be quadratic in $\bar{y}$). We can design the image rotation $\mathtt{R}$ of the first image so that $\bar{\mathbf{x}}' = (\mathtt{B}\mathtt{R}^{-1})\mathtt{R}\bar{\mathbf{x}}$, $\mathtt{Q} = \mathtt{B}\mathtt{R}^{-1}$ and $q_8 = 0$. The angle of the rotation $\mathtt{R}$ is α for which we have equation

$$h_7 \sin(\alpha) + h_8 \cos(\alpha) = 0.$$

Solving for α we get

$$\alpha = \arctan\left(-\frac{h_8}{h_7}\right). \tag{7}$$

Now we can rewrite the term $\bar{e}$ as follows

$$\tilde{e} = \tilde{x}^2 + \tilde{y}^2 + \left(\frac{q_1\tilde{x} + q_2\tilde{y} + q_3}{q_7\tilde{x} + q_9}\right)^2 + \left(\frac{q_4\tilde{x} + q_5\tilde{y} + q_6}{q_7\tilde{x} + q_9}\right)^2, \tag{8}$$

where $\tilde{\mathbf{x}} = \mathtt{R}\,\bar{\mathbf{x}} = \mathtt{R}\,\mathtt{L}\,\mathbf{x}$.

The partial derivative $\frac{\partial\tilde{e}}{\partial\tilde{y}}$ is linear in $\tilde{y}$. The extreme is reached in $\frac{\partial\tilde{e}}{\partial\tilde{y}} = 0$, so

$$\tilde{y} = -\frac{q_2q_3 + q_5q_6 + q_1q_2\tilde{x} + q_4q_5\tilde{x}}{{q_2}^2 + {q_5}^2 + {q_9}^2 + 2q_7q_9\tilde{x} + {q_7}^2\tilde{x}^2}. \tag{9}$$

Now we can simply substitute (9) into (8) and find the extreme of $\tilde{e}$. Solving

$$\frac{\partial\tilde{e}}{\partial\tilde{x}}(\tilde{x}, \tilde{y}) = 0$$

gives a polynomial of degree eight which this paragraph is too small to contain (see next section and the appendix B).

2.1 Implementation

In the previous section we derived the formula for computing the geometric error. Here we focus on the implementation.

First of all we can see that the image rotation matrix $\mathtt{R}$ depends only on h_7 and h_8 (7). From (5) we know that $\mathtt{R}$ stays untouched by the translations $\mathtt{L}$ and $\mathtt{L}'$. So the matrix $\mathtt{R}$ could be computed directly from $\mathtt{H}$ and is the same for all the correspondences.

Coefficients of the resulting polynomial of degree eight are sums of products of entries of the matrix $\mathtt{Q}$, which are quite complicated. We can apply image rotation matrix $\mathtt{R}'$ to the second image. We have

$$\mathtt{R}'\,\tilde{\mathbf{x}}' = \mathtt{R}'\,\mathtt{Q}\,\tilde{\mathbf{x}},$$

and $\mathtt{Q}' = \mathtt{R}'\,\mathtt{Q}$. To decrease the number of summands, we can design this rotation in the same way as the matrix $\mathtt{R}$ to make $q'_4 = 0$. Note, that q'_8 stays unchanged by the rotation $\mathtt{R}'$, so $q'_8 = q_8 = 0$. Matrix $\mathtt{R}'$ differs for each correspondence.

For the resulting polynomial see appendix B.

3 Experiments

In this section we present several results of experiments in which we tested properties of different error measures.

As we mentioned in the introduction, points that satisfy $\hat{\mathbf{x}}' \approx \mathtt{H}\,\hat{\mathbf{x}}$ form a two–dimensional variety $\mathcal{V}_\mathtt{H}$ in R^4 (see fig.2). The method derived in this paper finds $\hat{\mathbf{x}}$ that minimizes $d_\perp^2 = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \mathtt{H}\hat{\mathbf{x}})^2$. This could be interpreted as finding the nearest point $\hat{\mathbf{X}}$ on the variety $\mathcal{V}_\mathtt{H}$ to the measured point $\mathbf{X}$. Another type of the error, the simplest one, is to measure the distance in the second image between the measured point $\mathbf{x}'$ and the transferred point $\mathtt{H}\mathbf{x}$. We call this *algebraic error* and denote it as d_{alg}.

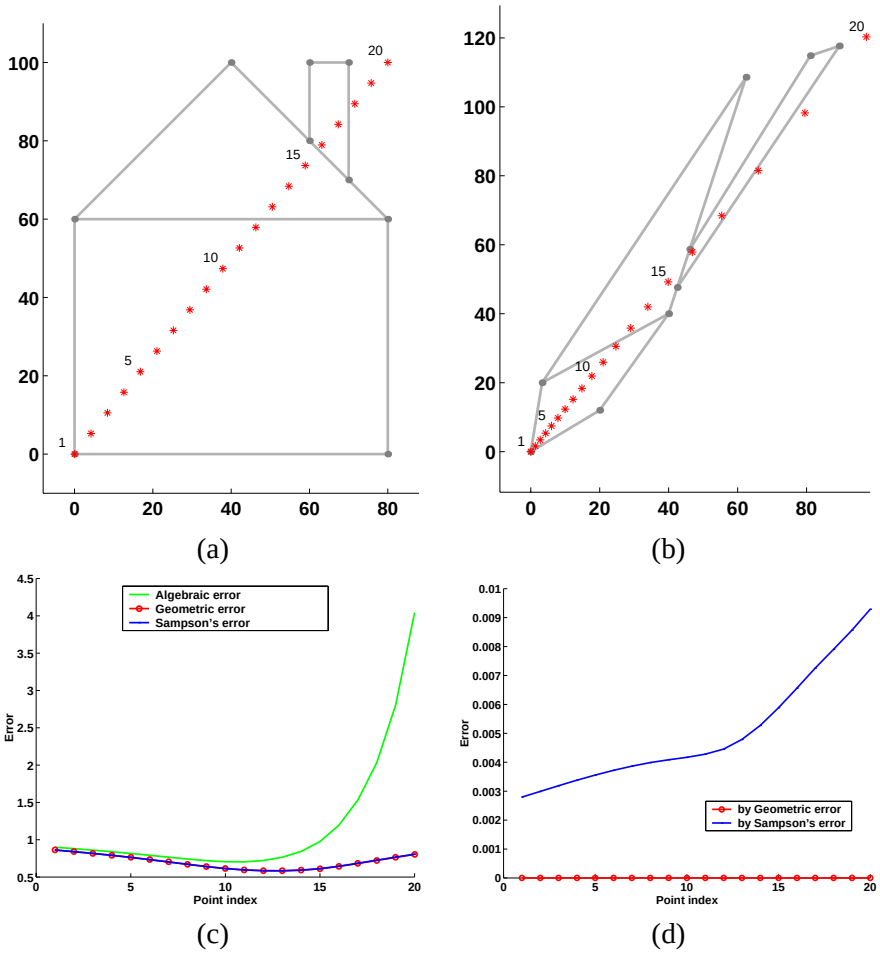


Figure 3: The two images of the house (a) and (b) linked by the homography $\mathtt{H}$, see Experiment 1. Constant noise was added to each point labeled with stars *. Graph (c) compares calculated errors (algebraic, geometric, and Sampson's) on each correspondence. Whereas points obtained by our approach satisfy $\hat{\mathbf{x}}' = \mathtt{H}\hat{\mathbf{x}}$ exactly, the full line in the graph (d) shows the distance between $\mathbf{x}'_S$ and $\mathtt{H}\,\mathbf{x}'_S$, which ideally should be zero.

This is equivalent to finding point $\mathbf{X}_{alg}$ on the variety $\mathcal{V}_{\mathtt{H}}$ with the first two coordinates equal to x and y respectively. The second two coordinates are uniquely determined by $\mathtt{H}$. The third examined method of measuring the error was the *Sampson's approximation* [4]. This method works as follows. The distance to the variety $\mathcal{V}_{\mathtt{H}}$ is approximated by the first order Taylor expansion around point (x, y, x', y'), denoted as $t_{\mathtt{H},\mathbf{x},\mathbf{x}'}$. Then points $\mathbf{x}_S$ and $\mathbf{x}'_S$ are obtained by minimizing $d_S^2 = d(\mathbf{x}, \mathbf{x}_S)^2 + d(\mathbf{x}', \mathbf{x}'_S)^2$ under condition $t_{\mathtt{H},\mathbf{x},\mathbf{x}'}(x_S, y_S, x'_S, y'_S) = 0$. It means that $\mathbf{x}_S$ and $\mathbf{x}'_S$ may not satisfy $\mathbf{x}'_S \approx \mathtt{H}\,\mathbf{x}'_S$.

Experiment 1 *It is a well known fact that the algebraic error is not a good approximation of the geometric error. In particular, the algebraic error may equal quite different values for the same geometric error depending on the position of corresponding points in images. Let us show the magnitude of such difference for one example of points in a plane and one homography.*

There are two images (fig.3) linked by a homography $\mathtt{H}$. *The points drawn as stars * are linked by the same homography too. There are twenty points denoted by *, which are numbered from left to*

right. We have added random noise to coordinates of all the points. The noise was constant over all correspondences. Then, we have measured the errors of these noisy correspondences (geometric error, Sampson's error, and algebraic error). The graph (c) in fig. 3 shows for each point the averages of these errors over one hundred repetitions. While the geometric error $d_\perp$ and the Sampson's error d_S are more or less constant for all points in the image, the algebraic error d_{alg} depends on the position of the measured image points (remember the additional noise was the same for all correspondences).

The algebraic error is so dependent not only on the noise but also on the transformation. On the other hand, we can see that the Sampson's distance is a good approximation of the geometrical error.

Experiment 2 *Now, let the points where the Sampson's error is reached be $\mathbf{x}_S$ and $\mathbf{x}'_S$ respectively. The Sampson's approximation is not accurate, i.e. the correspondence $\mathbf{x}_S \leftrightarrow \mathbf{x}'_S$ does not satisfy $\mathbf{x}'_S \approx \mathtt{H}\,\mathbf{x}_S$. It is interesting to see that even though the points $\mathbf{x}_S$, $\mathbf{x}'_S$ are not exactly related by homography $\mathtt{H}$, the distance between $\mathtt{H}\mathbf{x}_S$ and $\mathbf{x}'_S$ is very small as shows the full line in the graph (d) in the figure 3.*

We may draw the following conclusion from the above experiments. While the algebraic error is indeed not a good approximation of the geometric distance, the Sampson's distance seems to be sufficiently good for many practical situations.

4 Application

The method presented in this paper would be useful in applications where high accuracy is desired. In this section we will mention one problem where we can use the geometric error for homography to improve the accuracy of the reconstruction of planes in the scene. We call it *planar triangulation*. It is an extension of the triangulation problem [3].

The two rays in space, the first one from camera center $\mathbf{C}$ through image point $\mathbf{x}$ in the first image and the other one from $\mathbf{C}'$ through $\mathbf{x}'$, will intersect only if $\mathbf{x}'^T\mathtt{F}\mathbf{x} = 0$. If there is a noise in image coordinates, the rays may not meet.

In the triangulation problem [3], it is assumed that the fundamental matrix $\mathtt{F}$ is known exactly. For this fundamental matrix, the points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ are found, so that $\tilde{\mathbf{x}}'^T\mathtt{F}\tilde{\mathbf{x}} = 0$ and sum of the square distances $d(\mathbf{x}, \tilde{\mathbf{x}})^2 + d(\mathbf{x}', \tilde{\mathbf{x}}')^2$ is minimal.

Assume there is a (dominant) plane in the scene and $\mathtt{H}$ is the homography induced by this plane. When the triangulation method [3] is used, the additional constraint of the planarity is omitted, the reconstructed points may not lie in one plane. The homography $\mathtt{H}$ is compatible [4, sec. 12] with the fundamental matrix if, and only if for all $\hat{\mathbf{x}}$

$$(\mathtt{H}\,\hat{\mathbf{x}})^T\mathtt{F}\hat{\mathbf{x}} = 0.$$

This means all the correspondences satisfying $\hat{\mathbf{x}}' \approx \mathtt{H}\hat{\mathbf{x}}$ will automatically satisfy the epipolar geometry $\hat{\mathbf{x}}'^T\mathtt{F}\hat{\mathbf{x}} = 0$ and hence the two rays in space passing through $\mathbf{x}$ and $\mathbf{x}'$ respectively will intersect. Moreover all these intersections in space given by correspondences satisfying homography $\mathtt{H}$ lie on the plane inducing $\mathtt{H}$.

Experiment 3 *We have made synthetic experiments with the planar triangulation on images of an artificial scene (fig. 4 (a), (b)). From noise free images we obtained the fundamental matrix $\mathtt{F}$ and the homography $\mathtt{H}$. For testing purposes we used only the points on the front face of the building. Then, we added Gaussian noise with standard deviation σ to the image coordinates. From these noisy points we*

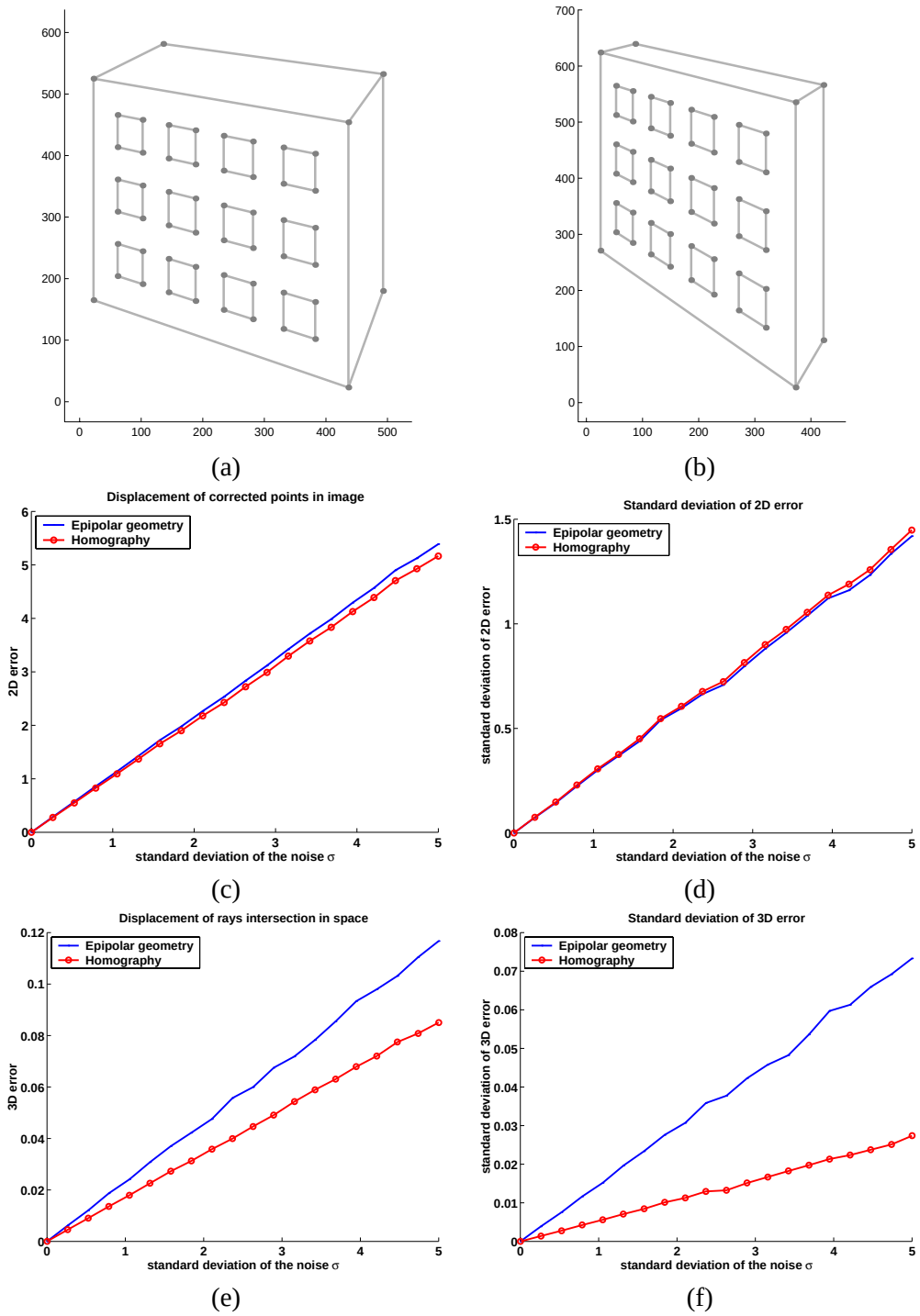


Figure 4: Synthetic experiment with images (a) and (b). Graphs compare errors in triangulation using fundamental matrix `F` (standard) and homography `H` (planar) in images (c) and (d) and in space (e) and (f). For testing were used only points lying in the plane of the frontal side of the building. The dimensions of the building are $9 \times 7 \times 1$ units.

calculated corrected points using the standard and the planar triangulation. Figure 4 gives the comparison of the distance of corrected points to the original noise free points, denoted as 2D-error (in pixels), and its standard deviation – graphs (c) and (d). As a next step we made 3D reconstruction using corrected points (both from the standard and the planar triangulation). The distance of reconstructed 3D points to the the original 3D points is denoted as 3D-error (in units, the building dimensions are $9 \times 7 \times 1$*) – graphs (e) and (f).*

The result of this experiment shows that the decrease in the error in 2D is not significant. On the other hand, the 3D error is considerably decreased by the planar triangulation.

When we tried to use Sampson's approximation followed by the standard triangulation (it consists of computing pseudo–inversion and solving the polynomial of degree six), we got similar result as when using the planar triangulation.

The experiment shows, that the accuracy of the reconstruction of a planar scene could be improved by using the planar triangulation instead of the standard one. Using the Sampson's approximation together with the standard triangulation gives very similar results as the planar triangulation but it is more computationally expensive and the planarity of the reconstructed scene is not guaranteed.

5 Conclusions

In this paper, a new method for computing the geometric error for homography was introduced. The main contribution of the paper is the derivation of the formula for computing the error. This formula has not been known before. It is interesting to see that the error is obtained as a solution of a degree eight polynomial. We have also proved that there indeed exist a corrected correspondence that minimizes the geometric distance to the measured correspondence.

We tested three different methods of measuring correspondence error with respect to given homography H. The geometric error was the best one, followed by the Sampson's error. The worst one, as expected, was the algebraic error. On the other hand, our experiments had shown that the Sampson's error is sufficiently precise for a wide range of applications including RANSAC. This paper also shows applications, where the use of the geometric error could bring higher accuracy. This statement is encouraged with experiments with the planar triangulation.

The interesting question, whether the Sampson's distance is always a good approximation of the geometric error, is left for further research.

References

[1] O. Chum. Rekonstrukce 3D scény z korespondencí v obrazech. Master's thesis, MFF UK, Prague, Czech republic, january 2001.

[2] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, June 1981.

[3] R. Hartley and P. Sturm. Triangulation. In *ARPA94*, pages II:957–966, 1994.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] Y. Kanazawa and K. Kanatani. Stabilizing image mosaicing by model selection. In *SMILE*, pages 10–17, 2000.

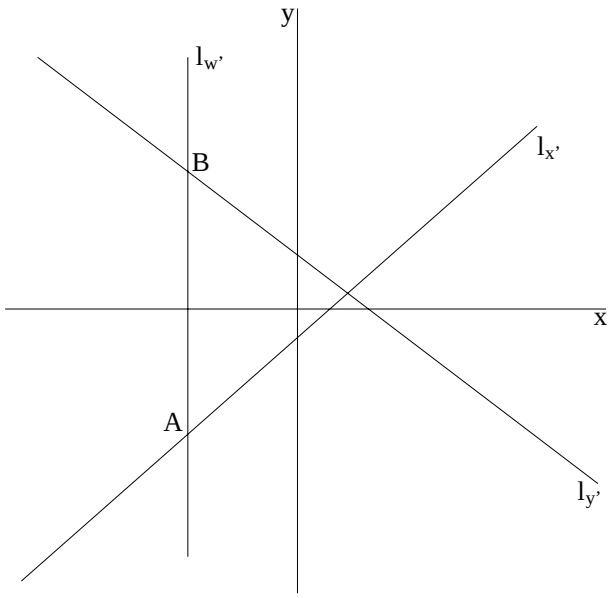


Figure 5: Lines $\ell_{\mathbf{x}'}$, $\ell_{\mathbf{y}'}$, and $\ell_{\mathbf{w}'}$ are sets of points, where $x' = 0$, $y' = 0$, and $w' = 0$ respectively.

[6] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. International Conference on Computer Vision*, pages 754–760, 1998.

[7] P. Sampson. Fitting conic sections to "very scattered" data: An iterarive refinement of the Bookstein algorithm. *CGIP*, 18:97–108, 1982.

A Proof

Theorem: Let H be a regular matrix of the following form

$$\mathtt{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & 0 & h_9 \end{pmatrix}.$$

Then the function

$$e = x^2 + y^2 + \left(\frac{x'}{w'}\right)^2 + \left(\frac{y'}{w'}\right)^2. \tag{10}$$

where

$$x' = h_1 x + h_2 y + h_3 \tag{11}$$
$$y' = h_4 x + h_5 y + h_6 \tag{12}$$
$$w' = h_7 x + h_9. \tag{13}$$

has a global minimum. In this minimum the partial derivatives of e are defined and equal to zero.

Proof: First of all we give some notation used through the whole proof. Let us write e as a sum of three functions $e_1 = x^2 + y^2$, $e_2 = (x'/w')^2$, and $e_3 = (y'/w')^2$, i.e. $e = e_1 + e_2 + e_3$. Since all e_i,

$i \in \{1, 2, 3\}$, are nonnegative, we have

$$e \geq e_i.$$

We can also define three lines, $\ell_{\mathbf{x}'}$, $\ell_{\mathbf{y}'}$, and $\ell_{\mathbf{w}'}$ in $\mathbb{R}^2$ letting x', y', and w' equal zero in (11), (12), and (13) respectively. Let $\mathbf{A}$ be the point of intersection of $\ell_{\mathbf{w}'}$ with $\ell_{\mathbf{x}'}$ and $\mathbf{B}$ be the point where $\ell_{\mathbf{w}'}$ intersects $\ell_{\mathbf{y}'}$. Since $\mathtt{H}$ is regular, there does not exist any $\mathbf{x} = (x, y, 1)^T$, so that $\mathtt{H}\mathbf{x} = \mathbf{0}$. Thus $\mathbf{A}$ and $\mathbf{B}$ are two different points. Whole situation is depicted in the figure 5.

The function e is continuous and even differentiable throughout the region where the denominator $h_7 x + h_9$ is nonzero and finite, i.e., in $\mathbb{R}^2 \setminus \ell_{\mathbf{w}'}$. The term e_1 tends to plus infinity in all points of $\ell_{\mathbf{w}'}$ except for $\mathbf{A}$ where it is guaranteed to be nonnegative. Analogously, the term e_2 tends to plus infinity in all points of $\ell_{\mathbf{w}'}$ except for $\mathbf{B}$ where it is guaranteed to be nonnegative. Their sum, as well as e, tends to plus infinity in all points of $\ell_{\mathbf{w}'}$.

We choose a point in $\mathbb{R}^2 \setminus \ell_{\mathbf{w}'}$ and take the value of e in it for a constant K. The set

$$I = \{(x, y) \in \mathbb{R}^2 \setminus \ell_{\mathbf{w}'} \mid e(x, y) \leq K\}$$

is nonempty and closed. It is also bounded because it is a subset of the circle

$$\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq K\}\ .$$

Therefore I is a compact set containing all global minima of e. At least one global minimum of e exists because the values of e on I are images of a compact set under a continuous mapping, thus they form a compact subset of $\mathbb{R}$. □

B Coefficients of the polynomial

After applying the rotations on both images, we have homography $\bar{Q}$ in the form

$$\bar{Q} = \begin{pmatrix} \bar{q}_1 & \bar{q}_2 & \bar{q}_3 \\ 0 & \bar{q}_5 & \bar{q}_6 \\ \bar{q}_7 & 0 & \bar{q}_9 \end{pmatrix}.$$

Resulting polynomial is in the form as follows $\sum_{i=0}^{8} \bar{p}_i \tilde{x}^i$. Here is the list of the coefficients p_i expressed in entries $\bar{q}$ of the matrix $\bar{Q}$. We use following substitutions

$$t = \bar{q}_3\, \bar{q}_5 - \bar{q}_2\, \bar{q}_6 \qquad \text{and} \qquad r = \bar{q}_2^2 + \bar{q}_5^2 + \bar{q}_9^2.$$

The polynomial coefficients are:

$$\begin{aligned}
\bar{p}_0 &= \bar{q}_9^3 \left(\left(-\bar{q}_3^2 - \bar{q}_6^2\right) \bar{q}_7\, \bar{q}_9 + \bar{q}_1\, \bar{q}_3\, r\right) + \bar{q}_1\, \bar{q}_5\, \bar{q}_9\, r\, t - \bar{q}_7 \left(\bar{q}_9^2 + r\right) t^2 \\
\bar{p}_1 &= -4\, \bar{q}_3^2\, \bar{q}_7^2\, \bar{q}_9^3 - 4\, \bar{q}_6^2\, \bar{q}_7^2\, \bar{q}_9^3 + 3\, \bar{q}_1\, \bar{q}_3\, \bar{q}_7\, \bar{q}_9^2\, r + \bar{q}_9\, r \left(\bar{q}_1^2 \left(\bar{q}_5^2 + \bar{q}_9^2\right) + \bar{q}_9^2\, r\right) - \bar{q}_1\, \bar{q}_5\, \bar{q}_7\, r\, t - 4\, \bar{q}_7^2\, \bar{q}_9\, t^2 \\
\bar{p}_2 &= \bar{q}_7 \left(\bar{q}_9 \left(-6 \left(\bar{q}_3^2 + \bar{q}_6^2\right) \bar{q}_7^2\, \bar{q}_9 - \bar{q}_1\, \bar{q}_3\, \bar{q}_7 \left(\bar{q}_9^2 - 3\, r\right) + 4\, \bar{q}_9^3\, r + 3\, \bar{q}_9\, r^2 + \bar{q}_1^2\, \bar{q}_9 \left(\bar{q}_5^2 + \bar{q}_9^2 + 3\, r\right)\right) - 5\, \bar{q}_1\, \bar{q}_5\, \bar{q}_7\, \bar{q}_9\, t - 2\, \bar{q}_7^2\, t^2\right) \\
\bar{p}_3 &= \bar{q}_7^2 \left(\bar{q}_9 \left(-4 \left(\bar{q}_3^2 + \bar{q}_6^2\right) \bar{q}_7^2 + 4\, \bar{q}_9^4 + 14\, \bar{q}_9^2\, r + 3\, r^2\right) + \bar{q}_1^2 \left(-\left(\bar{q}_5^2\, \bar{q}_9\right) + 3\, \bar{q}_9 \left(\bar{q}_9^2 + r\right)\right) + \bar{q}_1\, \bar{q}_7 \left(\bar{q}_3 \left(-3\, \bar{q}_9^2 + r\right) - 3\, \bar{q}_5\, t\right)\right) \\
\bar{p}_4 &= \bar{q}_7^3 \left(\left(-\bar{q}_3^2 - \bar{q}_6^2\right) \bar{q}_7^2 - 3\, \bar{q}_1\, \bar{q}_3\, \bar{q}_7\, \bar{q}_9 + 16\, \bar{q}_9^4 + 18\, \bar{q}_9^2\, r + r^2 + \bar{q}_1^2 \left(-\bar{q}_5^2 + 3\, \bar{q}_9^2 + r\right)\right) \\
\bar{p}_5 &= \bar{q}_7^4 \left(-\left(\bar{q}_1\, \bar{q}_3\, \bar{q}_7\right) + \bar{q}_1^2\, \bar{q}_9 + 25\, \bar{q}_9^3 + 10\, \bar{q}_9\, r\right) \\
\bar{p}_6 &= \bar{q}_7^5 \left(19\, \bar{q}_9^2 + 2\, r\right) \\
\bar{p}_7 &= 7\, \bar{q}_7^6\, \bar{q}_9 \\
\bar{p}_8 &= \bar{q}_7^7.
\end{aligned}$$

Experiments on High Resolution Images Towards Outdoor Scene Classification

A. Monadjemi, B. T. Thomas, M. Mirmehdi
Department of Computer Science, University of Bristol,
Bristol, BS8 1UB, England
e-mail: {monadjem,barry,majid}@cs.bris.ac.uk

Abstract

We examine the use of high frequency features in high resolution images to increase texture classification accuracy when used in combination with lower frequency features. We used Gabor features derived from sections of 4032×2688 images. A neural network classifier was used to determine the classification performance of lower and high frequency features when used separately and then in combination. Feature shuffling and Principal Component Analysis was applied to determine both the role of each feature in the classification and to extract a smaller reduced feature set involving both lower and high frequency features.

1 Introduction

In recent years, our research group has developed a neural network based system for classifying images of typical outdoor scenes to an area accuracy of approximately 90% [4, 1]. The system is trained with features extracted from segmented regions of our image database. Texture information is represented in our system using Gabor filters. A common problem confronting this approach is that many regions in typical outdoor scenes are too small to allow a significant range of spatial frequency to be included in the feature set.

This paper presents a pilot study designed to establish if high resolution images would provide a sufficient increase in texture information to justify the extra computational complexity. Our approach is to train a classifier to distinguish between four object classes using frequency information alone. Patches extracted from high resolution images are used in the training. The patch size is representative of the spatial scale of objects typical in our existing low-resolution database of outdoor scenes. They are large enough, however, at this higher resolution to allow classification performance to be assessed over a significant range of higher spatial frequencies.

High frequency information in images can be commonly associated with edges and noise [12]. We are not concerned, at least directly, with edge information. Instead, we are examining overall texture which will embody both edge and noise in 'homogeneous' regions as well as any fine or coarse resolution characteristics. Fine resolution textures, such as roads and pavements, are

particularly rich in HF information content. Some researchers disregard the higher frequencies (HF) since the power spectra of natural images show an exponential decay with frequency and in most cases the image acquisition is made through a conventional optical system that filters out the very high frequencies [10]. Nevertheless, we show in this study that higher frequency information extracted from high resolution images can improve the classification performance. It must be emphasized that in the high resolution images we have used for our experiments in this paper, what we refer to as *lower frequencies* (LF) correspond to the highest frequencies found in normal lower resolution (e.g. 512×512) images. Indeed, in some cases, these high frequencies may not even be assumed necessary, e.g. Drimbarean and Whelan [3] used relatively low frequencies by employing an octave spaced frequency set of 2,4 and 8 cycles per image size as the central frequency of a set of Gabor filters for a series textural analysis experiments. Nestares et al. [10], like many other investigators, selected $\frac{Nyquist}{2}$ as the highest central frequency of their implemented Gabor filter banks.

A number of works have considered HFs directly or indirectly, by investigating higher resolution images. Staunton [13] presented a method for measuring the HF performance of digital image acquisition systems. On the same topic, Legault [8] discussed the requirements of a high resolution imaging system for astronomical image processing. Both of these works have paid considerable attention to the Modulation Transfer Function as a criterion for high frequency response measurement. Myint [9] employed wavelet transforms as a multi-band approach to analyse textures in high resolution multi-spectral remote-sensed images. The author demonstrated the crucial role of the spatial resolution factor, which could be interpreted as the importance of higher frequency data, in dealing with fine resolution textures. Indeed, many high resolution image processing applications have concentrated on astronomy or remote sensing, where the objects sought are usually very small and imaging facilities are excellent in quality.

In this paper, we perform experiments using Gabor features obtained from both the lower and high frequency regions of the frequency space. We verify and compare the strength of LF and HF features used alone and in combination. We also determine the extra added-value obtained in using HF features as supplementary information for our example scene classification task. Within the context of this example, we examine how many and which features produce a minimised and robust set (of features) for classification. We conjecture that the usefulness of HF features is applicable across the board for high resolution texture analysis. In Section 2, we introduce our data set and choice of the feature extraction method. In Section 3 we discuss the training and testing of the neural network and briefly examine separate LF and HF classification. Experiments on combining LF and HF features will be reported in Section 4. We perform further experiments on the importance of each feature in the union of LF and HF features in Sections 5 and 6 using feature shuffling and principal component analysis. The paper is concluded in Section 7.

2 Feature Selection

Our high resolution images are 4032×2688 pixels. Considering the minimum wavelength of 2 pixels and the smaller dimension, the maximum possible frequency given a unit length will

Original Size	Patch Size	Maximum Analysable Frequency in Patch
4032×2688	256	$256/2 = 128$
1024×1024	$256 \times 1024/2688 = 97.52$	$97.52/2 = 48.76$
512×512	$256 \times 512/2688 = 48.76$	$48.76/2 = 24.38$

Table 1: The highest analysable frequency in different resolutions.

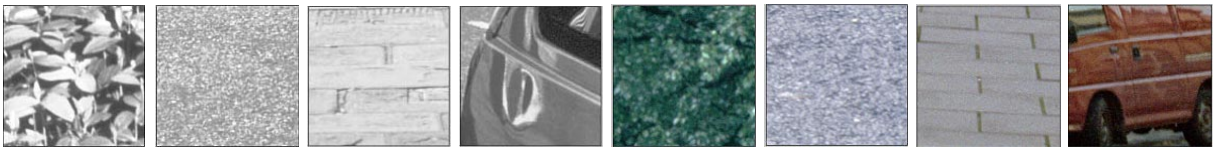


Figure 1: Eight sample patches consisting of tree, road, pavement, car classes.

be 1344 (i.e.$\frac{2688}{2}$). On the other hand, images in vision applications have diverse but mostly lower resolutions, for instance 256×256 pixels as in many cases in [11] and 712×576 pixels in [4]. This latter size is in itself quite large by average standards but still results in images which are at least 4.6 (i.e. $\frac{2688}{576}$) times smaller than our high resolution images. Consequently, the maximum frequency in a 712×576 image would be 4.6 times smaller than a 4032×2688 image. This means that any useful information in regions greater than approximately 292 (i.e. $\frac{Maximum\ Frequency}{Resolution\ Ratio} = \frac{1344}{4.6}$) in the frequency domain of our high resolution images, would be lost in the lower resolution ones of [4]. Hence, the lower frequencies we refer to in this paper correspond to the highest frequencies present in normal, say 512×512, images. We perform this same comparison for 1024×1024 and 512×512 images in Table 1.

In this work, we wish to classify our images into 4 categories: trees, pavements, cars, and roads. The input images were 256×256 patches obtained from 4032×2688 pixel high resolution outdoor scenes. A total of 146 patches were collected, divided into a training set of 114 and a test set of 32 patches. Figure 1 shows some typical examples of our input patches.

The use of Gabor filters as a tool for the analysis of textures is now a common idea, mostly due to their ability to extract information in different spatial frequency ranges and orientations. Daugman [2] originally proposed this family of 2-D filters in the 1980s as a building block for interpreting the orientation-selective and spatial-frequency-selective receptive field properties of neurons in the human visual cortex, and as useful operators for practical computational image processing. A few examples of many works that use Gabor-based features for texture classification are [5, 6, 3]. Thus, Gabor filters were also used to select the texture features in our experiments. The Gabor filter in the frequency domain is [2]:

$$g(u,v) = e^{-\pi(\frac{u_p^2}{\sigma_x^2}+\frac{v_p^2}{\sigma_y^2})}.e^{-2\pi j(x_0 u + y_0 v)} \quad (1)$$

where $u_p = (u-\omega_x)*cos(\theta)+(v-\omega_y)*sin(\theta)$, and $v_p = -(u-\omega_x)*sin(\theta)+(v-\omega_y)*cos(\theta)$, are the rotated/displaced coordinates in the frequency plan, ω_x and ω_y are filter central frequencies (modulation factors) in x and y directions, θ is filter orientation parameter, σ_x and σ_y are filter standard deviations in x and y directions, and x_0 and y_0 are horizontal and vertical

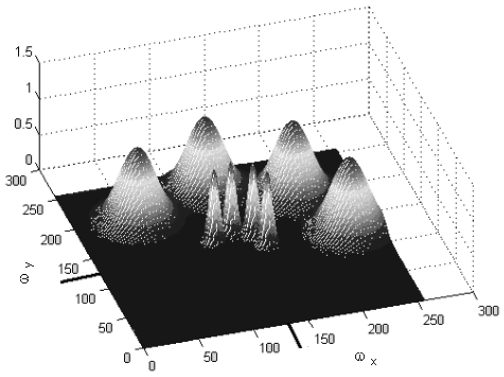


Figure 2: Applied Gabor filter bank, 4 lower frequency with $\omega = 25$ and $\sigma = 10$ (inner), and 4 higher with $\omega = 100$ and $\sigma = 40$ (outer). Orientations were $\theta = 0, \pi/4, 2\pi/4, 3\pi/4$.

displacements in the spatial domain. We keep $x_0 = 0$, $y_0 = 0$, and set $\omega_x = \omega_y$, and $\sigma_x = \sigma_y$ in all the experiments. As we intended to compare both low and high frequency features, two different Gabor filter banks with different central frequencies and bandwidths in the frequency domain were considered:

$$LF\ Filter\ \Rightarrow\ G(\omega_1, \sigma_1, \theta_1),\ \ where\ \omega_1 = 25,\ \sigma_1 = 10,\ \theta_1 = 0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4} \tag{2}$$

$$HF\ Filter\ \Rightarrow\ G(\omega_2, \sigma_2, \theta_2),\ \ where\ \omega_2 = 100,\ \sigma_2 = 40,\ \theta_2 = 0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4} \tag{3}$$

Thus, each filter bank contains four filters with the same central frequency and standard deviation but different orientations. Figure 2 shows the filter banks in the frequency domain, where the four inner peaks are the lower frequency filters ($\omega_1 = 25$) and the four larger, outer peaks are the higher frequency ones ($\omega_2 = 100$). The values for $(\omega_1, \omega_2, \sigma_1, \sigma_2)$ were chosen to reflect the coverage and density of the corresponding features in the frequency space which is also consistent with past applications of Gabor filters. More specifically, the selected LF central frequency ($\omega_1 = 25$) are close to maximum frequencies in lower resolution images such as a 512×512 image (as shown earlier in Table 1), and the HF central frequency ($\omega_2 = 100$) is considerably further away from the highest analysable frequency for patches in lower resolution images. Therefore, we would like to consider if using higher frequency features obtained from high resolution images would improve classification performance and this is what we will examine in the next section.

3 Classification using LF and HF Features Separately

For classification we employed a back-propagation neural network (BPNN) with one hidden layer. We experimented with many different numbers of hidden layer nodes, but only report

Hidden nodes	LF ($\omega = 25$) SSE	HF ($\omega = 100$) SSE
5	8.35	28.02
10	1.9	13.61
15	2.02	6.73
20	0.62	3.60

Table 2: SSE measures for Lower ($\omega = 25$) and higher ($\omega = 100$) frequency Training.

	LF ($\omega = 25$)			HF ($\omega = 100$)		
Hidden nodes	SSE	CE Error	CE %	SSE	CE Error	CE %
5	6.56	5	15.6%	15.21	16	50.0%
10	12.99	10	31.2%	18.50	14	43.7%
15	6.86	5	15.6%	26.50	18	56.2%
20	11.94	10	31.2%	20.39	15	46.9%

Table 3: Classification Errors for testing using LF and HF features separately.

our results for 5,10,15 and 20 hidden nodes, where the classifier showed the best performances overall. To measure the classification error, we used the Sum of Squared Errors, SSE, as the difference between the groundtruth G, and the network classification C across the N number of classes:

$$SSE = \sum_{i=1}^{N} (G_i - C_i)^2 \tag{4}$$

Initially, we trained the network using only LF features. Then, we trained it using only HF features. Table 2 represents these results for different numbers of hidden nodes and confirms the accuracy of LF features over HF features as expected.

After completing the training stage, the 32 unknown test images were fed to the classifiers. The results of separate LF and HF classification are shown in Table 3, where CE is the Classification Error and refers to the number of incorrect class assignments out of the total 32 inputs. It is also presented as a percentage. Again as expected, this experiment showed that the lower frequency features play the main role in texture classification in both low and high resolution images. Next, we examined if this performance can be enhanced by combining higher frequency features with low frequency ones, particularly given that they are expected to carry richer information in very high resolution images.

4 Classification using Combined LF and HF Features

We trained our BPNN on the 114 training images as before using our full complement of eight LF and HF features. The combined-features classifier had a much reduced SSE of 0.06, as shown in Table 4, compared to both separate LF and HF training SSE values of 0.62 and 3.60

	Training	Testing		
Hidden Nodes	SSE	SSE	CE Error	CE %
5	3.2	6.40	4	12.5%
10	0.63	7.23	3	9.3%
15	0.09	7.60	5	15.6%
20	0.06	8.60	8	25%

Table 4: Testing the classifier with eight LF and HF features.

LF ($\omega = 25$)						
Class	TP	FN	TN	FP	S_n	S_p
Car	8	0	24	0	100%	100%
pavement	8	0	19	5	100%	79.1%
road	6	2	24	0	75%	100%
Tree	5	3	24	0	62.5%	100%

Table 5: Sensitivity and specificity for LF features with 5 hidden nodes.

respectively. The performance of the combined classifier also improved in the testing stage correspondingly. The last three columns of Table 4 present these for both the actual SSE and CE values (out of 32 and in percentage form). The best combined-features result was 9.3% error compared to the best at 15.6% and 43.7% for LF and HF features respectively (cf. Table 3). This 6.7% minimum improvement is of an order that can be very significant in texture analysis.

On average, the combined features obtained an overall improvement of 16% in SSE and 27% in CE error compared to using just LF features for classification. We also used sensitivity S_n and specificity S_p measures to verify our results:

$$S_n = \frac{TP}{TP + FN} \qquad S_p = \frac{TN}{TN + FP} \tag{5}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative in final classification. The S_n and S_p results shown in Tables 5 to 8 further validate that the combined feature classifiers perform better than either single-band classifiers. Tables 5 and 6 present the S_n and S_p factors for the best LF feature classifier (5 hidden nodes) and HF feature classifier (10 hidden nodes) respectively. Similarly, Table 7 presents S_n and S_p results for the best combined LF and HF classifier. Table 8 demonstrates average improvement rates of 6.3% and 3.9% achieved in sensitivity and specifity respectively, when comparing the values for LF and combined LF and HF classification. Now that we have observed the improvements in combining HF features in high resolution images with the LF features, we need to determine the level of their independence and share in the combined-feature classifier's overall performance.

5 Feature Shuffling

To determine the importance of HF features in the overall classification performance, a feature *shuffling* procedure was performed. Initially, we carried out training as before using both the LF

HF ($\omega = 100$)						
Class	TP	FN	TN	FP	S_n	S_p
Car	3	5	23	1	37.5%	95.8%
pavement	6	2	19	5	75%	79.1%
road	5	3	20	4	62.5%	83.3%
Tree	4	4	20	4	50%	83.3%

Table 6: Sensitivity and specificity for HF features with 10 hidden nodes.

Combined LF and HF ($\omega = 25, 100$)						
Class	TP	FN	TN	FP	S_n	S_p
Car	8	0	24	0	100%	100%
pavement	7	1	22	2	87.5%	91.7%
road	7	1	24	0	87.5%	100%
tree	7	1	23	1	87.5%	95.8%

Table 7: Sensitivity and specificity for combined LF and HF features with 10 hidden nodes.

and HF features. However in the testing stage, a shuffling process was implemented whereby one feature out of our full complement of 8 was randomly replaced with a corresponding feature from another texture class before feeding it to the classifier. For example, we replaced the third HF feature for a car class with that of a tree class or replaced the second LF feature for a pavement class with that of a car class. The purpose of the shuffling process is to demonstrate whether each particular feature plays a significant role in the overall combined feature classification. If there is little change in the classification result then the feature we replaced is probably an insignificant one. However, if by shuffling, the classification accuracy is significantly reduced, then that feature must be important. The reason for shuffling the data instead of simple replacement with a constant, say zero, was to cause less harm to the trained network weights and especially its biases. A typical random replacement algorithm was run many times across each feature vector to perform the shuffling in our experiments.

Table 9 shows the results of shuffling either the LF feature $G(\omega = 25, \sigma = 10, \theta = 90)$ or the HF feature $G(\omega = 100, \sigma = 40, \theta = 90)$ averaged over a number of runs. Shuffling reduced the system performance in each case. Since LF features contribute significantly to overall classification, the damage inflicted through LF feature shuffling was considerably bad. For example, the performance degraded to 42.1% compared to the result of 9.3% in Table 4 for 10 hidden nodes. However, quite importantly, we can observe that shuffling the HF feature also reduced the performance to 28.1% error in a similar comparison. In Table 10 we present the average results of a series of experiments wherein we shuffled all the features one at a time.

Classifier $\rightarrow$	LF ($\omega = 25$)	HF ($\omega = 100$)	LF and HF
Avg. sensitivity	84.3%	56.2%	90.6%
Avg. specificity	92.9%	85.3%	96.8%

Table 8: Average of sensitivity and specificity.

	Shuffling LF feature $G(\omega = 25, \sigma = 10, \theta = 90^o)$			Shuffling HF feature $G(\omega = 100, \sigma = 40, \theta = 90^o)$		
Hidden nodes	SSE	CE Error	CE %	SSE	CE Error	CE %
5	19.61	12	37.5%	7.32	5.5	17.1%
10	20.47	13.5	42.1%	15.32	9	28.1%
15	21.47	14.5	45.3%	7.53	10.5	32.8%
20	16.71	14.5	45.3%	13.16	10	31.2%

Table 9: Average results when shuffling the LF feature $G(\omega = 25, \sigma = 10, \theta = 90)$ and the HF feature $G(\omega = 100, \sigma = 40, \theta = 90)$ in each case across all 32 samples

	Total Average for LF ($\omega = 25$)			Total Average for HF ($\omega = 100$)		
Hidden Nodes	SSE	CE Error	CE %	SSE	CE Error	CE %
5	16.24	11.37	35.5%	9.44	6.87	21.4%
10	18.82	13	40.6%	16.45	9.62	30.0%
15	17.38	14.75	46.0%	14.47	11.12	34.7%
20	16.14	13.25	41.4%	14.18	11.37	35.5%
Average	17.14	13.09	40.9%	13.63	9.75	30.4%

Table 10: Average of shuffling effect on all lower and higher frequency features

This table presents similar conclusions in that all the LF and HF features play a significant role in the overall classification even though the influence of the lower frequency features is greater.

6 Reducing Feature Dimensionality

Principal Component Analysis (PCA) [7] is a popular approach used in pattern recognition studies for reducing problem dimensionality by seeking and eliminating redundant features. None of our features are redundant as we showed in the previous section. However, we still apply PCA to our 8 feature set to both examine more closely the contribution of each feature and to create a reduced set comprising composite arrangements of the individual LF or HF features that can achieve a similar rate of classification.

When examining the covariance matrix of our 8 feature set we found the relative correlation of the HF features was greater than LF features suggesting a lack of variance in the HF features. Table 11 presents the eigenvectors, the eigenvalues λ and cumulative eigenvalues λ_{cum} of the full LF and HF feature set. The first three eigenvalues contained 95.5% of the total features variance i.e. 51.2%+35.1%+9.2%, and we only considered these. For the first eigenvalue, the four HF features showed the largest values which were in the range 0.466-0.472. In the second largest eigenvalue, the four LF features demonstrated the largest variance. Finally, in the third largest eigenvalue, the eigenvector corresponding to the vertical LF feature contained the largest absolute value of 0.917. These values have been highlighted in Table 11. Hence, we transformed our 8-dimensional feature space into a new 3-dimensional set using these new features: the average of the HF features, the average of the 1^{st}, 2^{nd} and 4^{th} LF features and

$LF, 0^o$	0.204	**0.489**	0.348	-0.265	0.718	0.109	0.028	-0.031
$LF, 45^o$	0.195	**0.500**	0.137	0.816	-0.159	-0.016	-0.022	-0.034
$LF, 90^o$	0.094	0.346	**-0.917**	-0.050	0.162	-0.000	0.014	0.003
$LF, 135^o$	0.180	**0.520**	0.131	-0.509	-0.642	-0.082	-0.003	0.045
$HF, 0^o$	**0.467**	-0.167	0.002	-0.011	0.11	-0.761	-0.393	0.078
$HF, 45^o$	**0.472**	-0.171	-0.009	0.038	-0.012	0.194	0.397	0.742
$HF, 90^o$	**0.467**	-0.173	-0.037	-0.027	-0.082	0.600	-0.593	-0.178
$HF, 135^o$	**0.469**	-0.183	-0.200	-0.017	-0.049	-0.044	0.579	-0.638
λ	4.099	2.809	0.737	0.188	0.110	0.044	0.009	0.006
λ (%)	51.2%	35.1%	9.2%	2.3%	1.3%	0.5%	0.1%	0.1%
λ_{cum} (%)	51.2%	86.3%	95.5%	97.8%	99.1%	99.6%	99.7%	$\approx$ 100%

Table 11: Eigenvector and eigenvalues after the PCA analysis of the 8 features.

	Training	**Testing**		
Hidden Nodes	SSE	SSE	CE Error	CE %
5	8.30	7.39	5	15.6%
10	2.33	8.19	4	12.5%
15	1.16	14.62	10	31.2%
20	0.56	11.75	10	31.2%

Table 12: Training and Testing results with 3 features.

finally the 3^{rd} LF feature, $G(\omega = 25, \sigma = 10, \theta = 90^o)$.

Following the training of a BPNN classifier using these three features, we obtained SSE values (for 5,10,15, and 20 hidden nodes) of $\{$8.30,2.33,1.16,0.56$\}$ as shown in the second column of Table 12. These compare very well with the 8-feature training SSE values outlined in the second column of Table 4. The last three columns of Table 12 show the testing classification results. The best result was obtained with 10 hidden nodes where only 4 out of the 32 patches were incorrectly classified compared to 3 as the best result of the 8-feature classification shown in Table 4. All training and testing conditions were kept the same across the experiments.

Carrying out PCA meant that we reduced our 8 feature set to one averaged HF feature and two LF features. To evaluate the importance of this new averaged HF feature, we performed new training and testing using just the two LF features. There were significant drops in accuracy for both stages. For example, in testing, the classification dropped to 7 errors for 10 hidden nodes.

7 Conclusion

In this work we have demonstrated the importance and practicality of high frequency features as an additional aid in texture analysis in high resolution images. We have shown that the performance of a single classifier can increase considerably if both lower and high frequencies are used together. It is worth reminding the reader that the lower frequencies we refer to in this paper correspond to the highest frequencies present in lower resolution images. We also

demonstrated the importance of each of the individual features through feature shuffling. PCA was then used to reduce the LF and HF features to a more compact set which at 12.5% CE error still showed a massive improvement over the separate LF or HF results in table 3. It also is very comparable to the results in Table 4 when they are combined.

Increasingly, higher and higher resolution devices such as digital cameras are becoming available. More powerful computational resources will make the handling of very high resolution images a routine matter, giving the analysis of higher frequency information an important role in computer vision.

References

[1] A.A.Clark. *Region Classification for the Interpretation of Video Sequences*. PhD thesis, Department of Computer Science, University of Bristol, 1999.

[2] J. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. on Acoustic Speech and Signal Processing*, 36(7):1169–1179, 7 1988.

[3] A. Drimbarean and P.F. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161–1167, 8 2001.

[4] M. R. Everingham, B. T. Thomas, T. Troscianko, and D. Easty. Neural-network virtual reality mobility aid for the severely visually impaired. In *Proc. 2nd European Conference on Disability, Virtual Reality and Associated Technologies*, pages 183–192, September 1998.

[5] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:102–113, 1989.

[6] A. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[7] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[8] T. Legault. What is a mtf curve? In High Resolution CCD Imaging, http://perso.club-internet.fr/legault/mtf.html.

[9] S.W. Myint. Image texture analysis with high-resolution multispectral image data using wavelet transforms. In *University Consortium for Geographic Information Science*, pages 121–130, 2000.

[10] O. Nestares, R. Navarro, J. Portilla, and A. Tabernero. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, 1998.

[11] T. Reed and J. Du Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIP Image Understanding*, 57(3):359–372, 1993.

[12] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing Analysis and Machine Vision*. International Thomson Computer Press, 1996.

[13] R.C. Staunton. Measuring the high frequency performance of digital image acquisition systems. *Electronics Letters*, 33(17):1448–1449, 8 1997.

Index of Authors