Vision Systems

Segmentation and Pattern Recognition

Vision Systems

Segmentation and Pattern Recognition

Edited by
Goro Obinata and Ashish Dutta

I-TECH Education and Publishing

Published by the I-Tech Education and Publishing, Vienna, Austria

Abstracting and non-profit use of the material is permitted with credit to the source. Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published articles. Publisher assumes no responsibility liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained inside. After this work has been published by the Advanced Robotic Systems International, authors have the right to republish it, in whole or part, in any publication of which they are an author or editor, and the make other personal use of the work.

© 2007 I-Tech Education and Publishing
www.ars-journal.com
Additional copies can be obtained from:
publication@ars-journal.com

First published June 2007
Printed in Croatia

A catalog record for this book is available from the Austrian Library.
Vision Systems: Segmentation and Pattern Recognition, Edited by Goro Obinata and Ashish Dutta

p. cm.
ISBN 978-3-902613-05-9
1. Vision Systems. 2. Pattern. 3. Segmentation. 4.Obinata & Dutta.

Preface

Research in computer vision has exponentially increased in the last two decades due to the availability of cheap cameras and fast processors. This increase has also been accompanied by a blurring of the boundaries between the different applications of vision, making it truly interdisciplinary. In this book we have attempted to put together state-of-the-art research and developments in segmentation and pattern recognition.

The first nine chapters on segmentation deal with advanced algorithms and models, and various applications of segmentation in robot path planning, human face tracking, etc. The later chapters are devoted to pattern recognition and covers diverse topics ranging from biological image analysis, remote sensing, text recognition, advanced filter design for data analysis, etc.

We would like to thank all the authors for entrusting us with their best work.

The editors would also like to express their sincere gratitude to the anonymous reviewers with out whose sincere efforts this book would not have been possible. The contributions of the editorial members of Advanced Robotic Systems Publishers, responsible for collection of manuscripts, correspondence etc., are also sincerely acknowledged.

We hope that you will enjoy reading this book.

Editors

Goro Obinata
Centre for Cooperative Research in Advanced Science and Technology
Nagoya University, Japan

Ashish Dutta
Dept. of Mechanical Science and Engineering
Nagoya University, Japan

Contents

Preface ..... **V**

1. Energy Feature Integration for Motion Segmentation **001**
Raquel Dosil, Xose R. Fdez-Vidal, Xose M. Pardo and Anton Garcia

2. Multimodal Range Image Segmentation **025**
Michal Haindl and Pavel Zid

3. Moving Cast Shadow Detection **047**
Wei Zhang, Q.M. Jonathan Wu and Xiangzhong Fang

4. Reaction-Diffusion Algorithm for Vision Systems **060**
Atsushi Nomura, Makoto Ichikawa, Rismon H. Sianipar and Hidetoshi Miike

5. A Parallel Framework for Image Segmentation Using Region Based Techniques **081**
Juan C. Pichel, David E. Singh and Francisco F. Rivera

6. A Real-Time Solution to the Image Segmentation Problem: CNN-Movels **099**
Giancarlo Iannizzotto, Pietro Lanzafame and Francesco La Rosa

7. Optimizing Mathematical Morphology for Image Segmentation and Vision-based Path Planning in Robotic Environments **117**
Francisco A. Pujol, Mar Pujol and Ramon Rizo

8. Manipulative Action Recognition for Human-Robot Interaction **131**
Zhe Li, Sven Wachsmuth, Jannik Fritsch and Gerhard Sagerer

9. Image Matching based on Curvilinear Regions **149**
J. Perez-Lorenzo, R. Vazquez-Martin, R. Marfil, A. Bandera and F. Sandoval

10. An Overview of Advances of Pattern Recognition Systems in Computer Vision **169**
Kidiyo Kpalma and Joseph Ronsin

11. Robust Microarray Image Processing **195**
Eugene Novikov and Emmanuel Barillot

12. Computer Vision for Microscopy Applications **221**
Nikita Orlov, Josiah Johnston, Tomasz Macura, Lior Shamir and Ilya Goldberg

13. Wavelet Evolution and Flexible Algorithm for Wavelet Segmentation, Edge Detection and Compression with Example in Medical Imaging **243**
Igor Vujovic, Ivica Kuzmanic, Mirjana Vujovic, Dubravka Pavlovic and Josko Soda

14. Compression of Spectral Images **269**
Arto Kaarna

15. Data Fusion in a Hierarchical Segmentation Context: The Case of Building Roof Description **299**
Frederic Bretar

16. Natural Scene Text Understanding **307**
Celine Mancas-Thillou and Bernard Gosselin

17. Image Similarity based on a Distributional "Metric" for Multivariate Data **333**
Christos Theoharatos, Nikolaos A. Laskaris,
George Economou and Spiros Fotopoulos

18. The Theory of Edge Detection and Low-level Vision in Retrospect **352**
Kuntal Ghosh, Sandip Sarkar and Kamales Bhaumik

19. Green's Functions of Matching Equations: A Unifying Approach for Low-level Vision Problems **381**
Jose R. A. Torreo, Joao L. Fernandes, Marcos S. Amaral and Leonardo Beltrao

20. Robust Feature Detection Using 2D Wavelet Transform under Low Light Environment **397**
Youngouk Kim, Jihoon Lee, Woon Cho,
Changwoo Park, Changhan Park and Joonki Paik

21. Genetic Algorithms: Basic Ideas, Variants and Analysis **407**
Sharapov R.R.

22. Genetic Algorithm for Linear Feature Extraction **423**
Alberto J. Perez-Jimenez and Juan Carlos Perez-Cortes

23. Recognition of Partially Occluded Elliptical Objects using Symmetry on Contour **437**
June-Suh Cho and Joonsoo Choi

24. Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics **451**
Peng-Yeng Yin

25. Pseudogradient Estimation of Digital Images Interframe Geometrical Deformations **465**
A.G. Tashlinskii

26. Anisotropic Filtering Techniques applied to Fingerprints **495**
Shlomo Greenberg and Daniel Kogan

27. Real-Time Pattern Recognition with Adaptive Correlation Filters **515**
Vitaly Kober, Victor H. Diaz-Ramirez,
J. Angel Gonzalez-Fraga and Josue Alvarez-Borrego

1

Energy Feature Integration for Motion Segmentation

Raquel Dosil, Xosé R. Fdez-Vidal, Xosé M. Pardo & Antón García
Universidade de Santiago de Compostela
Spain

1. Introduction

This chapter deals with the problem of segmentation of apparent-motion. Apparent-motion segmentation can be stated as the identification and classification of regions undergoing the same motion pattern along a video sequence. Motion segmentation has a great importance in robotic applications such as autonomous navigation and active vision. In autonomous navigation, motion segmentation is used in identifying mobile obstacles and estimating their motion parameters to predict trajectories. In active vision, the system must identify its target and control the cameras to track it. Usually, segmentation is based on some low level feature describing the motion of each pixel in a video frame. So far, the variety of approaches to deal with the problems of motion feature extraction and motion segmentation that has been proposed in literature is huge. However, all of them suffer from different shortcomings and up to date there is no completely satisfactory solution.

Recent approaches to motion segmentation include, for example, that of Sato and Aggarwal (Sato & Aggarwal, 2004), where they define the Temporal Spatio-Velocity (TSV) transform as a Hough transform evaluated over windowed spatio-temporal images. Segmentation is accomplished by thresholding of the TSV image. Each resulting blob represents a motion pattern. This solution has proved to be very robust to occlusions, noise, low contrast, etc. Its main drawback is that it is limited to translational motion with constant velocity.

It is very common to use a Kalman filter to estimate velocity parameters from intensity observations (Boykov & Huttenlocher, 2000). Kalman filtering alone presents severe problems with occlusions and abrupt changes, like large inter-frame displacements or deformations of the object. If a prior model is available, the combined use of Kalman filtering and template matching is the typical approach to deal with occlusions. For instance, Kervrann and Heitz (1998) define an a priori model with global and local deformations. They apply matching with spatial features for initialization and reinitialization of global rigid transformation and local deformation parameters in case of abrupt changes and Kalman filtering for tracking otherwise. Nguyen and Smeulders (2004) perform template matching and updating by means of Kalman filtering.

Template matching can deal even with total occlusions during a period of several frames. Nevertheless, when no prior model is available, the most common approach is statistical region classification, like Bayesian clustering (Chang et al., 1997; Montoliu and Pla, 2005). These techniques are very sensitive to noise and aliasing. Furthermore, they do not provide

a method for correlating the segmentations obtained for different frames to deal with tracking. Tracking is straightforward when the identified regions keep constant motion parameters along the sequence and different objects undergo different motion patterns. Otherwise, it is difficult to know the correspondences between the regions extracted from different frames, especially when large displacements or occlusions take place.

An early approach by Wang and Adelson (1994) tackle this issue using a layered representation. Firstly, they perform motion segmentation by region clustering under affine motion constraints. Layers are then determined by accumulating information about different regions from different frames. This information is related to texture, depth and occlusion relationships. The main limitations of this model, that make it unpractical in most situations, are that it needs a large number of frames to compute layers and significant depth variations between layers.

A very appealing alternative for segmentation is the application of an active model at each frame guided by motion features or a combination of motion and static features (Paragios & Deriche, 2000). Deformable models are able to impose continuity and smoothness constraints while being flexible.

The performance of any segmentation technique is strongly dependent on the chosen low-level features to characterize motion. In segmentation using active models, low-level features are employed to define the image potential. The simplest approach uses temporal derivatives as motion features, as in the work of Paragios and Deriche (2000). They use de inter-frame difference to statistically classify image points into static or mobile. Actually, the inter-frame difference is not a motion estimation technique, since it only performs motion detection without modelling it. It can only distinguish between static and mobile regions. Therefore, this method is only valid for static background scenes and can not classify motion patterns according to their velocity and direction of motion.

Most motion segmentation models are based on the estimation of optical flow, i.e., the 2D velocity of image points or regions, based on the variation of their intensity values. Mansouri and Konrad (2003) have employed optical flow estimation to segmentation with an active model. They propose a competition approach based on a level set representation. Optimization is based on a maximum posterior probability criterion, leading to an energy minimization process, where energy is associated to the overall residuals of mobile objects and static background. Residuals are computed as the difference between measured intensities and those estimated under the constraint of affine transformation motion model. However, optical flow estimations present diverse kinds of problems depending on the estimation technique (Barron et al., 1994; Stiller & Konrad, 1999). In general, most optical flow estimation techniques assume brightness constancy along frames, which in real situations does not always hold, and restrict allowed motions to some specific model, such as translational or affine motion. Particularly, differential methods for estimating the velocity parameters consistent with the brightness constancy assumption are not very robust to noise, aliasing, occlusions and large inter-frame displacements.

Alternatively, energy filtering based algorithms (Heeger, 1987; Simoncelli & Adelson, 1991; Watson & Ahumada, 1985; Adelson & Bergen, 1985; Fleet, 1992) estimate motion from the responses of spatio-temporal filter pairs in quadrature, tuned to different scales and orientations. Spatio-temporal orientation sensitivity is translated into sensitivity to spatial orientation, speed and direction of motion. These techniques are known to be robust to noise and aliasing, to give confident measurements of velocity and to allow an easy treatment of

the aperture problem, i.e., the reliable estimation of the direction of motion. However, to the best of our knowledge there is not motion segmentation method based on energy filtering.
Another important subject in segmentation with active models is how to initialize the model at each frame. A common solution is to use the segmentation of each frame to initialize the model at the next frame. Paragios and Deriche (2000) use this approach. The first frame is automatically initialized based on the inter-frame difference between the first two frames. The main problem of the initialization with the previous segmentation arises with total occlusions, when the object disappears from the scene for a number of frames, since no initial state is available when the object reappears. The case of large inter-frame displacements is also problematic. The object can be very distant from its previous position, so that the initial state might not be able converge to the new position. Tsechpenakis et al. (2004) solve these problems by initializing each frame, not using the previous segmentation, but employing the motion information available for that frame. In that work, motion features are only employed for initialization and the image potential depends only on spatial information.

1.2 Our Approach

In this chapter we present a model for motion segmentation that combines an active model with a low-level representation of motion based on energy filtering. The model is based solely on the information extracted from the input data without the use of prior knowledge. Our low level motion representation is obtained from a multiresolution representation by clustering of band-pass versions of the sequence, according to a criterion that links bands associated to the same motion pattern. Multiresolution decomposition is accomplished by a bank of non-causal spatio-temporal energy filters that are tuned to different scales and spatio-temporal orientations. The complex-valued volume generated as the response of a spatio-temporal energy filter to a given video sequence is here called a *band-pass feature*, *subband feature*, *elementary energy feature* or simply *energy feature*. We will call *integral features, composite energy features* or simply *composite features* to motion patterns with multiple speed, direction and scale contents generated as a combination of elementary energy features in a cluster. The set of filters associated to an energy feature cluster are referred to as *composite-feature detector*. Segmentation is accomplished using composite features to define the image potential and initial state of a geodesic active model (Caselles, 1997) at each frame. The composite feature representation will be applied directly, without estimating motion parameters.
Composite energy features have proved to be a powerful tool for the representation of visually independent spatial patterns in 2D data (Rodriguez-Sánchez et al., 1999), volumetric data (Dosil, 2005; Dosil et al., 2005b) and video sequences (Chamorro-Martínez et al., 2003). To identify relevant composite features in a sequence, it is necessary to define an integration criterion able to relate elementary energy features contributing to the same motion pattern. In previous works (Dosil, 2005; Dosil et al., 2005a; Dosil et al., 2005b), we have introduced an integration criterion inspired in biological vision that improves the computational cost and performance of earlier approaches (Rodriguez-Sánchez et al., 1999; Chamorro-Martínez et al., 2003). It is based on the hypothesis of Morrone and Owens (1987) that the Human Visual System (HVS) perceives features at points of locally maximal Phase Congruence (PC). PC is the measure of the local degree of alignment of the local phase of Fourier components of a signal. The sensitivity of the HVS to PC has also been

studied by other authors (Fleet, 1992; Oppenheim & Lim, 1981; Ross et al., 1989; du Buf, 1994). As demonstrated by Venkatesh and Owens (1990), points whose PC is locally maximal coincide with the locations of energy maxima. Our working hypothesis is that local energy maxima of an image are associated to locations where a set of multiresolution components of the signal contribute constructively with alignment of their local energy maxima. Hence, we can identify composite features as groups of features that present a high degree of alignment in their energy maxima. For this reason, we employ a measure of the correlation between pairs of frequency features as a measure of similarity for cluster analysis (Dosil et al., 2005a).

Here, we extend the concept of PC for spatio-temporal signals to define our criterion for spatio-temporal energy feature clustering. We will show that composite features thus defined are robust to noise, occlusions and large inter-frame displacements and can be used to isolate visually independent motion patterns with different velocity, direction and scale content.

The outline of this chapter is as follows. Section 2 is dedicated to the composite feature representation model. Section 3 is devoted to the proposed method for segmentation with active models. In section 4 we illustrate the behaviour of the model in different problematic situations, including some standard video sequences. In 5 we expound some conclusions of the work.

2. Composite-Feature Detector Synthesis

The method for extraction of composite energy features consists of the decomposition of the image in a set of band-pass features and their subsequent grouping according to some dissimilarity measure (Dosil, 2005; Dosil et al., 2005a). The set of frequency features involved in the process is determined by selecting from a predefined spatio-temporal filter bank those bands that are more likely to be associated to relevant motion patterns, which we call *active* bands. Composite-feature detectors are clusters of these active filters. Each visual pattern is reconstructed as a combination of the responses of the filters in a given cluster. Filter grouping is accomplished by applying hierarchical cluster analysis to the set of band-pass versions of the video sequence. The dissimilarity measure between pairs of frequency features is related to the degree of phase congruence between a pair of features, through the quantification of the alignment among their local energy maxima. The following subsections detail the process.

2.1 Bank of Spatio-Temporal Filters

The bank of spatio-temporal filters applied here (Dosil, 2005; Dosil et al., 2005b) uses an extension to 3D of the log Gabor function (Field, 1994). The filter is designed in the frequency domain, since it has no analytical expression in the spatial domain. Filtering is realized as the inner product between the transfer function of the filter and the Fourier transform of the sequence. Filtering in the Fourier domain is very fast when using Fast Fourier Transform and Inverse Fast Fourier Transform algorithms.

The filters' transfer function T is designed in spherical frequency coordinates as the product of separable factors R and S in the radial and angular components respectively, such that $T = R \cdot S$. The radial term R is given by the log Gabor function (Field, 1993)

$$R(\rho;\rho_i)=\exp\left(-\frac{\left(\log\left(\rho/\rho_i\right)\right)^2}{2\left(\log\left(\sigma_{\rho i}/\rho_i\right)\right)^2}\right), \tag{1}$$

where $\sigma_{\rho i}$ is the standard deviation and ρ_i the central radial frequency of the filter.
The angular component is designed to achieve orientation selectivity in both the azimuthal component ϕ_i of the filter, which reflects the spatial orientation of the pattern in a frame and the direction of movement, and the elevation component θ_i, related to the velocity of the motion pattern. For static patterns $\theta_i=0$. To achieve rotational symmetry, S is defined as a Gaussian on the angular distance α between the position vector of a given point $\mathbf{f}$ in the spectral domain and the direction of the filter $\mathbf{v}=(\cos\phi_i\cdot\cos\theta_i,\cos\phi_i\cdot\sin\theta_i,\sin\phi_i)$ (Faas & van Vliet, 2003)

$$S(\phi,\theta;\phi_i,\theta_i)=S(\alpha)=\exp\left(-\alpha^2/2\sigma_{\alpha i}^2\right),\ \text{ with } \alpha(\phi_i,\theta_i)=\operatorname{acos}\left(\mathbf{f}\cdot\mathbf{v}/\|\mathbf{f}\|\right), \tag{2}$$

where $\mathbf{f}$ is expressed in Cartesian coordinates and $\sigma_{\alpha i}$ is the angular standard deviation.
Active filters are selected from a predefined band partition of the 3D frequency space. Frequency bands are determined by the central frequency (ρ_i,ϕ_i,θ_i) of the filters and their width parameters $(\sigma_{\rho i},\sigma_{\alpha i})$. In the predefined bank, frequency is sampled so that $\rho_i=\{1/2, 1/4, 1/8, 1/16\}$ in pixels^{-1}. Parameter $\sigma_{\rho i}$ is determined for each band in order to obtain 2 octave bandwidth. θ_i is sampled uniformly while the number of ϕ_i samples decreases with elevation in order to keep the "density" of filters constant, by maintaining equal arc-length between adjacent ϕ_i samples over the unit radius sphere. Following this criterion, the filter bank has been designed using 23 directions, i.e. (ϕ_i,θ_i) pairs, yielding 92 bands. $\sigma_{\alpha i}$ is set to 25º for all orientations. Hence, the bank involves 4×23 filters that yield a redundant decomposition and cover a wide range of the spectrum.

2.2 Selection of Active Bands

To achieve improved performance, it is convenient to reduce the number of bands involved in cluster analysis. The exclusion of frequency channels that are not likely to contribute to motion patterns facilitates the identification of clusters associated to composite motion features. Furthermore, it reduces computational cost. Here, we have introduced a channel selection stage based on a statistical analysis of the amplitude responses of the band-pass features. Selected channels are called *active*.
Our method for the selection of active channels is based on the works of Field (1994) and Nestares et al. (2004). Field has studied the statistics of the responses of a multiresolution log-Gabor wavelet representation scheme that resembles the coding in the visual system of mammalians. He has observed that the filter responses histograms are not Gaussian, but leptokurtic distributions –pointed distributions with long tails–, revealing the sparse nature of both the sensory coding and the features from natural images. According to Field, when the parameters of the wavelet codification fit those in the mammalian visual system, the histogram of the responses is highly leptokurtic. This is reflected in the fourth cumulant of the distribution. Namely, he uses the kurtosis to characterize the sparseness of the response.

Regarding spatio-temporal analysis, Nestares et al. (2000) applied channel selection to a bank of spatio-temporal filters, with third order Gaussian derivatives as basis functions, based on the statistics of filters responses. They have observed that features corresponding to mobile targets present sparser responses than those associated to background –weather static or moving. This fact is illustrated in Fig. 1. They measure different statistical magnitudes reflecting sparseness of the amplitude response, realize a ranking of the channels based on such measures and perform channel selection by taking the n first channels in the ranking, where n is a prefixed number.
Based on these two works, we have designed our filter selection method. The statistical measure employed to characterize each channel is the kurtosis excess γ_2

$$\gamma_2 = k_4 / k_2^2 - 3 \tag{3}$$

where k_4 and k_2 are respectively the fourth and second cumulants of a histogram. If the kurtosis excess takes a positive value, the distribution is called leptokurtic and presents a

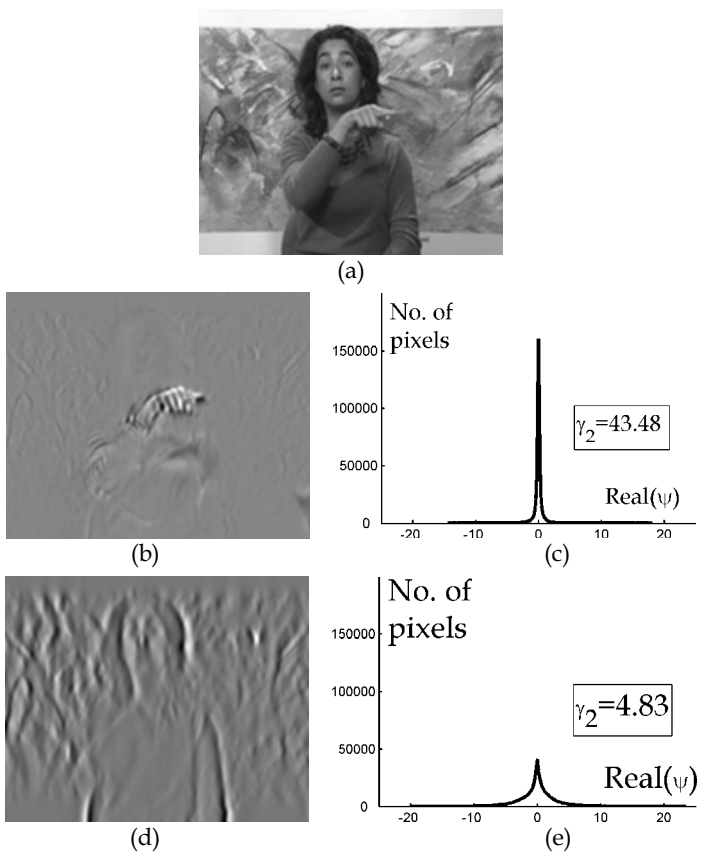


Fig. 1. (a) A frame of the standard sequence *Silent*, showing a moving hand. (b) and (d) A frame of the real component of two band-pass features of the *Silent* video sequence. (c) and (e) Histograms corresponding to band-pass features in (b) and (d)

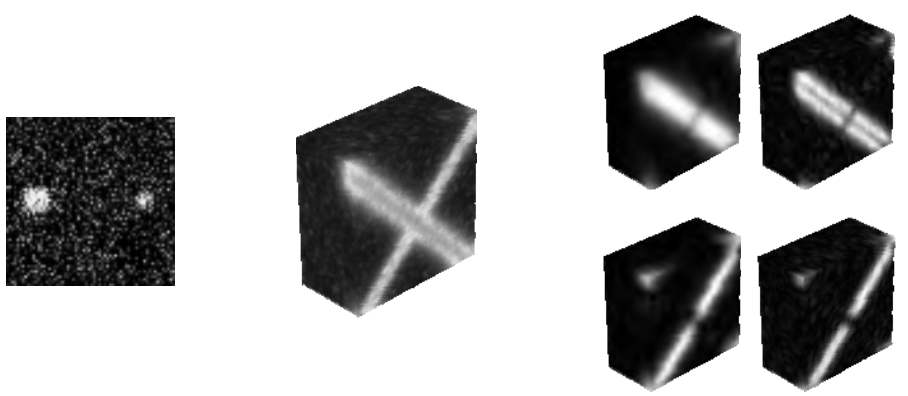
narrow peak and long tails. If it is negative, the distribution is called platykurtic and presents a broad central lobe and short tails. Distributions with zero kurtosis excess, like the Gaussian distribution, are called mesokurtic.
We measure γ_2 for both the real and imaginary components of each feature ψ_i and then compose a single measure δ

$$\delta_i = \gamma_2(\mathrm{Re}(\psi_i)) + \gamma_2(\mathrm{Im}(\psi_i)) \tag{4}$$

Instead of selecting the n first channels in the ranking of δ, we perform cluster analysis to identify two clusters, one for active channels with large values of δ and another for non active channels. Here, we have applied a k-means algorithm. The cluster of active channels is identified as the one with larger average δ.

2.3 Energy Feature Clustering

Integration of elementary features is tackled in a global fashion, not locally (point-wise). Besides computational efficiency, this provides robustness, since it intrinsically correlates same-pattern locations –in space and time–, avoiding grouping of disconnected regions.
As aforementioned, it seems plausible that the visual system of humans perceives features where Fourier components are locally in phase (Morrone & Owens, 1987). Our criterion for integration of frequency features is based on the assumption that a maximum in phase congruence implies the presence of maxima in the same location in a subset of subband versions of the data. Points of locally maximal phase congruence are also points of locally maximal energy density (Venkatesh & Owens, 1990). Hence, subband images contributing



(a) (b) (c)

Fig. 2. (a) A frame of a synthetic video sequence, where two light spots move from side to side with opposite direction. (b) A cut along the temporal axis of the total energy of the sequence. (c) Energy of some band-pass versions of the sequence. Those on top row correspond to one of the spots and present some degree of concurrence on their local energy maxima. Bottom row shows two band-pass features correspondent to the other motion pattern.

to the same visual pattern should present a large degree of alignment in their local energy maxima, i.e., their energy maxima present some degree of concurrence –see Fig. 2.
Here, the dissimilarity between two subband features is determined by estimating the degree of alignment between the local maxima of their local energy. Alignment is quantified using the correlation coefficient ρ of the energy maps of each pair $\{\psi_i, \psi_j\}$ of subband features. This measure has proved to produce good results in visual pattern extraction from volumetric data (Dosil, 2005; Dosil et al., 2005b). If $A(\psi) = ||\psi|| = (\mathrm{Im}(\psi)^2 + \mathrm{Re}(\psi)^2)^{1/2}$, the actual distance is calculated from $\rho(A_i, A_j)$ as follows

$$D_\rho(A_i, A_j) = \left(1 - \sqrt{\left(1 + \rho(A_i, A_j)\right)/2}\right)^2 . \tag{5}$$

This distance function takes values in the range [0,1]. The minimum value corresponds to perfect match of maxima –linear dependence with positive slope– and the maximum corresponds to the case of perfect fit with negative slope, like, for example, an image and its inverse. This measure does not depend on the selection of any parameter and does not involve the discrete estimation of joint and/or marginal probabilities –histograms.
Our approach generates visual patterns by clustering of active bands. Dissimilarities between each pair of frequency features are computed to build a dissimilarity matrix. To determine the clusters from the dissimilarity matrix, a hierarchical clustering method has been chosen, using a Ward's algorithm to determine inter-cluster distance, which has proved to improve other metrics (Jain & Dubes, 1988). The number of clusters N_c that a hierarchical technique generates is an input parameter of the algorithm. The usual strategy to determine the N_c is to run the algorithm for each possible N_c and evaluate the quality of each resulting configuration according to a given validity index. A modification of the Davies-Boulding index proposed by Pal and Biswas (1996) has proved to produce good results for our application. It is a graph-theory based index that measures the compactness of the clusters in relation to their separation.
A stage of cluster merging follows cluster analysis. Clusters with average intercluster correlation values close to one –specifically, greater than 0.75– are merged to form a single cluster. This is made since we can not evaluate the quality of a single cluster containing all features. Besides, hierarchical algorithms can only analyse the magnitude of a distance in relation to others, not in an absolute fashion. This fact is often a cause of wrong classification, splitting clusters into smaller subgroups.

2.4 Composite Feature Reconstruction

The response ψ to an energy filter is a complex-valued sequence, where the real and imaginary components account for even and odd symmetric features respectively. In this section we describe how elementary complex features in a cluster are combined to obtain a composite-feature Ψ. We will use real, imaginary or amplitude representations depending on the application. For simple visualization we will employ only the real components. In the definition of the image potential of an active model, we are only interested on odd-symmetric components, which represent mobile contours, so only the imaginary parts of the elementary features will be involved. For initialization we are interested in the regions occupied by the moving objects, so the amplitude of the responses $||\psi||$ is the chosen representation.
Here we define the general rule for the reconstruction of Ψ based on a given representation

E of the responses of the filters, that can be either $\mathrm{Re}(\psi)$, $\mathrm{Im}(\psi)$ or the amplitude $A(\psi) = ||\psi|| = (\mathrm{Im}(\psi)^2 + \mathrm{Re}(\psi)^2)^{1/2}$. The easiest way of constructing the response Ψ of a set Ω_j of filters in a cluster j is by linear summation

$$\Psi^j(x,y,t) = \sum_{i \in \Omega_j} E_i(x,y,t). \tag{6}$$

However, simple summation presents one important problem. There might be features in the cluster that contribute, not only to the corresponding motion pattern, but also to other patterns or static structures in the sequence. Only points with contributions from all features in the cluster should have a non null response to the composite feature detector. To avoid this problem, we define the composite feature as the linear summation of elementary features weighted by a mask indicating locations with contribution of all features in the clustering. The mask is constructed as the summation of the thresholded responses $\widetilde{E}_i$ of the elementary features, normalized by the total number of features. Thresholding is accomplished by applying a sigmoid to the responses of elementary features, so that $\widetilde{E}_i \in [0,1]$. Therefore, the mask takes value 1 wherever all features contribute to the composite pattern and rapidly decreases otherwise, with a smooth transition

$$\Psi^j(x,y,t) = \frac{\sum_{i \in \Omega_j} \widetilde{E}_i}{Card(\Omega_j)} \sum_{i \in \Omega_j} E_i, \tag{7}$$

where Ω_j is the set of all bands in cluster j. The effect of masking is illustrated in Fig. 3. Reconstruction using different representations for E is illustrated in Fig. 4.
For visualization purposes, we will employ the real component $\Psi^j_{even} = \Psi^j(E_i = \mathrm{Re}(\psi_i))$. The odd-symmetric representation of Ψ is constructed by full-wave rectification of expression in equation (7), so that $\Psi^j_{odd} = \left|\Psi^j(E_i = \mathrm{Im}(\psi_i))\right|$ does not have into account the sign of the contour. The amplitude representation $\Psi^j_{amp} = \Psi^j(E_i = \|\psi_i\|)$ is used for initialization in general situations. The even-symmetric representation is used for initialization of objects with uniform contrast and is defined by applying a half-wave rectification $\max(\pm\Psi^j_{even}, 0)$, with sign depending on the specific contrast.

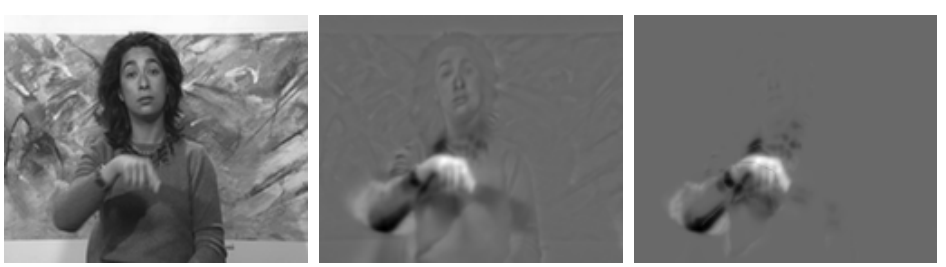


Fig. 3. A frame of the "silent" video sequence: *Left*: Input data. *Centre*: Even-symmetric representation of the response of one of the composite features detected, corresponding to the moving hand, calculated using equation (6) and, *Right*: using equation (7)

3. Motion Pattern Segmentation

The previously described method for feature clustering is able to isolate different static and dynamic patterns from a video sequence. Nevertheless, it is not suitable by itself to segment mobile objects for several reasons. To begin with, the mobile contours might present low contrast in some regions, giving place to disconnected contours. Furthermore, when the moving object is occluded by static objects, its contour presents static parts, so that the representation with motion patterns is incomplete. This happens also when a contour is oriented in the direction of motion; only the motion of the beginning and end of the segment is detected. For these reasons, we will produce a higher-level representation of the motion patterns from the proposed low-level motion representation.
In this work we have chosen an active model as a high level representation technique, namely, the geodesic active model. We will perform a segmentation process for each composite feature, which we will refer to as Ψ, omitting the superindex. From that pattern, we derive the initial state of the model and the image potential in each frame. After evolving a geodesic model in each frame, the segmented sequence is generated by stacking the segmented frames. A scheme of the segmentation method is presented in Fig. 5. Next subsections describe the technique in depth.

3.1 Geodesic Active Model

To accomplish segmentation, here we have chosen an implicit representation for object boundaries, where the contour is defined as the zero level set of an implicit function. Implicit active models present important advantages regarding parametric representations. The problem of contour re-sampling when stretching, shrinking, merging and splitting is avoided. They allow for the simultaneous detection of inner and outer contours of an object and naturally manage topological changes. Inner and outer regions are determined by the sign of the implicit function.

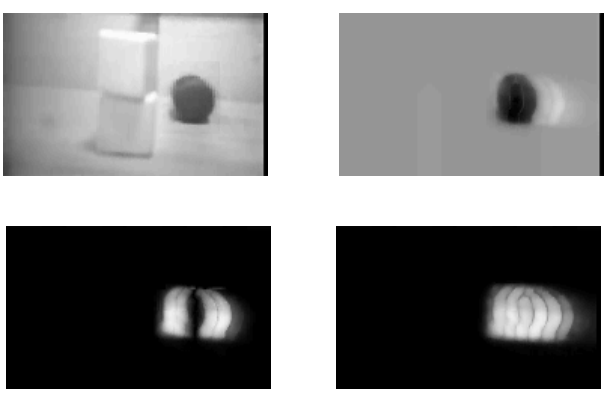


Fig. 4. Top left: One frame of an example sequence with a moving dark cylinder. The remainder images show different representations for one of the composite features identified by the presented representation method. Top right: Even representation. Bottom left: Odd representation. Bottom right: Amplitude representation.

The optimization model employed for segmentation is the geodesic active model (Caselles et al., 1997). The evolution of the contour is determined from the evolution of the zero-level set of an implicit function representing the distance u to the contour. Let $\Omega := [0, a_x] \times [0, a_y]$ be the frame domain and consider a scalar image $u_0(x, y)$ on Ω. We employ here symbol τ for time in the evolution equations of u to distinguish it from the frame index t. Then, the equations governing the evolution of the implicit function are the following:

$$\begin{aligned} u(x, y, t = t_k, \tau = 0) &= u_0(x, y, t = t_k) \quad \text{on } \Omega \\ \frac{\partial u}{\partial \tau} &= g(s)|\nabla u|(\kappa + c) + \nabla g(s) \nabla u \quad \text{on } \Omega \times (0, \infty) \end{aligned}, \tag{8}$$

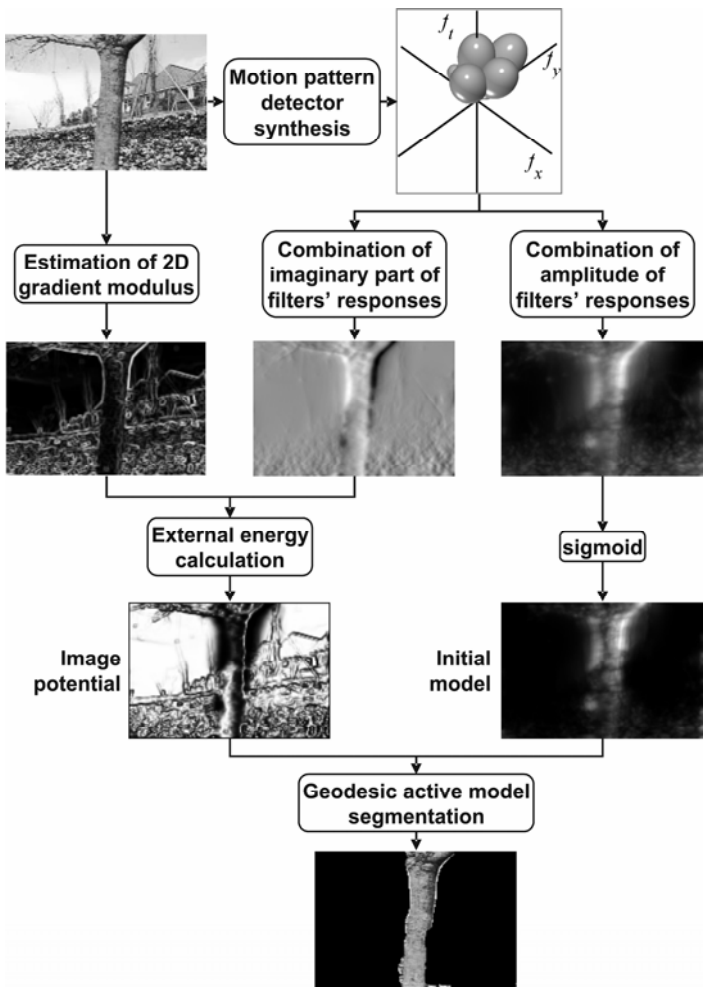


Fig. 5. Scheme of the segmentation technique

where c is real constant, g is a function with values in the interval [0, 1] that decreases in the presence of relevant image features, s is the selected image feature and κ is the curvature.
If the second term in the right side of the previous equation is not considered, what remains is the expression for the *geometric active model*, where $g \cdot (\kappa + c)$ represents the velocity of the evolving contour. The role of the curvature can be interpreted as a geometry dependent velocity. Its effect is also equivalent to the internal forces in a thin-plate-membrane spline model, also called *snake* (Kass et al., 1988). Constant c represents a constant velocity or *advection* velocity in the geometric active model and is equivalent to a balloon force in the snake model. Factor $g(s)$ has the effect of stopping the contour at the desired feature locations. The second term in the right side is the image dependent term, which pushes the level-set towards image features. It is analogous to external forces in the snake formulation. This term did not appear in the geometric active model, which made necessary the use of a constant velocity term to approach the level-set to the object boundary. With the use of a feature attraction term this is no longer necessary. However, if the model is initialized far away from the image features to be segmented, the attraction term may not have enough influence over the level-set. As a result, the constant velocity term is often used to compensate for the lack of an initialization stage.
The concrete implementation of the geodesic active model used here is the one described in (Weickert & Kühne, 2003). We do not employ balloon forces, since with the initialization, described in subsection 3.3, they are no longer needed, so then $c = 0$. In the following subsection we define the image potential as a function of the composite energy features.

3.2 Image Potential Definition

The expression for the image potential function is the same as in (Weickert & Kühne, 2003)

$$g(s) = \frac{1}{1 + (s/s_{\min})^p}, \tag{9}$$

with p and $s_{\min}$ being real constants.
The potential of the mobile contour depends on the odd-symmetric representation of the motion pattern, Ψ_{odd}, reconstructed as the rectified sum of the imaginary components of the responses to its constituent filters. This motion pattern may present artefacts, due to the diffusion of patterns from neighbouring frames produced when applying energy filtering. This situation is illustrated in Fig. 6.a and b. To minimize the influence of these artefacts, the motion pattern is modulated by a factor representing the localization of spatial contours. It is calculated from the 2D contour detector response by thresholding using a sigmoid function

$$C_m(x, y, t_k) = \frac{1}{1 + \exp(-K(C_s(x, y, t_k) - C_0))} \frac{\Psi_{odd}(x, y, t_k)}{\max(\Psi_{odd}(x, y, t_k))}, \tag{10}$$

where C_s is a spatial contour detector based on the frame gradient, C_0 is the gradient threshold and K is a positive real constant. The specific values taken here are $C_0 = 0.1$ and $K = 20$. The effect of this modulation can be observed in Fig. 6.c and d.
Although here we are interested in segmenting objects based on their motion features, it is convenient to include a spatial term in the potential. This is necessary to close the contour when part of the boundary of the moving object remains static –when there is a partial

occlusion by a static object or scene boundary or when part of the moving contour is parallel to the direction of motion. Therefore, the image feature s is the weighted sum of two terms, C_m and C_s, respectively related to spatio-temporal and pure spatial information.

$$s = w_s C_s + w_m C_m \text{, with } w_s + w_m = 1 \text{ and } w_s, w_m > 0 \tag{11}$$

The weight of the spatial term w_s must be much smaller than the motion term weight w_m, so that the active model does not get "hooked" on a static contours not belonging to the target object. Here, the values of the weights have been set as follows: $w_s = 0.1$ and $w_m = 0.9$.
The spatial feature employed to define the spatial potential is the regularized image gradient. Regularization of a frame is accomplished here by feature-preserving 2D anisotropic diffusion, which brakes diffusion in the presence of contours and corners. The 3D version of the filter is described in (Dosil & Pardo, 2003). If $I^*(x, y, t_k)$ is the smoothed version of the kth frame, then

$$C_s(x, y, t_k) = \left\| \nabla I^*(x, y, t_k) \right\| \Big/ \max\left(\left\| \nabla I^*(x, y, t_k) \right\| \right) \tag{12}$$

In the potential function g, $p = 2$ and $s_{\min}$ is calculated so that, on average, $g(s(x, y)) = 0.01$, $\forall x, y : C_m(x, y) > 0.1$. Considering the geodesic active model in a front propagation framework, $g = 0.01$ means a sufficiently slow speed of the propagating front to produce stopping in practical situations.

3.3 Initialization

The initial state of the geodesic active model is defined, in a general situation, from the amplitude representation of the selected motion pattern Ψ_{amp} unless other solution is specified. To enhance the response of the cluster we apply a sigmoid thresholding to Ψ_{amp}. The result is remapped to the interval [–1,1]. The zero-level of the resulting image is the initial state of the contour.

$$u_0(x, y, t_k) = \frac{2}{1 + \exp\left(-K\left(\Psi_{amp}(x, y, t_k) - \Psi_0 \right) \right)} - 1 \tag{13}$$

When the object remains static during a number of frames the visual pattern has a null response. For this reason, the initial model is defined as the weighted sum of two terms, respectively associated to the current and previous frames. The contribution from the previous frame must be very small.

$$u_0(x, y, t_k) = w_k \left(\frac{2}{1 + \exp\left(-K\left(\Psi_{amp}(x, y, t_k) - \Psi_0 \right) \right)} - 1 \right) + w_{k-1}\, u_{\tau = \tau_{\max}}(x, y, t_{k-1}) \tag{14}$$

with w_k and w_{k-1} being positive real constants that verify $w_k + w_{k-1} = 1$. In the experiments presented in next section, $w_k = 0.9$, $w_{k-1} = 0.1$, $K = 20$ and $\Psi_0 = 0.1$.

4 Results

In this section, some results are presented to show the behaviour of the method in problematic situations. The results are compared to an alternative implementation that

employs typical solutions for initialization and definition of image potential in a way similar to that of Paragios and Deriche (2000): the initial state is the segmentation of the previous frame and the image potential depends on the inter-frame difference. However, instead of defining the image potential from the temporal derivative using a Bayesian classification, the image potential is the same as with our method, except that the odd-symmetric representation of the motion pattern is replaced by the inter-frame difference $I_t(x,y,t_k)=I(x,y,t_k)-I(x,y,t_{k-1})$. This is to compare the performance of our low-level features with inter-frame difference under the equal conditions. The initial state for the first frame is defined by user interaction.
The complete video sequences with the original data and the segmentation results are available at http://www-gva.dec.usc.es/~rdosil/motion_segmentation_examples.htm.They are summarized in the next subsections.

4.1 Moving Background

In this example, we use part –27 frames– of the well-known sequence "flower garden". It is a static scene recorded by a moving camera –see Fig. 7. The estimation of the inter-frame difference along frames produces large values at every image contour. The temporal derivative can be thresholded, or more sophisticated techniques for classifying regions into mobile or static can be employed, as in (Paragios & Deriche, 2000). However, it is not possible to isolate independent motion patterns just from the information brought by I_t. In contrast, visual pattern decomposition allows isolation of motion patterns with different speeds, which in the 3D spatio-temporal domain is translated into patterns with different orientations. This is made clear visualizing a cut of the image and the motion patterns in the *x*-*t* plane, as in Fig. 8.
Consequently, the image potential estimated from the temporal derivative feature presents

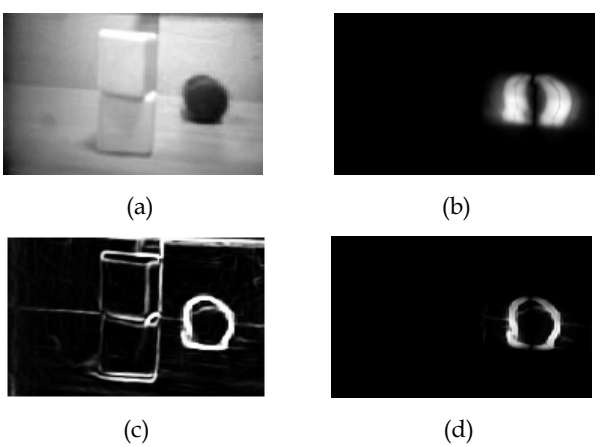


Fig. 6. (a) One frame of an example sequence where the dark cylinder is moving from left to right. For one of the composite features detected: (b) Ψ_{odd} representation. (c) Gradient after sigmoid thresholding. (d) Motion feature C_m from equation (10) as the product of images (b) and (c).

deep minima all over the image and the active model is not able to distinguish foreground objects from background, as can be seen in Fig. 9. The image potential in our implementation considers only the motion pattern corresponding to the foreground object, leading to a correct segmentation, as shown in Fig. 5.

4.2 Large Inter-Frame Displacements

When the sampling rate is too small in relation to the speed of the moving object, it is difficult to find the correspondence between the positions of the object in two consecutive frames. Most optical flow estimation techniques present strong limitations in the allowed displacements. Differential methods, based on the brightness constancy assumption, try to find the position of a pixel in the next frame imposing some motion model. Frequently, the search is restricted to a small neighborhood. This limitation can be overcome by coarse-to-fine analysis or by imposing smoothness constraints (Barron et al., 1994). Still, large displacements are usually problematic. The Kalman filter is not robust to abrupt changes when no template is available (Boykov & Hutterlocher, 2000).

When using the inter-frame difference in combination with an active model, the correspondence is accomplished through the evolution of the model from the previous state to the next one (Paragios & Deriche, 2000). However, when initializing with the previous segmentation, the model is not able to track the target if the previous segmentation does not intersect the object in the following frame. Fig. 10 shows an example case of this situation, taken from the standard sequence "table tennis". The alternative implementation of the active model fails to track the ball, as shown in the images of the second row of the figure. When using energy features, the composite motion patterns are isolated from each other. In this way, the correspondence of the motion estimations in different frames is naturally provided by the representation scheme, as shown in the third row of Fig. 10. This is also a property of techniques for motion estimation based on the Hough transform (Sato & Aggarwal, 2004). Nevertheless, this approach in not appropriate for this sequence, since the speed of the moving objects is variable in magnitude and direction –it is an oscillating movement –, so that it does not describe a straight line or a plane in the spatio-temporal domain –see left image in Fig. 10. Unlike the Hough transform, composite features combine elementary velocity-tuned features to deal with complex motion patterns, as can be seen in the image at the right of Fig. 10. We take advantage of both the isolation of the motion pattern and the integration of different velocity components associated to the moving objects, to initialize the model at each frame. Hence, the model arrives to a correct segmentation of the ball –see Fig. 10, bottom row– besides the large displacement produced and the changing direction or movement.

4.3 Total Occlusions

Occlusions give rise to the same problem as with fast objects. Again, initialization with composite frequency-features leads to a correct segmentation even when the object disappears from the scene during several frames. An example of this is presented in Fig. 12. In segmentation based on region classification (Chang et al., 1997; Montoliu & Pla, 2005) the statistical models extracted for each of the identified regions could be employed for tracking by finding the correspondence among them in different frames. However, occlusions carry

the additional problem of determining when the object has left the scene and when it reappears. The same problem applies for Kalman filter segmentation. Returning to the alternative implementation of the active model, when the object leaves the scene and no

Fig. 7. A frame of the "flower garden" video sequence.

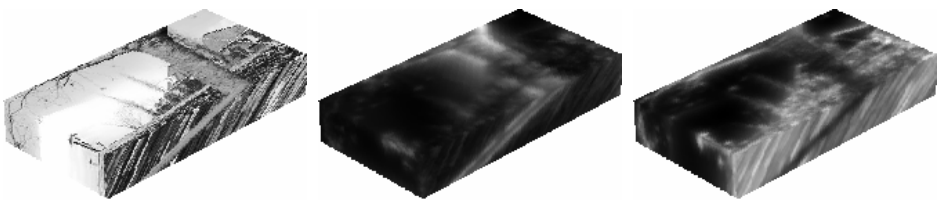


Fig. 8. A transversal cut of the original sequence: *Left*: Input data. *Centre and Right*: Ψ_{amp} of the two motion patterns isolated by the composite-feature representation model

Fig. 9. For the frame in Fig. 7, *Left*: Inter-frame difference, *Centre*: Image potential derived from I_t, *Right*: Segmentation obtained using image potential from image at the centre and initialization with the segmentation from previous frame.

other motion features are detected, the model collapses and the contour disappears from the scene in the remainder frames –see Fig. 12, second row. The solution of Paragios and Deriche could be employed to reinitialize the model by applying motion detection again, but it can not be ensured that the newly detected motion feature corresponds to the same pattern.

Again, due to the nature of our representation, the composite energy-features do not need a stage for finding correspondence between regions occupied by a motion pattern in different frames –see Fig. 12, third row. The model collapses when the cylinder disappears behind a static object and is reinitialized automatically when it reappears, without the need of a prior model. Initialization for this particular example has been achieved by using $\max(-\Psi_{even}, 0)$ of the selected composite feature, instead of the amplitude representation. This is because the target object does not present severe contrast changes in its surface, so half-wave rectification of the even-symmetric representation allows better localization of the object, facilitating convergence –the real component are inverted before rectification, since the object has negative contrast.

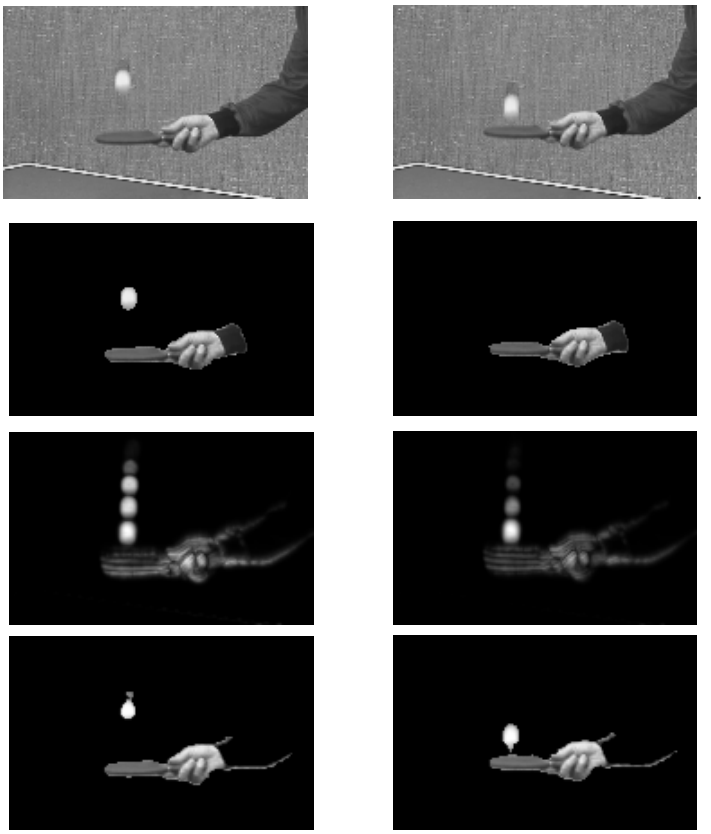


Fig. 10. *Top*: Two consecutive frames of the "table tennis" video sequence. For frames on top row, *2nd Row*: Segmentations produced by the alternative active model, *3rd Row*: Ψ_{amp} of the selected composite-feature, *Bottom*: Segmentation obtained with one of the detected composite-features

Fig. 13 shows another example presenting occlusions where the occluding object is also mobile. As can be seen, the alterative active model fails in segmenting both motion patterns, both due to initialization with previous segmentation and incapability of distinguishing both motion patterns, while our model properly segments both patterns using the composite-features provided by our representation scheme.

4.4 Complex Motion Patterns

The following example shows the ability of the method to deal with complex motion patterns and complex scenarios. In particular, the following sequence, a fragment of the standard movie know as "silent", presents different moving parts, each one with variable speed and direction and deformations as well, over a textured static background. As can be seen in the images from the top row of Fig. 14, the motion pattern of the hand can not be properly described by an affine transformation. Moreover, the brightness constancy assumption is not verified here.

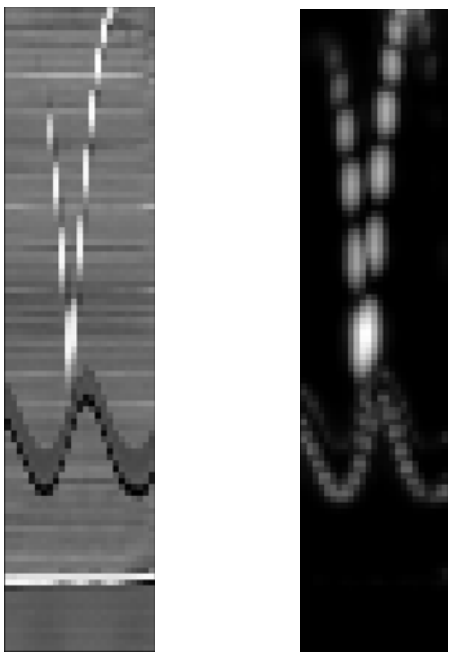


Fig. 11. Left: A cut of the "table tennis" sequence in the x-t plane. The white pattern corresponds to the ball and the gray/black sinusoidal pattern bellow corresponds to the bat. Right: A cut of the in the x-t plane of the Ψamp representation of the composite feature used in segmentation in bottom row of Fig. 14.

The active model based on the inter-frame difference is not able to properly converge to the contour of the hand, as seen in second row of Fig. 14. This is due to both the interference of other moving parts or shadows and wrong initialization. From the results, it can be seen that, despite the complexity of the image, the composite-feature representation model is able to isolate the hand and properly represent its changing shape in different frames – Fig. 14.

4.5 Discussion

In the examples presented, it can be observed that the proposed model for the representation of motion is able to group band-pass features associated to visually independent motion patterns without the use of prior knowledge. It must be said that the multiresolution scheme defined in section 2.1 has a great influence in the results, specially the selection of the number of filters and the angular bandwidth, which is related to the ability of the model to discriminate between different but proximal orientations, speeds and directions of motion.

In the comparison with the alternative implementation, which uses typical solutions for initialization and image potential definition, the proposed approach outperforms. Although there are other approaches that may present improved performance in solving some of the reported problems, it seems that none of them can successfully deal with all of them.

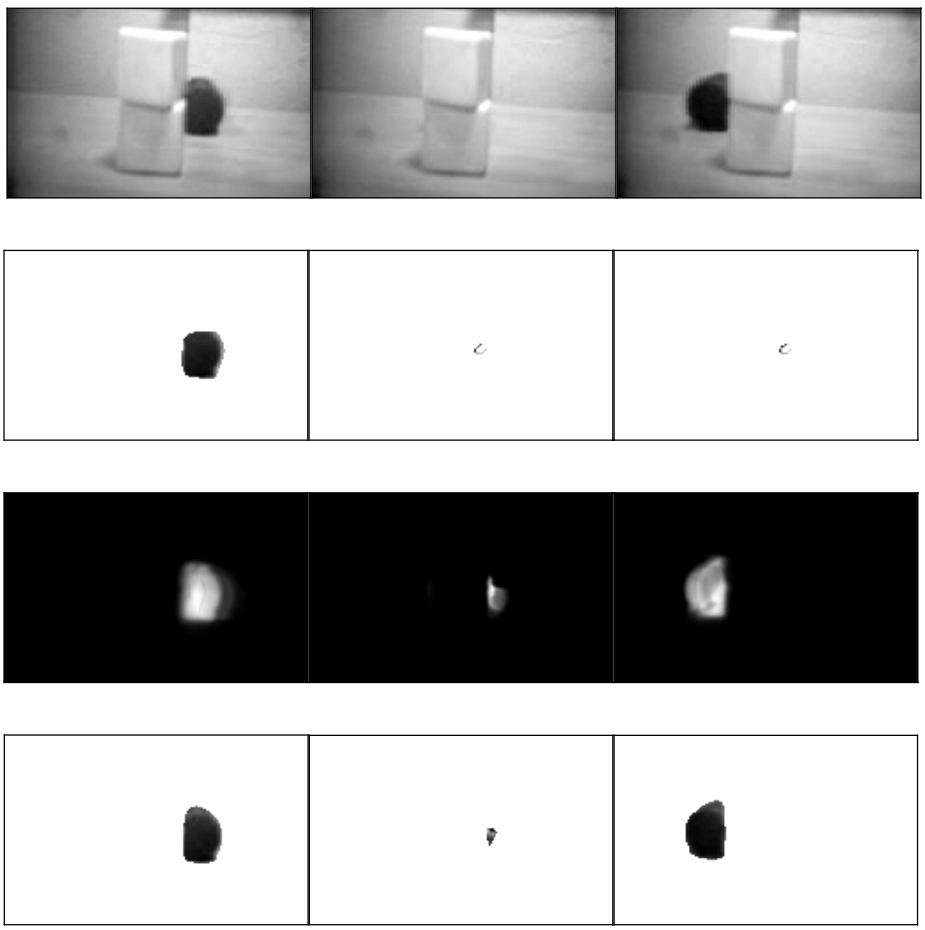


Fig. 12. *Top:* Three frames of a video sequence where a moving object is totally occluded during several frames. *2nd Row*: Segmentation using initialization with previous segmentation. *3rd Row*: Initialization of the frames using the Ψ_{amp} representation of one of the detected composite-feature. *Bottom*: Segmentation using initialization with the composite feature

The key characteristic of composite-feature representation scheme is that integration is accomplished by clustering on frequency bands, not by point-wise region clustering. This fact yields a representation that intrinsically correlates information from different frames, in a way similar to techniques based on the Hough transform –but not limited to constant speed and direction. This property is responsible for the robustness to partial and total occlusions and large inter-frame displacements or deformations. Furthermore, the proposed representation scheme does not limit the possible motion patterns to predefined models, like translational or affine motion, thanks to the composition of elementary motion features. This is evident in example from section 4.4 –"silent" video sequence– where also local deformations of the target appear. Besides, energy filtering provides robustness to noise and aliasing.

On the other hand, composite features present a larger temporal-diffusion effect than, for example, the inter-frame difference. However, this effect is suitably corrected by the gradient masking. Naturally, other typical shortcomings associated to velocity tuned filters can appear. For instance, there may be problems with low contrast regions, since the representation model is related to the contrast of features. This is observed in the example of

Fig. 13. Three frames of a sequence showing two occluding motion patterns. 1st row: Input data. 2nd and 3rd rows: Inter-frame difference based segmentation, using a different initialization for each of the motion patterns. 4th and 5th rows: Ψeven of two of the obtained composite-features, corresponding to the two motion patters. 6th and 7th rows: Segmentations produced using composite-features from rows 4th and 5th respectively.

section 4.1, where the contour between tree and flower bed is poorly defined –see Fig. 5.

5. Conclusions

In this chapter, a new active model for the segmentation of motion patterns from video sequences has been presented. It employs a motion representation based on composite energy features. It consists on the clustering of elementary band-pass features, which can be considered velocity tuned features. Integration is accomplished by extending the notion of phase congruence to spatio-temporal signals. The active model uses this motion information both for image potential definition and initialization of the model in each frame of the sequence.

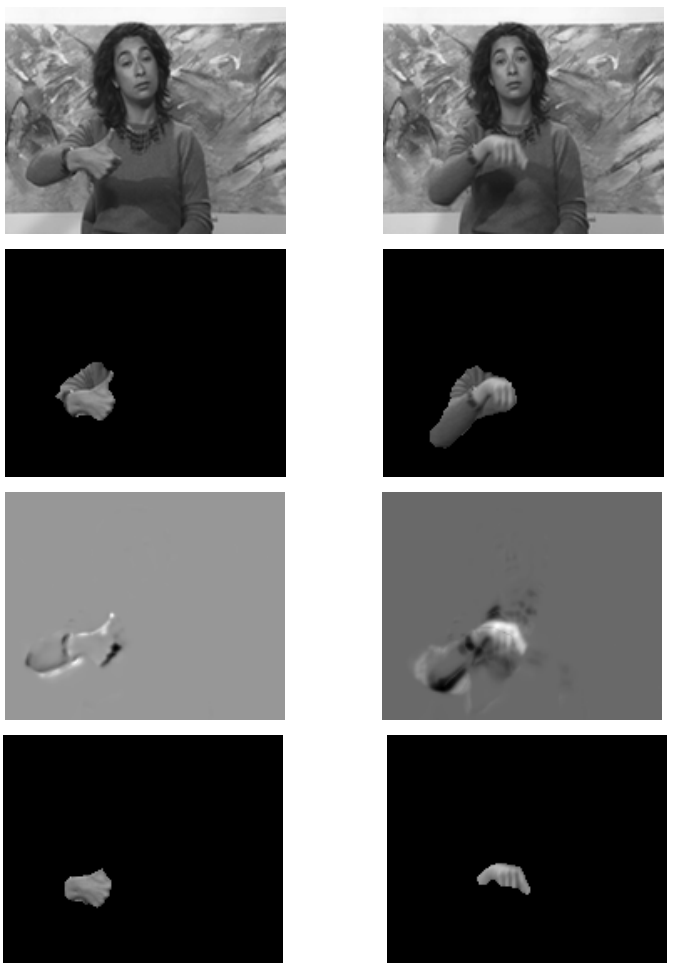


Fig. 14. Two frames of the "silent" video sequence: Top Row: Input data. 2nd Row: Segmentation using the active model based on the inter-frame difference. 3rd Row: Ψeven of the selected motion pattern. Bottom Row: Segmentation using the active model based on the composite-feature

The motion representation has proved to be able to isolate independent motion patterns from a video sequence. The integration criterion of spatio-temporal phase congruence gives place to a decomposition of the sequence into visually relevant motion patterns without the use of a priori knowledge.
The combination of geodesic active models and our motion representation yields a motion segmentation tool that presents good performance in many of the typical problematic situations, where previous approaches fail to properly segment and track, such as presence of noise and aliasing, partial and total occlusions, large inter-frame displacements or deformations, moving background and complex motion patterns. In the comparison with an alternative implementation, that employs typical solutions for initialization and definition of image potential, our method shows enhanced behavior.

6. Acknowledgements

This work has been financially supported by the Ministry of Education and Science of the Spanish Government through the research project TIN2006-08447.

7. References

Adelson, E.H. & Bergen, J.R. (1985). Spatiotemporal Energy Models for the Perception of Motion, J Opt Soc Am A, Vol. 2, No. 2, February 1985, pp. 284-299, ISSN: 1084-7529

Barron, J.L.; Fleet, D.J. & Beauchemin, S.S. (1994). Performance of Optical Flow Techniques, Int J Comput Vis, Vol. 12, No. 1, February 1994, pp. 43-77, ISSN: 0920-5691

Boykov, Y. & Huttenlocher, D.P. (2000). Adaptive Bayesian Recognition in Tracking Rigid Objects, Proceedings of the IEEE Comput Soc Conf Comput Vis Pattern Recogn (CVPR), Vol. II, pp. 697-704, Hilton Head Island (South Carolina), June 2000, IEEE Computer Society, Los Alamintos (CA), ISBN: 0-7695-0662-3

Caselles, V.; Kimmel, R. & Sapiro, G. (1997). Geodesic Active Contours, Int J Comput Vis, Vol. 22, No. 1, February 1997, pp. 61-79, ISSN: 0920-5691

Chamorro-Martínez, J.; Fdez-Valdivia, J.; García, J.A. & Martínez-Baena, J. (2003). A frequency Domain Approach for the Extraction of Motion Patterns, Proceedings of the IEEE Acoust Speech Signal Process, Vol. III, pp. 165-168, April 2003, IEEE Society, Los Alamitos (CA), ISBN: 0-7803-7663-3

Chang, M.M.; Tekalp, A.M. & Sezan, M.I. (1997). Simultaneous Motion Estimation and Segmentation, IEEE Trans Image Process, Vol. 6, No. 9, September 1997, pp. 1326-1333, ISSN: 1057-7149

Dosil, R. (2005) Data Driven Detection of Composite Feature Detectors for 3D Image Analysis. PhD Thesis, Universidade de Santiago de Compostela, ISBN: 84-9750-560-3, Santiago de Compostela (Spain) URL: http://www-gva.dec.usc.es/~rdosil/ficheiros/thesis_ dosil.pdf

Dosil, R.; Fdez-Vidal, X.R. & Pardo, X.M. (2005a). Dissimilarity Measures for Visual Pattern Partitioning, Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), Vol. II, pp. 287-294, Estoril (Portugal), June 2005, In: Lecture Notes in Computer Science, Vol. 3523, Marques, J. & Pérez de la Blanca, N. (Eds.), Springer-Verlag, Berlin Heidelberg, ISBN: 3-540-26154-0

Dosil, R.; Pardo, X.M. & Fdez-Vidal, X.R. (2005b). Decomposition of 3D Medical Images into Visual Patterns, IEEE Trans Biomed Eng, Vol. 52, No. 12, December 2005, pp. 2115-2118, ISSN: 0018-9294

Dosil, R. & Pardo, X.M. (2003). Generalized Ellipsoids and Anisotropic Filtering for Segmentation Improvement in 3D Medical Imaging, Image Vis Comput, Vol. 21, No. 4, April 2003, pp. 325-343, ISSN: 0262-8856

du Buf, J. (1994). Ramp Edges, Mach Bands and the Functional Significance of the Simple Cell Assembly, Biological Cybernetics, Vol. 70, No. 5, March 1994, pp. 449–461, ISSN: 0340-1200

Faas, F.G.A. & van Vliet, L.J. (2003). 3D-Orientation Space; Filters and Sampling, Proceedings of the 13th Scandinavian Conference in Image Analysis (SCIA), pp.36-42, Halmstad (Sweden), July 2003, In: Lecture Notes in Computer Science, Vol. 2749, Bigun, J. & Gustavsson, T. (Eds.), Springer-Verlag, Berlin Heidelberg, ISBN: 3-540-40601-8

Field, D.J. (1993). Scale–Invariance and self-similar "wavelet" Transforms: An Analysis of Natural Scenes and Mammalian Visual Systems, In: Wavelets, fractals and Fourier Transforms, pp. 151-193, Farge, M.; Hunt, J.C.R. & Vassilicos, J.C. (Eds.), Clarendon Press, Oxford, ISBN: 019853647X

Field, D.J. (1994). What is the Goal of Sensory Coding, Neural Computation, Vol. 6, No. 4, July 1994, pp. 559-601, ISSN: 0899-7667

Fleet, D. (1992). Measurement of Image Velocity, Kluwer Academic Publishers, ISBN: 0792391985, Massachusetts

Heeger, D.J. (1987). Model for the Extraction of Image Flow, J Opt Soc Am A, Vol. 4, No. 8, August 1987, pp. 1555-1471, ISSN: 1084-7529

Jain, A. & Dubes, R. Algorithms for Clustering Data, Prentice Hall, New Jersey, ISBN: 0-13-022278-X

Kass, M.; Witkin, A. & Terzopoulos, D. (1988). Snakes: Active Contour Models, Int J Comput Vis, Vol. 55, No. 4, January 1988, pp. 321-331, ISSN: 0920-5691

Kervrann, C. & Heitz, F. (1998). A Hierarchical Markov Modeling Approach for the Segmentation and Tracking of Deformable Shapes, Graph Model Image Process, Vol. 60, No. 3, May 1998, pp. 173-195, ISSN 1077-3169

Kovesi, P.D. (1996). Invariant Measures of Image Features from Phase Information, PhD. Thesis, The University or Western Australia, May 1996, URL: http://www.cs.uwa.edu.au/ pub/robvis/theses/PeterKovesi/

Mansouri, A.-J. & Konrad, J. (2003). Multiple Motion Segmentation with Level Sets, IEEE Trans Image Process, Vol. 12, No. 2, February 2003, pp. 201-220, ISSN: 1057-7149

Montoliu, R. & Pla, F. (2005). An Iterative Region-Growing Algorithm for Motion Segmentation and Estimation, Int J Intell Syst, Vol. 20, No. 5, May 2005, pp. 577-590, ISSN: 0884-8173

Morrone, M.C. & Owens, R.A. (1987). Feature Detection from Local Energy, Pattern Recognition Letters, Vol. 6, No. 5, December 1987, pp. 303-313, ISSN: 0167-8655

Nestares, O.; Miravet, C.; Santamaria, J. & Navarro, R. (2000). Automatic enhancement of noisy image sequences through localspatiotemporal spectrum analysis, Optical Engineering, Vol. 39, No. 6, June 2000, pp. 1457-1469, ISSN: 0091-3286

Nguyen, H.T. & Smeulders, A.W.M. (2004). Fast Occluded Object Tracking by a Robust Appearance Filter, IEEE Trans Pattern Anal Mach Intell, Vol. 26, No. 8, August 2004, pp. 1099-1104, ISSN: 0162-8828

Oppenheim, A. & Lim, J. (1981). The Importance of Phase in Signals, Proceedings of the IEEE, Vol. 69, No. 5, May 1981, pp. 529–541, ISSN: 0018-9219

Pal, N.R. & Biswas, J. (1996). Cluster Validation Using graph Theoretic Concepts, Pattern Recognition, Vol. 30, No. 6, June 1996, pp. 847-857, ISSN: 0031-3203

Paragios, N. & Deriche, R. (2000). Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects, IEEE Trans Pattern Anal Mach Intell, Vol. 22, No. 3, March 2000, pp. 266-279, ISSN: 1057-7149

Rodríguez-Sánchez, R.; García, J.A.; Fdez-Valdivia, J. & Fdez-Vidal, X.R. (1999). The RGFF Representational Model: A System for the Automatically Learned Partition of "Visual Patterns" in Digital Images, IEEE Trans Pattern Anal Mach Intell, Vol. 21, No. 10, October 1999, pp. 1044-1073, ISSN: 1057-7149

Ross, J.; Morrone, M.C. & Burr, D. (1989). The Conditions under which Mach Bands are Visible, Vision Research, Vol. 29, No. 6, 1989, pp. 699–715, ISSN: 0042-6989

Sato, K. & Aggarwal, J.K. (2004). Temporal Spatio-Temporal Transform and its Application to Tracking and Interaction, Comput Vis Image Understand, Vol. 96, No. 2, November 2004, pp. 100-128, ISSN: 1077-3142

Simoncelli, E.P. & Adelson, E.H. (1991). Computing Optical Flow Distributions using Spatio-Temporal Filters, MIT Media Lab. Vision and Modeling, Tech. Report No. 165, 1991, http://web.mit.edu/persci/people/adelson/pub_pdfs/simoncelli_comput.pdf

Stiller, C. & Konrad, J. (1999). Estimating Motion in Image Sequences: A Tutorial on Modeling and Computation of 2D Motion, IEEE Signal Processing Magazine. Vol. 16, No. 6, July 1999, pp. 71-91, ISSN: 1053-5888

Tsechpenakis, G.; Rapantzikos, K.; Tsapatsoulis, N. & Kollias, S. (2004). A Snake Model for Object Tracking in Natural Sequences, Signal Process Image Comm, Vol. 19, No. 3, March 2004, pp. 219-238, ISSN: 0923-5965

Venkatesh, S. & Owens, R. (1990). On the Classification of Image Features, Pattern Recognition Letters, Vol. 11, No. 5, May 1990, pp. 339-349, ISSN: 0167-8655

Wang, J.Y.A. & Adelson, E.H. (1994). Representing Moving Images with Layers, IEEE Trans Pattern Anal Mach Intell, Vol. 3, No. 5, September 1994, pp. 325-638, ISSN: 1057-7149

Watson, A.B. & Ahumada Jr., A.J. (1985). Model for Human Visual-Motion Sensing, J Opt Soc Am A, Vol. 2, No. 2, February 1985, pp. 322-342, ISSN: 1084-7529

Weickert, J. & Kühne, G. (2003). Fast Methods for Implicit Active Contour Models, In: Geometric Level Set Methods in Imaging, Vision and Graphics, pp. 43-58, Osher, S. & Paragios, N. (Eds.), Springer, ISBN: 0-387-95488-0, New York

2

Multimodal Range Image Segmentation

Michal Haindl & Pavel Žid
Institute of Information Theory and Automation, Academy of Sciences CR
Czech Republic

1. Introduction

Computer vision is an area of the computer science which aims to make computers to have some functions owned by human vision system. During the research on computer vision, we often think about "how human see". Obviously, when a human observer views a scene, the observer sees not the whole complex scene, but rather a collection of objects. A common experience of segmentation is the way that an image can resolve itself into a figure, typically the significant, important object, and a ground, the background on which the figure lies.
The capability of human beings to segment the complex scene into separated objects is so efficient that we regard it as a mystery. At meantime, it appeals the researchers of computer vision who want to let computer have the same capability. So the task, Image Segmentation, is presented.
The chapter describes new achievements in the area of multimodal range and intensity image unsupervised segmentation. This chapter is organized as follows. Range sensors are described in section 2 followed by the current state of art survey in section 3. Sections 4 to 6 describe our fast range image segmentation method for scenes comprising general faced objects. This range segmentation method is based on a recursive adaptive probabilistic detection of step discontinuities (sections 4 and 5) which are present at object face borders in mutually registered range and intensity data. Detected face outlines guides the subsequent region growing step in section 6 where the neighbouring face curves are grouped together. Region growing based on curve segments instead of pixels like in the classical approaches considerably speed up the algorithm. The exploitation of multimodal data significantly improves the segmentation quality. The evaluation methodology a range segmentation benchmarks are described in section 7. Following sections show our experimental results of the proposed model (section 8), discuss its properties and conclude (section 9) the chapter.

1.1 Image Segmentation

There is no single standard approach to segmentation. The definition of the goal of segmentation varies according to the type of the data and the application type. Different assumptions about the nature of the images being analyzed lead to use of different algorithms. One possible image segmentation definition is: "Image Segmentation is a process of partitioning the image into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous" (Pal & Pal, 1993).

The segmentation process is perhaps the most important step in image analysis since its performance directly affects the performance of the subsequent processing steps in image analysis and it significantly determines the resulting image interpretation. Despite its utmost importance, segmentation still remains as an unsolved problem in the general sense as it lacks a general mathematical theory. The two main difficulties of the segmentation problem are its underconstrained nature and the lack of definition of the "correct" segmentation. Perhaps as a consequence of these shortcomings, a plethora of segmentation algorithms has been proposed in the literature. These algorithms range from simple ad hoc schemes to more sophisticated ones using object and image models.
The area of segmentation algorithms typically suffers with the lack of benchmarking results and methodologies. With few rare exceptions in specific narrow applications single segmentation algorithm cannot be ranked and potential user has to experimentally validate several segmentation algorithms for his particular application.

1.2 Range Image Segmentation

Range images store, instead of brightness or colour information, the depth at which the ray associated with each pixel first intersects the object observed by a camera. In a sense, a range image is exactly the desired output of stereo, motion, or other shape-from vision modules. It provides geometric information about the object independent of the position, direction, and intensity of light sources illuminating the scene, or of the reflectance properties of that object.
Range image segmentation has been an instrument of computer vision research for nearly 30 years. Over that period several partial results have found its way into many industrial applications such as geometric inspection, reverse engineering or autonomous navigation systems. However similarly as in the spectral image segmentation area the range image segmentation problem is still far from being satisfactory solved.

2. Range Sensors

Range sensors can be grouped into the passive and active once. A rich variety of passive stereo vision techniques produce three-dimensional information. Stereo vision involves two processes: the binocular fusion of features observed by the two cameras and the reconstruction of their three dimensional preimage. An alternative to classical stereo is the photometric stereo (Horn, 1986). Photometric stereo is a monocular 3-D shape recovery method assuming single illumination point at infinity, Lambertian opaque surface and known camera parameters, that relies on a few images (minimally 3) of the same scene taken under different lighting conditions. If this before mentioned knowledge is not available, i.e., uncalibrated stereo, more intensity images are necessary. There are usually two processing steps: First, the direction of the normal to the surface is estimated at each visible point. The set of normal directions, also known as the needle diagram, is then used to determine the 3-D surface itself. At the limit, shape from shading requires a single image, but then solving for the normal direction or 3-D location of any point requires integration of data from all over the image.
Active sensing techniques promise to simplify many tasks and problems in machine vision. Active range sensing operates by illuminating a portion of the surface under controlled conditions and extracting a quantity from the reflected light (angle of return in

triangulation, time/phase/frequency delay in time of flight sensors) in order to determine the position of the illuminated surface area. This position is normally expressed in the form of a single 3-D point.

An active range sensor - a range camera - is a device which can acquire a raster (two-dimensional grid, or image) of depth measurements, as measured from a plane (orthographic) or single point (perspective) on the camera (Forsyth & Ponce, 2003). In an intensity image, the greyscale or colour of imaged points is recorded, but the depths of the points imaged are ambiguous. In a range image, the distances to points imaged are recorded over a quantized range. For display purposes, the distances are often coded in greyscale, usually that the darker a pixel is, the closer it is to the camera.

Intensity Image Range Image

Fig. 1. Example of registered intensity and range image

2.1. Triangulation Based (Structured Light) Range Sensors

Triangulation based range finders date back to the early seventies. They function along the same principles as passive stereo vision systems, one of the cameras being replaced by a source of controlled illumination (structured light). For example, a laser and a pair of rotating mirrors may be used to sequentially scan a surface. In this case, as in conventional stereo, the position of the bright spot where the laser beam strikes the surface of interest is found as the intersection of the beam with the projection ray joining the spot to its image. Contrary to the stereo case, however, the laser spot can normally be identified without difficulty since it is in general much brighter than the other scene points (in particular when a filter tuned to the laser wavelength is inserted in front of the camera), altogether avoiding the correspondence problem.

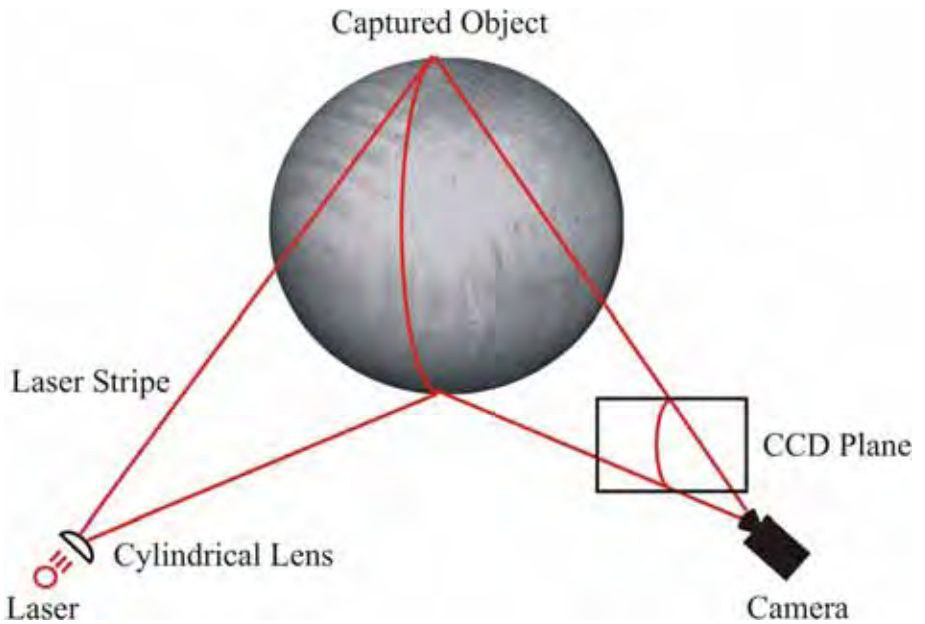


Fig. 2. Optical triangulation using laser beam for illumination.

Alternatively, the laser beam can be transformed by a cylindrical lens into a plane of light (Fig. 2.). This simplifies the mechanical design of the range finder since it only requires one rotating mirror. More importantly, perhaps, it shortens the time required to acquire a range image since a laser stripe, the equivalent of a whole image column, can be acquired at each frame.

A structured light scanner uses two optical paths, one for a CCD sensor and one for some form of projected light, and computes depth via triangulation. ABW GmbH and K2T Inc. are two companies which produce commercially available structured light scanners. Both of these cameras use multiple images of striped light patterns to determine depth. two example structured light patterns used by the K2T GRF-2 range camera are shown in Fig. 3.

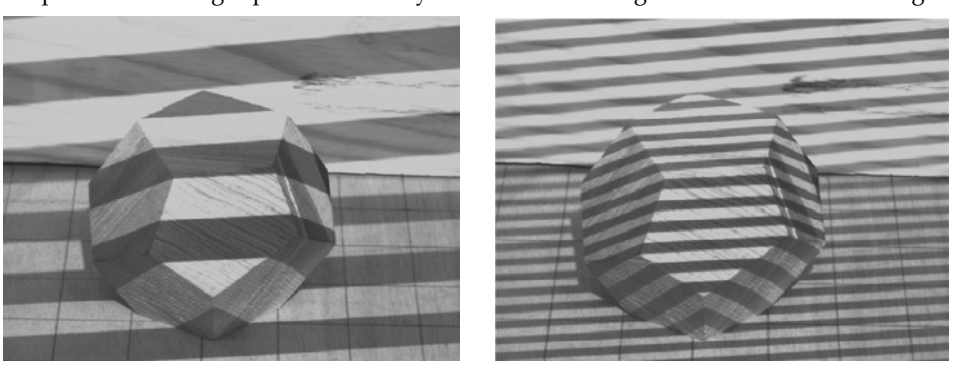


Fig. 3. Example images of two of the eight structured light patterns used by the K2T GRF-2 range camera.

Variants of these techniques include using multiple cameras to improve measurement accuracy and exploiting (possibly time coded) two dimensional light patterns to improve data acquisition speed. The main drawbacks of the active triangulation technology are relatively low acquisition speed and missing data at parts of the scene visible to the CCD sensor and not visible to the light projector. The resulting pixels in the range image, called shadow pixels, do not contain valid range measurements. Next difficulties arise from missing or erroneous data due to specularities. It is actually common to all active ranging techniques: a purely specular surface will not reflect any light in the direction of the camera unless it happens to lie in the corresponding mirror direction. Worse, the reflected beam may induce secondary reflections giving false depth measurements.

2.2. Time of Flight Range Sensors

The second main approach to active ranging involves a signal transmitter, a receiver, and electronics for measuring the time of flight of the signal during its round trip from the range sensor to the surface of interest (Dubrawski & Sawwa, 1996). This is the principle used in the ultrasound domain by the Polaroid range finder, commonly used in autofocus cameras from that brand and in mobile robots, despite the fact that the ultrasound wavelength band is particularly susceptible to false targets due to specular reflections. Time of flight laser range finders are normally equipped with a scanning mechanism, and the transmitter and receiver are often coaxial, eliminating the problem of missing data common in triangulation approaches. There are three main classes of time of flight laser range sensors:

- pulse time delay RS
  Pulse time delay sensor emits very brief, very intense pulses of light. The amount of time the pulse takes to reach the target and return is measured and converted to a distance measurement. The accuracy of these sensors is typically limited by the accuracy with which the time interval can be measured, and the rise time of the laser pulse.

- AM phase-shift RS
  AM phase-shift range finders measure the phase difference between the beam emitted by an amplitude-modulated laser and the reflected beam (see Fig. 4.), a quantity proportional to the time of flight.

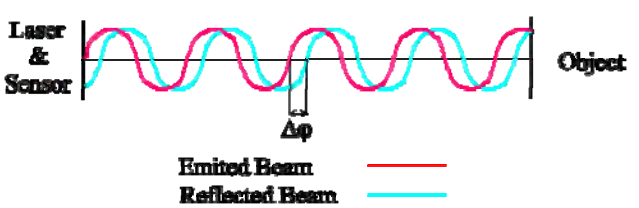


Fig. 4. Illustration of AM phase-shift range sensor measurement.

Measured distance r can be expressed as:

$$r = \Delta\varphi * \frac{\lambda_m}{4\pi} \quad (1)$$

where $\Delta\varphi$ is the phase difference between emitted and reflected beam and λ_m is the wave-length of modulated function. Due to periodical nature of modulated function the measurement is possible only in an ambiguity interval $r_a=\lambda_m/2$.

- FM beat RS
  FM beat sensors measure the frequency shift (or beat frequency) between a frequency-modulated laser beam and its reflection (see Fig. 5.), another quantity proportional to the round trip flight time.

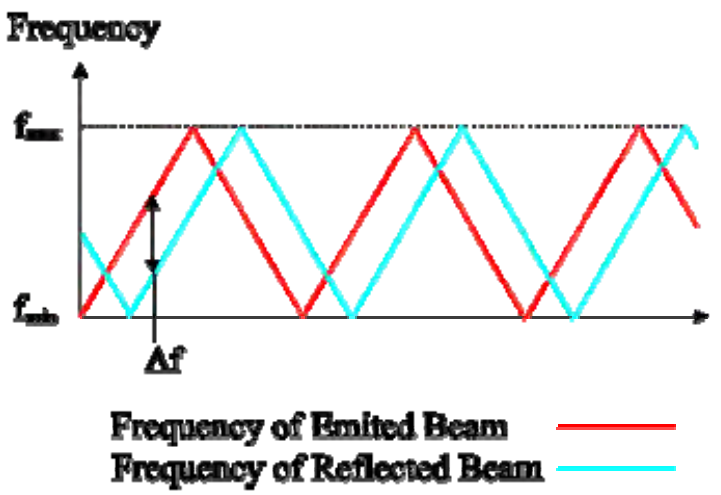


Fig. 5. Illustration of FM beat range sensor measurement.

Measured distance r can be expressed as:

$$r = c * \frac{f_b}{f_m * \Delta f}, f_b = |f_e - f_r| \tag{2}$$

where c is speed of light, f_m mean modulation frequency, Δf the difference between highest and lowest frequency in modulated run, f_e emitted beam frequency and f_r reflected beam frequency.

Time of flight range finders face the same problems as any other active sensors when imaging specular surfaces. They can be relatively slow due to long integration time at the receiver end. The speed of pulse time delay sensors is also limited by the minimum resolvable interval between two pulses. Compared to triangulation based systems, time of flight sensors have the advantage of offering a greater operating range (up to tens of meters), which is very valuable in outdoor robotic navigation tasks.

3. State of the Art

There are many spectral image segmentation algorithms published in computer vision literature and a number of good survey articles (Besl & Jain, 1985), (Sinha & Jain, 1994) is available but substantially less range image segmentation algorithms were published. Mutual comparison of their segmentation quality and performance is very difficult because of lack of sound experimental evaluation results. A rare exception in the area of planar face objects range segmentation is published in (Hoover et al., 1996a) together with experimental data available on their Internet server. Because this evaluation methodology became de facto standard in the area of planar range segmentation algorithms comparison, these data and results are used also for our algorithm evaluation.

3.1. Range Segmentation Principles

There are several methods for segmenting an image into regions, which, subsequently, can be analyzed, based on their shapes, sizes, relative positions, and other characteristics, and there are several possible categorizations of segmentation techniques. The most common categorization accepted also in this chapter sorts segmentation methods into three or four different philosophical perspectives. We name them after the terminology "pixel based segmentation, edge based segmentation, region based segmentation and hybrid segmentation".

3.1.1. Pixel Based Segmentation

Pixel based segmentation is the most local method to address the task of image segmentation. Every pixel has to be sorted to some certain class. At last, the pixels belonging to the same class which are contiguous will constitute one segmented region.

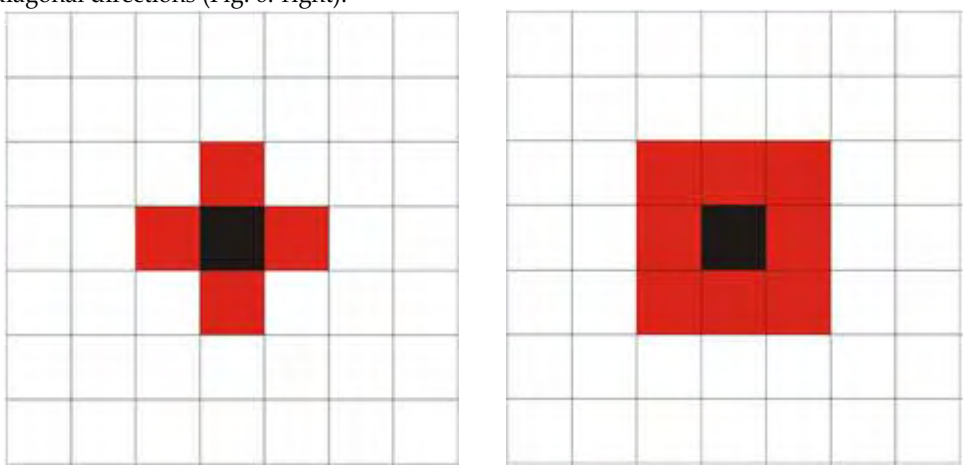
3.1.2. Edge Based Segmentation

Edge based segmentation is more global than pixel based segmentation, but it is more local when compared to the area based segmentation. So it is on the "middle level".
Edge based segmentation make use of the clue that "how human see" for the second time, because a person always has the principle that there is an edge in some certain sense between two segmentable objects (Zhang & Zhao, 1995), (Palmer et al., 1996). An edge pixel is characterized by a vector that shows a particular position, size and direction of discontinuity. Sometimes only the size is determined. The "direction" of the edge is perpendicular to the "direction" of the rim of the object.

3.1.3. Region Based Segmentation

Among the four surveyed approaches, region based segmentation is the most global method. This approach groups pixels into regions based upon two criteria: proximity and homogeneity. Most region based methods produce these groupings either by splitting the image, or its regions, into smaller regions (Lee et al., 1998), merging small regions into larger ones (Hoover et al., 1996a), (Besl & Jain, 1988), or splitting and merging until the criteria are maximally satisfied (Haralick & Shapiro, 1985), (Chang, 1994), (Hijjatoleslami & Kittler, 1998). In two-dimensional region growing, regions that have pixels that are "four-connected" (Fig. 6.-left), that is, directly neighboring each other in any of the four horizontal and vertical directions, are considered to be in proximity to one another. Other region growing

algorithms extend these criteria to "eight-connected" pixels by also including the four diagonal directions (Fig. 6.-right).



Four-Connected Neighbours & Eight-Connected Neighbours

Fig. 6. Neighbourhood examples.

The second criterion, homogeneity, is satisfied by an implementation specific function that quantifies the similarity between regions. This function may be based on a comparison of any single or combination of available region statistics .

3.1.4. Hybrid Segmentation

Hybrid techniques are trying to combine advantages of two or more previously described segmentation methods. They are expected to provide more accurate segmentation of images. Pavlidis et al. (Pavlidis & Liow, 1990) describe a method to combine segments obtained by using a region-growing approach, where the edges between regions are eliminated or modified based on contrast, gradient and shape of the boundary. Haddon and Boyce (Haddon & Boyce, 1990) generate regions by partitioning the image co-occurrence matrix and then refining them by relaxation using the edge information. Chu and Aggarwal (Chu & Aggarwal, 1993) present an optimization method to integrate segmentation and edge maps obtained from several channels, including visible, infrared, etc., where user specified weights and arbitrary mixing of region and edge maps are allowed. The method presented in this chapter can be classified as hybrid technique, because we use edge detection as the first step of the algorithm and segment based region growing as the second step.

3.2. Planar Face Segmentation Algorithms

A specially simplified range image segmentation task occurs when we may assume some additional prior information about the segmented scene. One of the most frequent assumptions is planarity of range scene objects faces. A planar surface can be characterized as a connected set of 3D surface points at which the two principal curvatures (alternatively, the Gaussian and mean curvatures) are zero. It gives the chance to simplify the model of the region and thus simplify the whole segmentation process.

3.2.1. The USF Range Segmentation Algorithm

This segmenter (Hoover et al., 1996a) works by computing a planar fit for each pixel and then growing regions whose pixels have similar plane equations. The pixel with the smallest interiorness measure is chosen as a seed point for region growing. The border of the region is recursively grown until no pixels join, at which time a new region is started using the next best available seed pixel (based on interiorness measure). Pixels are only allowed to participate in this process once. If a region's final size is below a threshold, then the region is discarded. This algorithm shows good segmentation results over test sets especially in under-segmentation measure and has only 5 parameters, but it is discriminated by its computational speed.

3.2.2. The WSU Range Segmentation Algorithm

The WSU range image segmentation method (Hoffman & Jain, 1987), (Flynn & Jain, 1991) is not optimized for polyhedral objects but can accommodate natural quadric surfaces as well. It was modified to accept only first-order surface fits, but no other special steps were taken to exploit the planar nature of the scenes. Prior to any processing, the range points are uniformly scaled to fit within a 5×5 cube. Than jump edge pixels are identified. Surface normals are estimated at each range pixel with no jump edges in a neighbourhood. The six-dimensional image is formed by concatenating the estimated surface normals to their corresponding pixels.

These 6-vectors are fed to a squared-error clustering algorithm, which finds groupings in the data set based on similarity between the data points. Since these points reflect both position and orientation, the tendency is to produce clustering consisting of connected image subsets, with pixels in each cluster having similar orientation. The selected clustering is converted into an image segmentation by assigning each range pixel to the closest cluster centre in the clustering. A further merging step joins segments if they are adjacent and have similar parameters. This algorithm achieved the worst result in all segmentation quality measures among the four compared algorithms. The next drawback of this algorithm is high number of tuneable parameters and relatively high computational times.

3.2.3. The UBP range segmentation algorithm

This segmenter (Jiang & Bunke, 1994) is based on the fact that, in the ideal case, the points on a scan line that belong to a planar surface form a straight 3D line segment. On the other hand, all points on a straight 3D line segment surely belong to the same planar surface. Therefore, they first divide each scan line into straight line segments and subsequently perform a region growing process using the set of line segments instead of the individual pixels.

A potential seed region for region growing is a triple of line segments on three neighbouring scan lines. The candidate with the largest total line segment length is chosen as the optimal seed region. In the subsequent region growing process, a line segment is added to the region if the perpendicular distance between its two end points and the plane equation of the region is within a dynamic threshold. This process is repeated until no more line segments can be added, at which time a new region is started using the next best available seed region. If a region's final size is below a threshold, then the region is discarded. This algorithm is probably the best one of all methods surveyed in (Hoover et al., 1996a). It achieves high number of correctly detected region over both test sets, low number of

undersegmented regions, but it oversegments some regions. The number of parameters is relatively high. This is the fastest algorithm surveyed in (Hoover et al., 1996a).

3.2.4. The UE Range Segmentation Algorithm

The UE segmentation algorithm (Hoover et al., 1996a) is a region growing type of algorithm along the lines of the USF segmenter. Initial surface normals are calculated at each pixel using a plane fit to the data. Depth and normal discontinuity detection is performed using simple thresholds between neighbouring pixels. Gaussian (H) and mean (K) curvature are estimated at each pixel. Pixels can be labelled as belonging to particular surface types based on the combined signs of the (H, K) values. Once each pixel is labelled properly with the signs of H and K, any eight--connected pixels of similar labelling are grouped to form initial regions. This segmentation map is then morphologically dilated and eroded in a specifiable manner to fill small Unknown areas, remove small regions, and separate thinly connected components. For each region in the initial segmentation above a minimal size a least squares surface fitting is performed. Then each region in turn is grown. The UE segmenter obtains slightly better measures of correct detection than does the UBP segmenter but the difference in processing speeds is noteworthy. The main drawback of this algorithm is high number of its parameters. Nearly a dozen values should be adjusted before the segmentation.

3.2.5. Robust Adaptive Segmentation (ALKS) Algorithm

The authors of this method (Lee et al., 1998) proposed an image segmentation technique using the robust, adaptive least *k*th order squares (ALKS) estimator which minimizes the *k*th order statistics of the squared of residuals. The optimal value of *k* is determined from the data, and the procedure detects the homogeneous surface patch representing the relative majority of the pixels. The method defines the region to be processed as the largest connected component of unlabelled pixels. Applies the ALKS procedure to the selected region and discriminates the inliers, labels the largest connected component of inliers as the delineated homogeneous patch and refines the model parameter estimates by a least--squares fit to the inliers. Than it repeats these steps until the size of the largest connected component is less than a threshold. To eliminate the isolated outliers surrounded by inliers an unlabelled pixel is allocated to the class of the majority of its labelled four-connected neighbours. For this method we can only compare results published in the article (Lee et al., 1998). These results show that the ALKS method tends to undersegment some faces. We have no information about computational times for this method.

3.2.6. Segmentation through the integration of different strategies (PPU)

The authors of this paper (Bock & Guerra, 2001) consider the problem of segmenting range images into planar regions.

The approach they present combines different strategies for grouping image elements to estimate the parameters of the planes that best represent the range data. The strategies differ not only in the way candidate planes are hypothesized but also in the objective function used to select the best plane among the potential candidates. The method they consider integrates in an effective way different strategies for plane recovery. There are mainly three procedures all based on random sampling. Three main procedures are invoked sequentially in a given order for new plane detection in iteration, the next one executed only when the previous ones do not detect a significant plane. Once the parameters of the best plane are

found at a given iteration, the plane is expanded over the entire image and all the fragments of the same surface in the image are labelled as belonging to the same plane. This method achieved the worst result in the comparison.

3.2.7. The OU segmentation algorithm

The segmentation algorithm OU (Jiang et al., 2000) is based on the analysis of intersection of the scene by arbitrary planes. At first, the range image is divided into two hemi-spaces by an imaginary plane, and then it binarizes each pixel on the range image. On that image, the issue of plane detection is turned into edge detection on the binary image. Authors applied Hough transform for derivative image to do it, because test images contain much noise. The topological information in the binary image can also be used for end-point determination and grouping detected line segments. The imaginary plane is translated step by step and the set of line segments is obtained. To accelerate this algorithm, the voting space for Hough transform is limited using information of prior line detection. Then planes are made by grouping line segments: If two lines share a plane, the lines are parallel to each other and have a close distance. Therefore all line segments are classified into several groups using gradient, distance and arrangement of end--points of the lines. The topological cue of binary image can be also exploited. Finally the range image is filled by polygons which are generated from two neighbouring lines. Against the case of fragmentation of polygons the normal vectors of each planar surface are evaluated and unified if the difference of normal vectors of neighbouring planar surfaces is small enough.

3.2.8. The UA segmentation algorithm

The segmentation algorithm UA (Jiang et al., 2000 performs a fast hierarchical processing in a multiresolution pyramid, or quadtree, based on (Loke & duBuf, 1998) and (Wilson & Spann, 1988). It must be noted that the method has only very recently been adapted for range images. Its current disadvantage is the information loss which is caused by linearly combining the components of the surface normal vector. A quadtree of L levels is built with at the base (level 0) the original range image on a regular grid. Low-resolution depth data at each higher level are determined by a low-pass filtering of depths at the lower level in no overlapping blocks of size 2×2. This reduces the noise, allowing the estimation of accurate normal vectors at the highest level $L - 1$. New filter technique is applied in order to increase the homogeneity of the data and to reduce the noise. At level $L - 1$, data clusters are determined using the local-centroid algorithm (Wilson & Spann, 1988). Thereafter, the segmentation at level $L - 1$ is obtained by setting each pixel to the label of its nearest cluster. Starting at level $L - 1$ and ending at level 1, the segmentation at each lower level is obtained by refining the boundary. At level 0, a component labelling is performed, because the segmentation may contain regions which have the same label, but which are not spatially connected. All regions smaller than a minimum region size are disregarded.

3.3. Non-Planar Face Segmentation Algorithms

If the measured range scene contains general objects and we cannot use the simplifying planar face assumption, the segmentation task is substantially more difficult. There is very few non-planar range segmentation algorithms published and no benchmarking methodology generally accepted.

3.3.1. The UBC Range Segmentation Algorithm

The UBC segmenter (Jiang & Bunke, 1998) consists of two parts: edge detection and grouping of edge points into closed regions. It makes use of the fact that each scan line (row, column or diagonal) of a range image is a curve in 3-D space. Therefore, it partitions each scan line into a set of curve segments by means of a splitting method. All the splitting points represent potential edges. The jump and crease edge strength of the edge candidates are evaluated by analytically computing the height difference and the angle between two adjacent curve segments, respectively. Each pixel can be assigned up to four edge strength values of each type (jump and crease) from the four scan lines passing through the pixel. These edge strength values are combined by taking the maximum to define the overall edge strength of each type. The grouping process is based on a hypochapter generation and verification approach. From the edge map, regions can be found by a component labelling. Due to the inevitable gaps in the edge chains, however, this initial grouping usually results in under-segmentation. To recognize the correctly segmented and under-segmented regions, a region test is performed for each region of the initial segmentation. If the region test is successful, the corresponding region is registered. Otherwise, the edge points within the region are dilated once, potentially closing the gaps. Then, hypochapter generation (component labelling) and verification (region test) are carried out for the region. This process is recursively done until the generated regions have been successfully verified or they are no longer considered because of too small a region size. The results suggest that the UBC segmentation algorithm substantially out-performs the BJ algorithm. However, these results should be interpreted carefully. The UBC segmenter has a fundamental limitation. The edge detection method described above is able to detect jump and crease edges but not smooth edges (discontinuities only in curvature). This seems to be true for all edge detectors reported in the literature.

3.3.2. Variable Order Surface Fitting (BJ) Algorithm

Besl and Jain developed a segmentation algorithm (Besl & Jain, 1988) which uses signs of surface curvature to obtain a coarse segmentation and iteratively refines it by fitting bivariate polynomials to the surfaces. The algorithm begins by estimating the mean and Gaussian surface curvature at every pixel and uses the signs of the curvatures to classify each pixel as belonging to one of eight surface types. The resultant coarse segmentation is enhanced by an iterative region growing procedure. For every coarse region, a subregion of a size at or above a threshold is selected to be a seed region. Low order bivariate polynomials are used to produce an estimated surface fit to the seed region. Next, all pixels in all regions of the image that are currently outside the seed region are tested for possible inclusion into the current region. The largest connected region which is composed of pixels in the seed region and pixels that passes the compatibility tests is chosen as the new seed region. Expansion continues until either there is almost zero change in region size since the last iteration, or when the surface fitting error becomes larger than a threshold. Finally, fit error is calculated, and if it falls below a threshold the region is accepted. If not, the region is rejected and the seed region that produced it is marked off so that it may not be used again.

3.3.3. Industrial Research Limited Simple Surface Segmentation (IRLBC and IRLRG)

The authors of this paper (McIvor et al., 1997) described a method for the recognition of simple curved surface patches from dense 3--D range data, such as that provided by a

structured light system. Patches from planes, spheres, cylinders, and ruled surfaces are considered. The approach can be summarised as follows. The first step is to estimate the local surface geometry (the principal quadric) at each visible surface point. Then points at which the signs of the Gaussian and mean curvatures are inconsistent with those of a particular surface type are rejected from further consideration. Each remaining point is mapped to a point in the parameter space of the surface type. By using an unsupervised Bayesian classification (IRLBC method) or region growing algorithm (IRLRG method), the clusters in parameter space that correspond to surface patches are identified, and the parameters of that surface can be determined.

4. Multimodal Range Image Segmentation

4.1. Face Outline Detection

We assume mutually registered range $(y_{t,r})$ and intensity $(y_{t,i})$ data $Y_t = [y_{t,r}, y_{t,i}]^T$ of the scene to be modeled in the unshaded part (scene part with valid range measurements) by an adaptive causal simultaneous autoregressive model (SAR) in some chosen direction:

$$Y_t = \gamma Z_t + \varepsilon_t \tag{3}$$

Where $\gamma = [A_1, \ldots, A_\eta]$ is the $2 \times \eta$ unknown parameter matrix and η = card I_t. We denote the $2\eta \times 1$ data vector $Z_t = [Y_{t-i}^T : \forall i \in I_t]$ with a multi-index $t = (m,n)$; m, n are the row and column indices, respectively. The multiindex changes according to chosen direction of movement on the image plane e.g. t-1=(m,n-1), t-2=(m,n-2),…,I_t is some contextual causal or unilateral neighbour index shift set. The white noise vector ε_t has zero mean and constant but unknown covariance matrix Ω. We further assume uncorrelated noise vector components, i.e., $E\{\varepsilon_{t,r}\ \varepsilon_{t,i}\} = 0\ \forall t$ and the probability density of ε_t to have the normal distribution independent of previous data and being the same for every time t. The task consists in finding the conditional prediction density $p(Y_t \mid Y^{(t-1)})$ given the known process history $Y^{(t-1)} = \{Y_{t-1}, Y_{t-2}, \ldots, Y_1, Z_t, Z_{t-1}, \ldots, Z_1\}$ and taking its conditional mean estimation $\hat{Y}$ for the predicted data. If the prediction error is greater than an adaptive threshold the algorithm assumes an object face edge pixel.

Assuming normality of the white noise component ε_t, conditional independence between pixels and the normal-Wishart parameter prior, we have shown (Haindl & Šimberová, 1992) that the conditional mean value is:

$$\hat{Y}_t = E\left[Y_t \middle| Y^{(t-1)}\right] = \hat{\gamma}_{t-1} Z_t \tag{4}$$

The following notation is used in (4):

$$\hat{\gamma}_{t-1} = V_{zz(t-1)}^{-1} V_{zy(t-1)}$$

$$V_{t-1} = \hat{V}_{t-1} + V_0$$

$$\hat{V}_{t-1} = \begin{pmatrix} \hat{V}_{yy(t-1)} & \hat{V}_{zy(t-1)}^T \\ \hat{V}_{zy(t-1)} & \hat{V}_{zz(t-1)} \end{pmatrix}$$

$$\hat{V}_{xw(t-1)} = \alpha \hat{V}_{xw(t-2)} + X_{t-1} W_{t-1}^T$$

and V_0 is a positive definite matrix. We assume slowly changing parameters, consequently these equations were modified using a constant exponential "forgetting factor" α to allow parameter adaptation. It is easy to check (see (Haindl & Šimberová, 1992)) also the validity of the following recursive parameter estimator:

$$\hat{\gamma}_t = \hat{\gamma}_{t-1} + (\alpha^2 + Z_t^T V_{zz(t-1)}^{-1} Z_t)^{-1} V_{zz(t-1)}^{-1} Z_t (Y_t - \hat{\gamma}_{t-1}^T Z_t)^T \tag{5}$$

Let us define the following three conditions with adaptive thresholds (7),(8):

$$Y_t \notin S \tag{6}$$

$$\left| \hat{y}_{t,r} - y_{t,r} \right| > \frac{2.5}{l} \sum_{j=1}^{l} \left| \hat{y}_{t-j,r} - y_{t-j,r} \right| \tag{7}$$

$$\left| \hat{y}_{t,i} - y_{t,i} \right| > \frac{2.5}{l} \sum_{j=1}^{l} \left| \hat{y}_{t-j,i} - y_{t-j,i} \right| \tag{8}$$

where S is the shaded (unmeasured part) of the range image. The pixel t is classified as an object edge pixel (a detected step discontinuity pixel) iff either the conditions (6),(7) or (6), non (7),(8) hold. Both adaptive thresholds are proportional to the local mean prediction error estimation.

4.2. Competing Models

Let us assume two SAR models (3) M_1 and M_2 with the same number of unknown parameters ($\eta_1 = \eta_2 = \eta$) and an identical neighbour index shift sets $I_{t'}$. They differ only in their forgetting factors $\alpha_1 > \alpha_2$. The model M_1, $\alpha_1 \approx 1$ represents homogeneous image areas while the second model better represents new information coming from crossing some face borders because it allows quicker adaptation to this new information. The optimal decision rule for minimizing the average probability of decision error chooses the maximum a posterior probability model, i.e. a model whose conditional probability given the past data is the highest one. Predictors used in the presented algorithm can be therefore completed as:

$$\hat{Y}_t = \begin{cases} \hat{\gamma}_{1,t-1} Z_t, & if\ (10)\ holds \\ \hat{\gamma}_{2,t-1} Z_t, & otherwise \end{cases} \tag{9}$$

where Z_t is a data vector identical to both models and

$$p(M_1 | Y^{(t-1)}) > p(M_2 | Y^{(t-1)}) \tag{10}$$

The analytical solution has the following form (Haindl & Šimberová, 1992):

$$p(M_i \mid Y^{(t-1)}) = k\Gamma\left(\frac{\psi(t-1)-\eta+2}{2}\right)\left|V_{i,zz(t-1)}\right|^{-\frac{1}{2}}\lambda_{i,t-1}^{-\frac{\psi(t-1)-\eta+2}{2}} \tag{11}$$

where k is a common constant. All statistics related to a model M_1 (6), (11) are computed using the exponential forgetting constant α_1 while symmetrical statistics of the model M_2 are computed using the second constant α_2.
The solution of (11) uses the following notations:

$$\psi(t) = \alpha^2\psi(t-1)+1 \tag{12}$$

$$\lambda_{t-1} = V_{yy(t-1)} - V_{zy(t-1)}^T V_{zz(t-1)}^{-1} V_{zy(t-1)} \tag{13}$$

The determinant $|V_{zz(t)}|$ as well as λ_t can be evaluated recursively (Haindl & Šimberová, 1992):

$$\left|V_{zz(t)}\right| = \left|V_{zz(t-1)}\right|\alpha_i^{2\eta}\left(1+Z_t^T V_{zz(t-1)}^{-1} Z_t\right)$$

$$\lambda_t = \lambda_{t-1}\alpha_i^2\left(1+\left(y_t - \hat{P}_{t-1}^T Z_t\right)^T \lambda_{t-1}^{-1}\left(y_t \hat{P}_{t-1}^T Z_t\right)\left(\alpha_i^2 + Z_t^T V_{zz(t-1)}^{-1} Z_t\right)^{-1}\right)$$

For numerical realization of the predictor (9) see discussion in (Haindl & Šimberová, 1992).

4.3. Face Detection

The previous step of the algorithm detects correct face outlines however some pixels on these edges can be either missing or edges can be incomplete. This missing information is estimated in a curve segment-based region growing process. Curves to be grown do not need to be of maximal length through the corresponding object face. Any curve segments can serve as initial estimation however longer curve segments speed up the region growing step. The only restriction imposed on them is that they are not allowed to cross face borders detected in the previous step of the algorithm. These curve segments can be generated in two mutually perpendicular directions but our current implementation uses only one of these directions.
A curve is represented using the cubic spline model:

$$Y_{r_i}^{s_i} = a_{s_i}\left(r_i - s_i\right)^3 + b_{s_i}\left(r_i - s_i\right)^2 + c_{s_i}\left(r_i - s_i\right) + d_{s_i}$$

for the interval $r_i \in \langle s_i; s_{i+\Delta}\rangle$, $\Delta=1$ for the single-scale version of the algorithm, $i=1$ for columnwise or $i=2$ for rowwise direction, and $r_j=s_j$ for $j \neq i$. Splines representing segments in a chosen direction are computed and parameter space Ξ is created over the image lattice. Two curve segments s_i,t_i in the same column (row) are merged together iff:

1. They have similar slope

$$\frac{\delta Y_{r_i}^{s_i}}{\delta r_i} \approx \frac{\delta Y_{r_i}^{t_i}}{\delta r_i}$$

The slope for identical steps Δ=1 is dependent on first three spline parameters, i.e.

$$\frac{\delta Y_{r_i}^{s_i}}{\delta r_i} = 3a_{s_i} + 2b_{s_i} + c_{s_i}$$

2. They have similar curvature

$$\frac{\delta^2 Y_{r_i}^{s_i}}{\delta r_i^2} \approx \frac{\delta^2 Y_{r_t}^{t_i}}{\delta r_i^2}$$

The curvature we approximate with the b_{Si} parameter

$$b_{s_i} = \frac{1}{2}\frac{\delta^2 Y_{r_i}^{s_i}}{\delta r_i^2}$$

Both conditions are satisfied if we require similar spline parameters a_{Si}, b_{Si}, c_{Si} for segments to be merged. The similarity measure chosen is the squared Euclidean distance. Similarly two parallel neighbouring curve segments r, ŕ r ∈ <s_i; s_i + 1> × r_j ŕ ∈ <s_i; s_i + 1> × (r_j + 1) are merged if they share similar parameters in their corresponding spline intervals (r_i). A fixed threshold we use in the current version depends on data; it has to be large enough to allow for parameters changes during a curved face following but simultaneously not too large to merge different faces together.

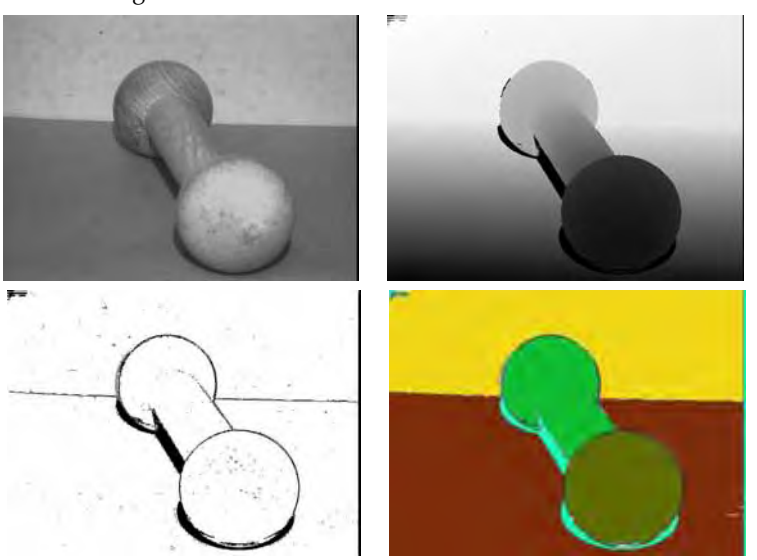


Fig. 7. Dumbbell intensity and range image, the combined edge map and the segmentation result.

5. Evaluation Methodology

For the segmentation quality evaluation we decided to use test data and methodology provided by (Hoover et al., 1996a) and (Hoover et al., 1996b) authors provided not only range image data, but the data together with ground truth segmentation of all images and a tool, which measures quality of segmentation results. Although no such experimental data can prove properties of a segmentation algorithm this set is large enough to suggest its expected behaviour and enables to rank the tested algorithm with some other previously published ones. Last but not least it is a rare world-wide accepted methodology for comparing planar face range image segmentation results.

The problem of image segmentation is a classical one and yet different definitions exist in the literature. Thus we begin by formally defining the problem we consider here. Let R represents the entire image region. We may view segmentation as a process that partitions R into n subregions $R_1, R_2, \ldots, R_n$, such that

1. $\bigcup_{i=1}^{n} R_i = R$
2. R_i is a connected region, i=1,2,…,n
3. $R_i \cap R_j = 0$ for all i and j, $i \neq j$
4. $Pred(R_i)$ = TRUE for i=1,2,…,n and
5. $Pred(R_i \cap R_j)$= FALSE for $i \neq j$

where $Pred(R_i)$ is a logical predicate over the points in set R_i and 0 is the empty set.

In some works the item 5 of this definition is modified to apply only to adjacent regions, as no bordering regions may well have the same properties, sometimes the item 2 is completely left out. Besides these inconsistencies, there are technical difficulties in using this definition for range image segmentation. Some range pixels do not contain accurate depth measurements of surfaces. This naturally leads to allowing non--surface pixels (areas), perhaps of various types. Regarding the above definition, non-surface areas do not satisfy the same predicate constraints (items 4 and 5) as regions that represent surfaces. It is also often convenient to use the same region label for all non-surface pixels in the range image, regardless of whether they are spatially connected. This violates item 2 of the above definition. Finally, we also require that the segmentation be 'crisp'. No sub pixel, multiple or 'fuzzy' pixel labelling are allowed. Comparison of machine segmentation (MS) of a range image to the ground truth (GT) is done as follows. Let M be the number of regions in the MS, and N be the number of regions in the GT. GT does not include any non-surface pixel areas. Similarly, MS does not include any pixels left unlabelled (or not assigned to a surface) by segmenter. Let the number of pixels in each machine-segmented region R_m (where m=1,…, M) be denoted P_M. Similarly, let the number of pixels in each ground truth region R_n (where n=1,…, N) be denoted P_n. Let $O_{mn} = R_m \cap R_n$ be the number of pixels of both regions R_m and R_n whose image coordinates occupy the same range in their respective images. Thus, if there is no overlap between the two regions, $O_{mn} = 0$, while if there is complete overlap, $O_{mn} = P_m = P_n$.

An $M \times N$ contingency table is created, containing O_{mn} for m=1,…, M and n=1,…, N. Implicitly attached to each entry are the percentages of overlap with respect to the size of each region. O_{mn}/P_m represents the percentage of m that the intersection of m and n covers.

Similarly, O_{mn}/P_n represents the percentage of n that the intersection of m and n covers. These percentages are used in determining region segmentation classifications.
We consider five types of region classification: **correct detection**, **over-segmentation**, **under-segmentation**, **missed** and **noise**. Over-segmentation, or multiple detections of a single surface, results in an incorrect topology. Under-segmentation, or insufficient separation of multiple surfaces, results in a subset of the correct topology and a deformed geometry. A missed classification is used when a segmenter fails to find a surface which appears in the image (false negative). A noise classification is used when the segmenter supposes the existence of a surface which is not in the image (false positive). Obviously, these metrics could have varying importance in different applications.
The formulas for deciding classification are based upon a threshold T, where $0.5 < T \leq 1.0$. The value of T can be set to reflect the strictness of definition desired. The following metrics define each classification:

1. An instance of a **correct detection** classification
   A pair of regions R_n in the GT image and R_m in the MS image are classified as an instance of correct detection if
   a) $O_{mn} \geq T \times P_m$ (at least T percent of the pixels in region R_m in then MS image are marked as pixels in region R_n in the GT image), and
   b) $O_{mn} \geq T \times P_n$ (at least T percent of the pixels in region R_n in then GT image are marked as pixels in region R_m in the MS image).

2. An instance of an **over-segmentation** classification
   A region R_n in the GT image and a set of regions in the MS image $R_{m1},\ldots,R_{mx}$, where $2 \leq x \leq M$, are classified as an instance of over-segmentation if
   a) $\forall\ i \in x,\ O_{mi\,n} \geq T \times P_m$ (at least T percent of the pixels in each region R_{mi} in the MS image are marked as pixels in region Rn in the GT image), and
   b) $\sum_{i=1}^{x} O_{mi\,n} \geq T \times P_n$ (at least T percent of the pixels in region R_n in the GT image are marked as pixels in the union of regions $R_{m1},\ldots, R_{mx}$ in the MS image).

3. An instance of an **under-segmentation** classification
   A set of regions in the GT image $R_{n1},\ldots, R_{nx}$, where $2 \leq x \leq M$, and a region R_m in the MS image are classified as any instance of under-segmentation if
   a) $\sum_{i=1}^{x} O_{mni} \geq T \times P_m$ (at least T percent of the pixels in region R_m in the MS image are marked as pixels in the union of regions $R_{n1},\ldots,R_{nx}$ in the GT image), and
   b) $\forall\ i \in x,\ O_{m\,ni} \geq T \times P_{ni}$ (at least T percent of the pixels in each region R_{ni} in the GT image are marked as pixels in region R_m in the MS image).

4. An instance of a **missed** classification
   A region R_n in the GT image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as missed.
5. An instance of a noise classification
   A region R_m in the MS image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as noise.

The authors of (Hoover et al., 1996a) created publicly available tool, which measures results of segmentation using described performance metrics. There are certainly many other possibilities how to compare segmentation results and some of them will result in different algorithms rating but above performance metrics is the only one which is generally accepted and hence enables mutual comparison of different published results.

6. Results

We tested the algorithm on a test set (Powel et al., 1998) of 39 range images from scenes containing planar, cylindrical, spherical, conical and toroidal object surfaces. This set was created by authors of (Powel et al., 1998) using a K^2T structured light scanner model GRF-2. The scanner precision is 0.1 mm and data were quantized into a $640 \times 480 \times 8$ bit data space. Single scenes have between 1 to 120 surface patches of varying sizes. We compared our results Fig. 8. with three previously published methods (Besl & Jain, 1988), (Jiang & Bunke, 1998), (Haindl & Žid, 1998). As can be seen on Fig. 8. (colour version) our method outperforms these alternative methods. The average improvement over our previously published method (Haindl & Žid, 1998) in the correct segmentation criterion (Hoover et al., 1996) is 30%. Alternatively the results were evaluated also visually comparing range data segmentation results with corresponding intensity images (Figs. 7., 9.).
Visual comparison of the results demonstrates very good quality of detected borders using our algorithm. The borders are clean and accurately located. The segmentation algorithm properly found most required non-planar object surfaces in our test examples.

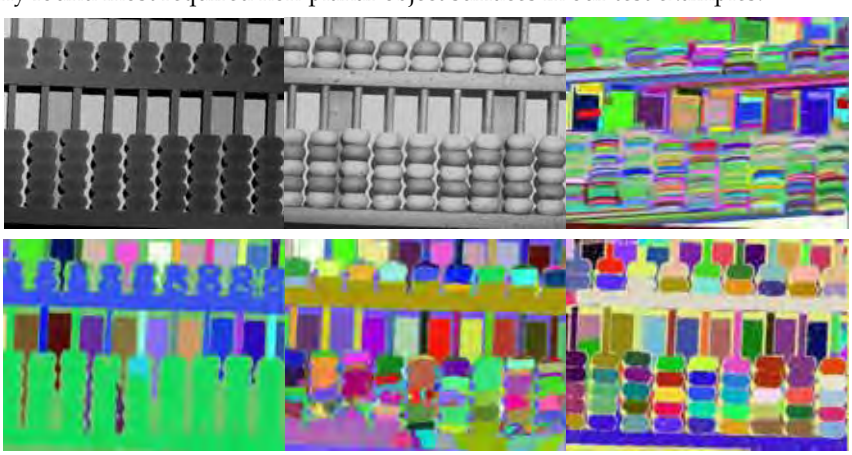


Fig. 8. Range, intensity measurements and the corresponding Besl & Jain (Besl & Jain, 1988) (upper row), UB (Jiang & Bunke, 1998), (Haindl & Žid, 1998) and the presented method abacus segmentation results.

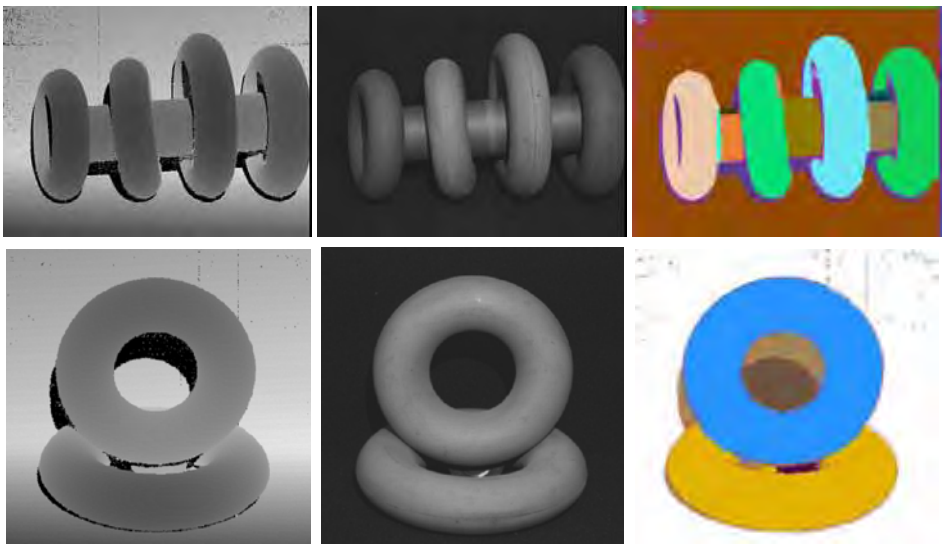


Fig. 9. Annuloid range, intensity and segmentation results images.

7. Conclusions

We proposed novel fast and accurate range segmentation method based on the combination of range & intensity profile modelling and curve-based region growing. A range profile is modelled using an adaptive simultaneous regression model. The recursive adaptive predictor uses spatial correlation from neighbouring data what results in improved robustness of the algorithm over rigid schemes, which are affected with outliers often present at the boundary of distinct shapes. A parallel implementation of the algorithm is straightforward, every image row and column can be processed independently by its dedicated processor. The region growing step is based on the cubic spline curve model. The algorithm performance is demonstrated on the set of test range images available on the University of South Florida web site. These preliminary test results of the algorithm are encouraging; the proposed method was mostly able to find objects present in all our experimental scenes with excellent border localization precision and outperformed the alternative segmenters. The proposed method is fast and numerically robust so it can be used in an on-line virtual reality acquisition system. However further work is still needed to replace current fixed region growing threshold with an adaptive threshold which could accommodate different types of range data, to test the performance on noisy laser data as well as on scenes with large number of curved faces.

8. Acknowledgements

This research was supported by the EC project no. FP6-507752 MUSCLE, grants No.A2075302, 1ET400750407 of the Grant Agency of the Academy of Sciences CR and partially by the MSMT grants 1M0572 DAR, 2C06019.

9. References

Besl P. J., Jain R. C., *Three-dimensional object recognition.* ACM Computing Surveys, 17, no.1, pp. 75-145, 1985.

Besl P. J., Jain R. C., *Segmentation Through Variable-Order Surface Fitting.* IEEE Transactions PAMI, Vol. 10, No.2, pp. 167-192, 1988.

Bock M. E., Guerra C., *Segmentation of range images through the integration of different strategies*, Vision, Modeling, and Visualization, pp. 27-33, 2001.

Chang Y. L., Li X., *Adaptive image region-growing*, IEEE Transaction on Image Processing, vol. 3, pp. 868–872, 1994.

Chu C., Aggarwal J. K., *The integration of image segmentation maps using region and edge information*, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 15, pp. 241–252, 1993.

Dubrawski A., Sawwa R., *Laserowe trójwymiarowe czujniki odległości w nawigacji ruchomych robotów (3-D Laser Range Finders for Mobile Robots' Navigation)*, 5th National Conference on Robotics, Swieradow Zdroj, Poland 1996.

Flynn P. J., Jain A. K., *BONSAI: 3D Object Recognition Using Constrained Search.*, IEEE Transactions PAMI, Vol. 13, No. 10, pp. 1066-1075, 1991.

Forsyth D. A., Ponce J., *Computer Vision A Modern Approach*, Prentice Hall, Pearson Education, Inc., 2003.

Haddon J., Boyce J., *Image segmentation by unifying region and boundary information*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, pp. 929-948, 1990.

Haindl M., Šimberová S., *Theory & Applications of Image Analysis*, chapter *A Multispectral Image Line Reconstruction Method*, pages 306--315, World Scientific Publishing Co., Singapore, 1992.

Haindl M., Žid P., *Range image segmentation by curve grouping*, In K. Dobrovodský, editor, Proceedings of the 7th International Workshop on Robotics in Alpe-Adria-Danube Region, pages 339--344, Bratislava, June 1998. ASCO Art.

Haralick R. M., Shapiro L. G., *Survey: Image segmentation techniques*, Computer Vision, Graphics, and Image Processing, vol. 29, pp. 100–132, 1985.

Hijjatoleslami S. A., Kittler J., *Region growing: A new approach*, IEEE Transaction on Image Processing, vol. 7, pp. 1079–1084, 1998.

Hoffman R. L., Jain A. K., *Segmentation and Classification of Range Images.*, IEEE Trans. PAMI, Vol. 9, No. 5, pp. 608-620, 1987.

Hoover A., Jean-Baptiste G., Jiang X. Y., Flynn P. J., Bunke H., Goldof D. B., Bowyer K., Eggert D. W., Fitzgibbon A., Fisher R. B., *An Experimental Comparison of Range Image Segmentation Algorithms.*, IEEE Trans. PAMI, 18, no.7, pp. 673-689, 1996a.

Hoover A., Jean-Baptiste G., Jiang X. Y., Flynn P. J., Bunke H., Goldof D. B., Bowyer K., Eggert D. W., Fitzgibbon A., Fisher R. B., *Range image segmentation comparison segmenter codes and results*, http://marathon.csee.usf.edu/range/seg-comp/results.html, 1996b.

Horn B. K. P., *Robot Vision*, MIT Press, 1986.

Internet: *OSU (MSU/WSU) Range Image Database*, http://sampl.eng.ohio-state.edu/~sampl/data/3DDB/RID/index.htm, 1999.

Jiang X. Y., Bunke H., *Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping.*, Machine Vision and Applications, 7, no. 2, pp. 115-122, 1994.

Jiang X. Y., Bunke H., *Range image segmentation: Adaptive grouping of edges into regions.*, Asian Conference on Computer Vision, Hong Kong, 1998.

Jiang X., Bowyer K., Morioka Y., Hiura S., Sato K., Inokuchi S., Bock M. E., Guerra C., Loke R. E., du Buf J. M. H., *Some Further Results of Experimental Comparison of Range Image Segmentation Algorithms*, 15th International Conference on Pattern Recognition, Vol. 4, pp. 877-881, 2000.

Lee K. M., Meer P., Park R. H., *Robust Adaptive Segmentation of Range Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 2, pp. 200-205, 1998.

Loke R. E. and du Buf J. M. H., *Hierarchical 3D data segmentation by shape-based boundary refinement in an octree using orientation-adaptive filtering*, Tech. Report UALG-ISACS-TR03, 1998.

Dr McIvor A. M., Penman D. W. and Waltenberg P. T., *Simple Surface Segmentation.*, DICTA/IVCNZ97, Massey University, pp. 141-146, 1997.

Pal N. R., Pal S. K., *A review of image segmentation techniques*, Pattern Recognition, Vol 26, pp. 1277-1294, 1993.

Palmer P. L., Dabis H., Kittler J., *A performance measure for boundary detection algorithms*, Computer Vision and Image Understanding, vol. 63, pp. 476–494, 1996.

Pavlidis T., Liow Y. T., *Integrating region growing and edge detection*, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 12, pp. 225–233, 1990.

Powell M., Bowyer K., Jiang X., Bunke H., *Comparing curved-surface range image segmenters*, In {\em ICCV'98}, Bombay, 1998. IEEE.

Sinha S. S., Jain R., *Handbook of Pattern Recognition and Image Processing*, Wiley, New York, 1994.

Wilson R. and Spann M., *Image Segmentation and Uncertainty*, Research Studies Press Ltd., Letchworth, 1988.

Zhang X., Zhao D., *Range image segmentation via edges and critical points.*, Proc. SPIE, 2501, no. 3, pp. 1626-1637, 1995.

3

Moving Cast Shadow Detection

Wei Zhang[1], Q.M. Jonathan Wu[1] and Xiangzhong Fang[2]
University of Windsor[1], Shanghai Jiao Tong University[2]
Canada[1], China[2]

1. Introduction

Moving shadow detection is an important topic in computer vision applications, including video conference, vehicle tracking, and three-dimensional (3-D) object identification, and has been actively investigated in recent years. Because, in real world scenes, moving cast shadows may be detected as foreground object and plauge the moving objects segmentation. For example, in traffic surveillance situation, shadows cast by moving vehicles may be segmented as part of vehicles, which not only interfere with the size and shape information but also generate occlusions (as Fig. 1 illustrates). At the same time, moving cast shadow detection can provide reference information to the understanting of the illumination in the scenes. Therefore, an effective shadow detection algorithm can greatly benefit the practical image analysis system.

Fig. 1. Examples of moving cast shadows.

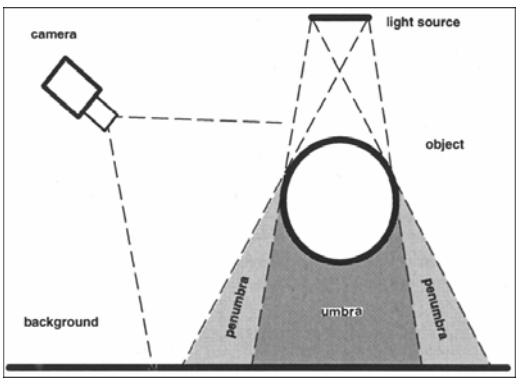


Fig. 2. Illumination model of moving cast shadows: the umbra, penumbra, and geometric relationship.

Essentially, shadow is formed by the change of illumination conditions and shadow detection comes down to a problem of finding the illumination invariant features. From the viewpoint of geometric relationship, shadow can be divided into umbra and penumbra (Stander et al., 1999). The umbra corresponds to the background area where the direct light is almost totally blocked by the foreground object, whereas in the penumbra area, the light is partially blocked (as Fig. 2 illustrates). From the viewpoint of motion property, shadow can be divided into static shadow and dynamic shadow. Static shadow is cast by static object while dynamic shadow is cast by moving object. In video surveilance application, static shadows have little effect on the moving objects segmentation. Therefore, we concentrate on the detection of dynamic/moving cast shadows in the image sequence captured by static camera in this chapter.

2. Illumination property of cast shadow

For an image acquired by camera, the intensity of pixel f(x,y) can be given as:

$$f(x,y)= i(x,y)\times r(x,y) \tag{1}$$

where i(x,y) represents the illumination component and r(x,y) represents the reflectance component of object surface. i(x,y) is computed as the amount of light power per receiving object surface area and can further be expressed as follows (Stander et al., 1999).

$$i(x,y) = \begin{cases} c_a + c_p \cdot \cos(j) & \text{illuminated area} \\ c_a + t(x,y) \cdot c_p \cdot \cos(j) & \text{penumbra area} \\ c_a & \text{umbra area} \end{cases} ; \tag{2}$$

where

- c_p intensity of the light source;
- φ angle enclosed by light source direction and surface normal;
- c_a intensity of ambient light;
- t transition inside the penumbra which depends on the light source and scene geometry, and $0\leq t(x,y)\leq 1$.

Many works have been put forward in the literature for moving shadow detection. From

the viewpoint of the information and model utilized, these methods can be classified into three categories: color model, textural model, and geometric model. Additionally, statistical model is used to tackle the problem. Most of the state-of-the-art are based on the reference image and we consider it has been acquired beforehand. Let the reference image and shaded image be B and F, respevtively. In the following part of this chapter, we introduce each categories of methods for moving cast shadow detection.

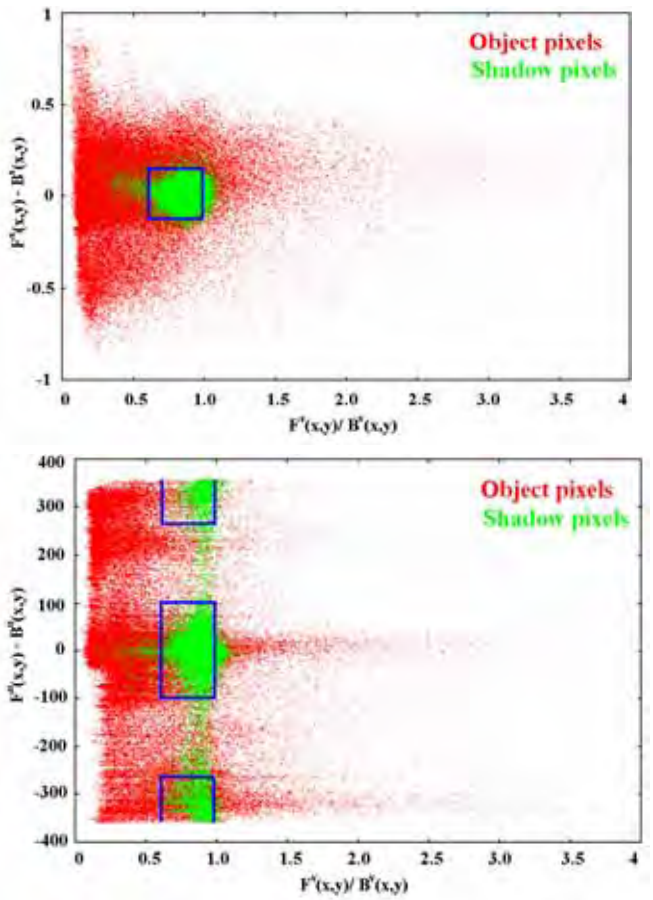


Fig. 3. The distribution of the background difference and background ratio in HSV color space: shadow pixels and foreground pixels.

3. Colour/Spectrum-based shadow detection

The color/spectrum model attempts to describe the color change of shaded pixel and find the color feature that is illumination invariant. Cucchiara *et al.* (Cucchiara et al., 2001; Cucchiara et al., 2003) investigated the Hue-Saturation-Value (HSV) color property of cast

shadows, and it is found that shadows change the hue component slightly and decrease the saturation component significantly. The distribution of $F^V(x, y)/B^V(x, y)$, $F^S(x, y)-B^S(x, y)$, and $|F^H(x, y)-B^H(x, y)|$ are given in Fig. 3 for shadows pixels and foreground pixels, respectively. It can be found that shadow pixels cluster in a small region and have distinct distribution compared with foreground pixels. The shadows are then discriminated from foreground objects by using empirical thresholds on HSV color space as follows.

$$(\alpha \le \frac{F^V(x,y)}{B^V(x,y)} \le \beta)\textbf{AND}((F^S(x,y) - B^S(x,y)) \le \tau_S)\textbf{AND}(\left|F^H(x,y) - B^H(x,y)\right| \le \tau_H) \quad (3)$$

By using above method, the shadow pixels can be discriminated from foreground pixels effectively. This method has been included in the Sakbot system (Statistical and Knowledge-Based Object Tracker).

Salvador *et al.* (Salvador et al. 2004) proposed a normalized RGB color space, $C_1C_2C_3$, to segment the shadows in still images and video sequences. The $C_1C_2C_3$ is defined as follows.

$$\begin{aligned} C_1(x,y) &= \arctan\frac{R(x,y)}{\max(G(x,y),B(x,y))}; \\ C_2(x,y) &= \arctan\frac{G(x,y)}{\max(R(x,y),B(x,y))}; \\ C_3(x,y) &= \arctan\frac{B(x,y)}{\max(R(x,y),G(x,y))}; \end{aligned} \quad (4)$$

After integrating the intensity of neighbouring region, the shadow is detected as the pixels change greatly in $C_1C_2C_3$ colour space. Considering the shadow decrease the intensity of RGB component in a same scale, it can be found that $C_1C_2C_3$ is illumination invariant.

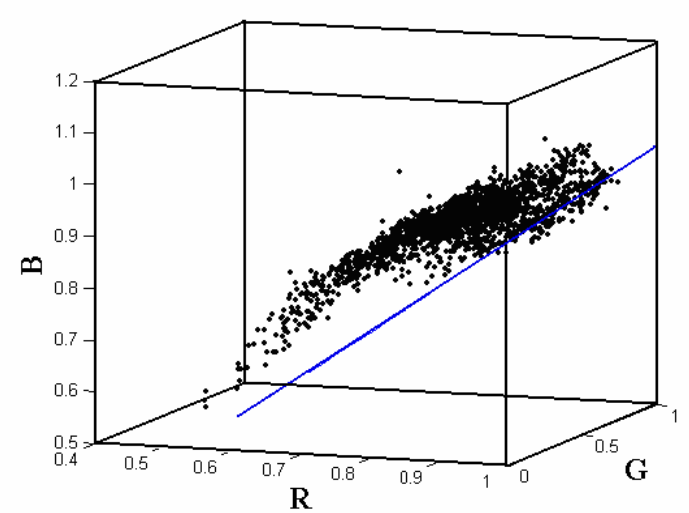


Fig. 4. A scatter plot in the color ratios space of a shaded pixels set. The line corresponds to the equal ratio in RGB components.

Siala *et al.* (Siala et al., 2004) consider the pixel's intensity change equally in RGB colour components and a diagonal model is proposed to describe the color distortion of shadow in RGB space. The color distortion is defined as ($d_R=F_R/B_R$, $d_G=F_G/B_G$, $d_B=F_B/B_B$), and the color distortion of shaded pixel is distributed near the line $d_R=d_G=d_B$ (as show in Fig. 4), which does not hold for foreground objects. Therefore, the shadow pixels are discriminated from foreground objects according to the distance between pixel's color distortion and the line $d_R=d_G=d_B$.

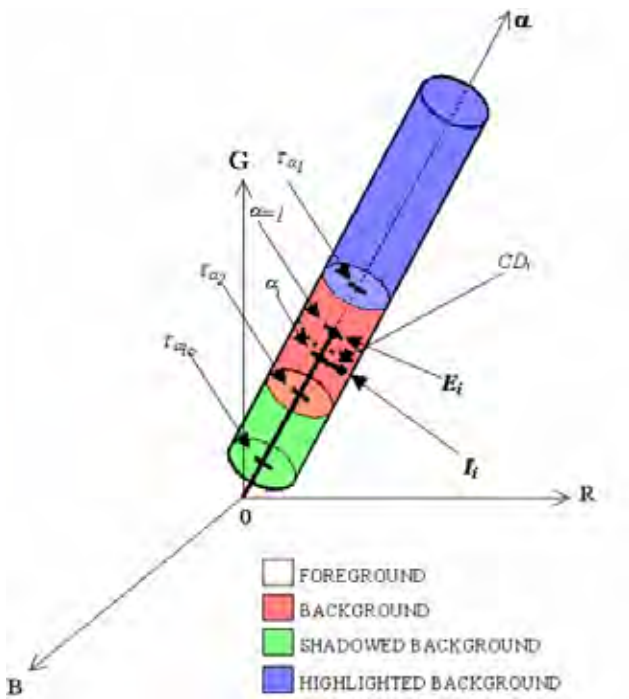


Fig. 5. Pixels classification using the normalized color distortion and normalized brightness distortion: original background, shaded background, highlight background, and moving foreground objects.

Horprasert *et al.* (Horprasert et al., 1999) proposed a computational color model which separates brightness from the chromaticity component using brightness distortions (BD) and chromaticity distortions (CD), which are defined as follows.

$$BD(x,y)=\frac{\frac{F_R(x,y)\cdot\mu_R(x,y)}{\sigma_R^2(x,y)}+\frac{F_G(x,y)\cdot\mu_G(x,y)}{\sigma_G^2(x,y)}+\frac{F_B(x,y)\cdot\mu_B(x,y)}{\sigma_B^2(x,y)}}{\left[\frac{\mu_R(x,y)}{\sigma_R(x,y)}\right]^2+\left[\frac{\mu_G(x,y)}{\sigma_G(x,y)}\right]^2+\left[\frac{\mu_B(x,y)}{\sigma_B(x,y)}\right]^2};$$

$$CD(x,y)=\sqrt{\left(\frac{F_R(x,y)-BD\cdot\mu_R(x,y)}{\sigma_R(x,y)}\right)^2\left(\frac{F_G(x,y)-BD\cdot\mu_G(x,y)}{\sigma_G(x,y)}\right)^2\left(\frac{F_B(x,y)-BD\cdot\mu_B(x,y)}{\sigma_B(x,y)}\right)^2}; \quad (5)$$

In which (μ_R, μ_G, μ_B) and (σ_R, σ_G, σ_B) are the arithmetic means and variance of the pixel's red, green, and blue values computed over N background frames. By imposing thresholds on the normalized color distortion (NCD) and normalized brightness distortion (NBD), the pixels are classified into original background, shaded background, highlight background, and moving foreground objects as follows.

$$\begin{cases} \text{Foreground}: \text{NCD} > \tau_{CD} \text{ OR NBD} < \tau_{alo}, \text{else} \\ \text{Background}: \text{NBD} < \tau_{a1} \text{ AND NBD} < \tau_{a2}, \text{else} \\ \quad \text{Shadow}: \text{NBD} < 0, \text{else} \\ \quad \text{Highlight}: \text{otherwise} \end{cases} \tag{6}$$

The strategy used in Eq. (6) is depicted in Fig. 5.

Nadimi, S. & Bhanu, B (Nadimi, S. & Bhanu, B., 2004) employed a physical approach for moving shadow detection in outdoor scenes. A dichromatic reflection model and a spatio-temporal albedo normalization test are used for learning the background color and separating shadow from foreground in outdoor image sequences. According to the dichromatic reflection model, pixel value F(x,y) in the outdoor scene can be represented as follows.

$$F(x,y) = \int_{\Delta\lambda_1} K_{(x,y),1} L_{(x,y),1}(\lambda) f(l,e,s) d\lambda + \int_{\Delta\lambda_2} K_{(x,y),2} L_{(x,y),1}(\lambda) d\lambda; \tag{7}$$

in which the first and second items correspond to the intensiy caused by the sun and sky; $K_{(x,y),1}$ and $K_{(x,y),2}$ are the coefficient of reflectances due to sun and sky; $L_{(x,y),1}$ and $L_{(x,y),2}$ are intensity of the illumination sources of sun and sky; f(l,e,s) is geometric term; l is the incident angle of illumination; e is the angle for viewing direction; s is the angle for specular reflection. The spatio-temporal albedo H between pixel F(x,y) and its neighboring pixel (take F(x+1,y) as example) is defined as follows.

$$\begin{gathered} H(F(x,y),F(x+1,y)) = \frac{R_1 - R_2}{R_1 + R_2}; \\ R_1 = \frac{F_{t+1}(x,y) - F_t(x,y)}{F_{t+1}(x,y) + F_t(x,y)}; R_2 = \frac{F_{t+1}(x+1,y) - F_t(x+1,y)}{F_{t+1}(x+1,y) + F_t(x+1,y)}; \end{gathered} \tag{8}$$

Pixel F(x,y) and F(x+1,y) is assumed to have the same reflectance if the following condition is satisfied:

$$C[F(x,y),F(x+1,y)] = \begin{cases} 1 & \text{if } |H(F(x,y),F(x+1,y))| < T \\ 0 & \text{Otherwise} \end{cases}; \tag{9}$$

Cavallaro *et al.* (Cavallaro et al., 2005) detected shadow by exploiting color information in a selective way. In each image the relevant areas to analyze are identified and the color components that carry most of discriminating information are selected for shadow detection.

Color model has shown its powerfulness in shadow detection. Nevertheless, the foreground objects may have similar color with the moving shadows, and it becomes not reliable to detect moving shadows by using only the color information of the isolated points.

4. Texture-based shadow detection

The principle behind the textural model is that the texture of foreground objects is different with that of the background, while the texture of shaded area remains the same as that of the background.

In (Xu et al., 2005), several techniques have been developed to detect moving cast shadows in a normal indoor environment. These techniques include the generation of initial change detection masks and canny edge maps, the detection of shadow region by multi-frame integration, edge matching, conditional dilation, and post-processing (as Fig.6 illustrates).

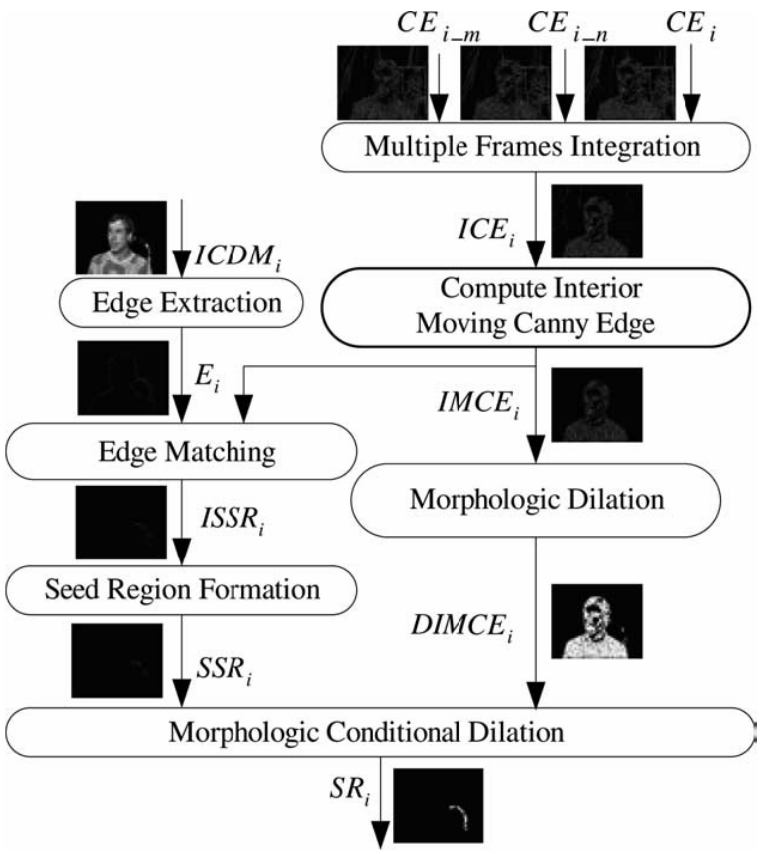


Fig. 6. Moving cast shadow detection by using the edge information.

McKenna *et al.* (McKenna et al., 2000) assumed cast shadow results in significant change in intensity without much change in chromaticity. Each pixel's chromaticity is modeled using its means and variances, and each background pixel's first-order gradient is modeled by using gradient means and magnitude variances. The moving shadows are then classified as background if the chromaticity or gradient information supports their classification. Leone *et al.* (Leone et al., 2006) represented textural information in terms of redundant systems of

functions, and the shadows are discriminated from foreground objects based on a pursuit scheme by using an over-complete dictionary. Matching Pursuit algorithm (MP) is used to represent texture as linear combination of elements of a big set of functions, and MP selects the best little set of atoms of 2D Gabor dictionary for features selection. Zhang *et al.* (Zhang et al., 2006) used the normalized coefficients of the orthogonal transformation for moving cast shadow detection. Five kind of orthogonal transforms (DCT, DFT, Haar Transform, SVD, and Hadamard Transform) are analyzed, and their normalized coefficients are proved to be illumination invariant in a small image block. The cast shadows are then detected by using a simple threshold on the normalized coefficients (as Fig.7 illustrates).
Zhang *et al.* (Zhang et al., 2006) use the ratio edge for shadow detection, which are defined as follows.

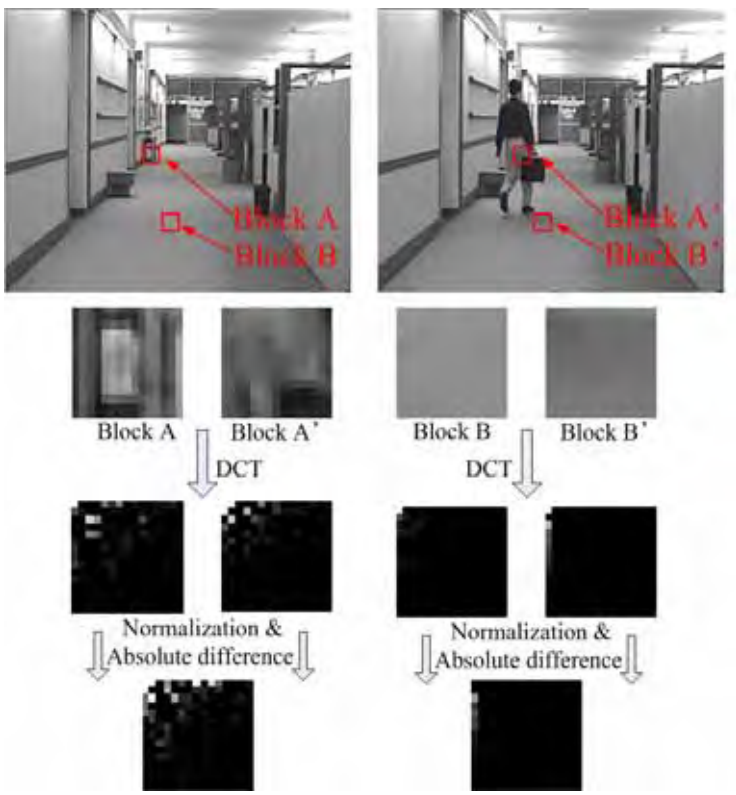


Fig. 7. Moving cast shadow detection based on the normalized coefficients of orthogonal transformation.

$$\Theta(x,y)=\{F(x+i,y+j) \mid 0<i^2+j^2\leq r^2\} \tag{10}$$

$$R(x,y)=\sum_{(i,j):F(i,j)\in\Theta(x,y)}\frac{F(x,y)}{F(i,j)}; \tag{11}$$

According to the illumination model in Eq. (2), the ratio edge is proved to be illumination

invariant. The shadow are then detected by imposeing a threshold on the ratio edge difference $R_D(x,y)$ defined as follows.

$$R_D(x,y) = \sum_{\substack{(i,j):\ B(i,j)\in\Theta_B(x,y)\\ F(i,j)\in\Theta_S(x,y)}} \left(\frac{B(x,y)}{B(i,j)} - \frac{F(x,y)}{F(i,j)}\right)^2; \tag{12}$$

Fig. 8. The textural property of ratio edge.

in which $\Theta_B(x,y)$ and $\Theta_S(x,y)$ are the neighoring region of B(x,y) and F(x,y), respectively. The ratio edge of Eq. (12) is given in Fig.8, it can be seen that ratio edge can represent the quanlity of the texture in the neighboring region.

Fung *et al.* (Fung et al., 2002) analyzed the characteristics of cast shadows in the luminance, chrominance, gradient density, and geometry domains, and a combined probability map is obtained which is called as shadow confidence score (SCS), as shown in Fig. 9.

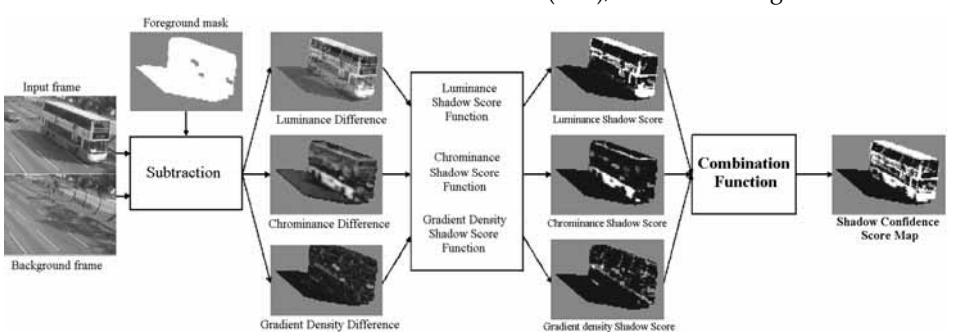


Fig. 9. Moving cast shadow detection based on shadow confidence score.

From the edge map of the input image, each edge pixel is examined to determine whether it belongs to the vehicle region based on its neighboring SCSs. The cast shadows are identified as those regions with high SCSs, which are outside the convex hull of the selected vehicle's edge pixels.

Textural model may be the most promising technique for shadow detection, whereas the state-of-the-art of textural model are intricate in implementation. Moreover, in the

homogeneous regions of the images, the textural information of the scenes may be very faint and cannot be captured by traditional methods.

5. Geometry-based shadow detection

Geometric model makes use of the camera location, the ground surface, and the object geometry, etc., to detect the moving cast shadows.

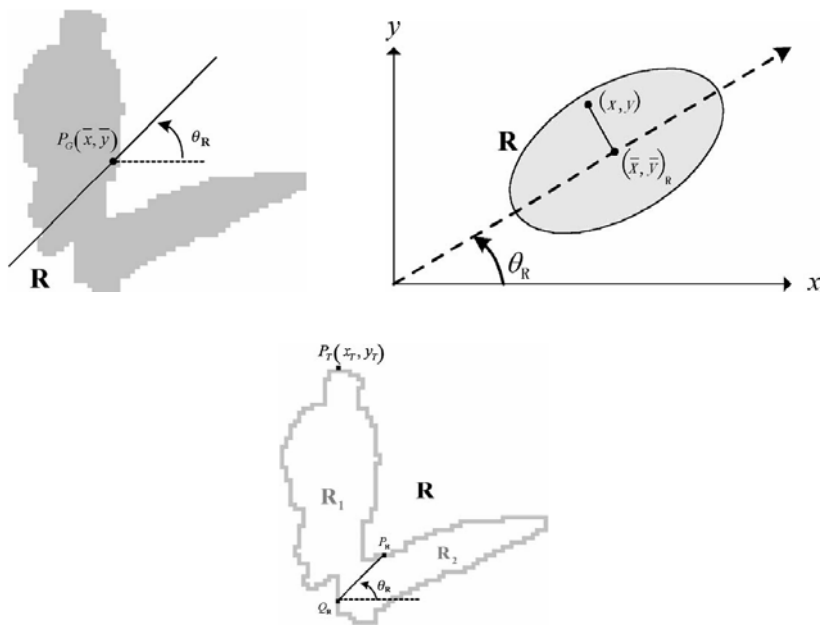


Fig. 10. The Gaussian geometric shadow model used for the detection of pedestrian's shadow.

In (Hsieh et al., 2003), Gaussian shadow model was proposed to detect the shadows of pedestrian. The model is parameterized with several features including the orientation, mean intensity, and center position of a shadow region (as Fig.10 illustrates), with the orientation and centroid position being estimated from the properties of object moments.

Hsieh *et al.* (Hsieh et al., 2004; Hsieh et al., 2006) proposed a histogram-based method to detect different lane dividing lines from traffic video sequence. According to these lines, a line-based shadow modeling process is applied to eliminate the shadows of vehicles. Two kinds of lines are used, including the ones parallel and vertical to lane directions, which can be used to eliminate shadows in the different positions of vehicles. Yoneyama *et al.* (Yoneyama et al., 2003; Yoneyama et al., 2005) proposed joint 2D vehicle/shadow models to suppress the moving shadows of vehicles. The proposed 2D vehicle/shadow models are classified into six types (as Fig.11 illustrates) and the parameters of these models can be estimated by fitting the segmented vehicles with these models.

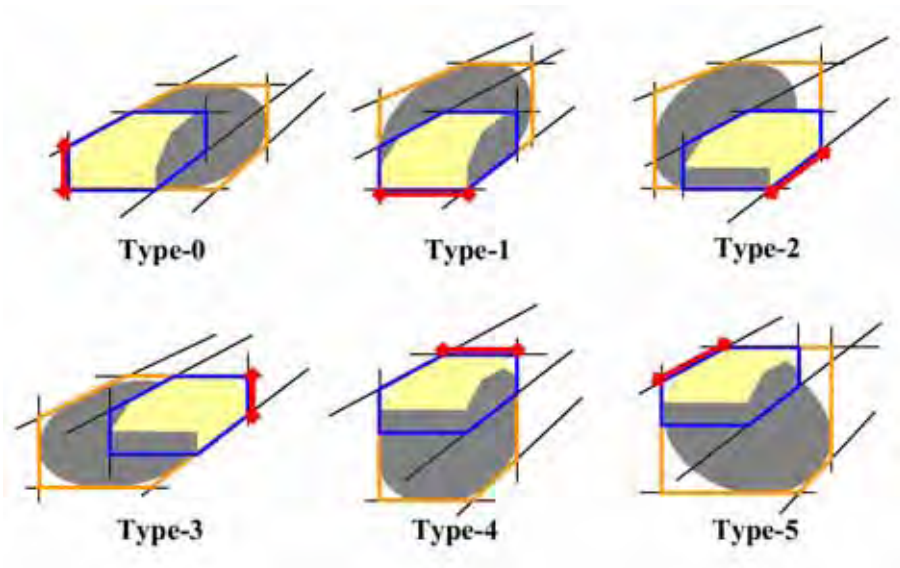


Fig. 11. Six vehicle model types with the corresponding cast shadow.

All these methods of geometric model strongly depend on the geometric relationships of the objects in the scenes, and when these geometric relationships change, these methods become ineffective.

6. Statistical inference for shadow model

Another useful tool for shadow detection is statistical model, which can further improve the performance of different shadow model. Most of these methods are based on the noise shadow model:

$$F(x,y) = \Phi(x,y) \cdot B(x,y) + \varepsilon(x,y); \varepsilon(x,y) \sim N(0,\sigma^2);$$
$$\Phi(x,y) = \frac{c_a + t(x,y) \cdot c_p \cdot \cos(j)}{c_a + c_p \cdot \cos(j)}; 0 \le \Phi(x,y) \le 1; \tag{13}$$

in which t(x,y), c_p, and c_a are ones defined in Eq. (2).
Toth *et al.* (Toth et al., 2004) use the quantity given in Eq. (14) for shadow detection, which is normally distributed with variance $(1+1/\Phi^2)\ \sigma^2$.

$$\tilde{B}(x,y) - \frac{1}{\Phi(x,y)} \cdot F(x,y) = \varepsilon(x,y) - \frac{1}{\Phi(x,y)} \cdot \varepsilon(x,y);$$
$$\tilde{B}(x,y) = B(x,y) + \varepsilon(x,y); \tag{14}$$

Each moving pixel is then classified into foreground object or shadow by performing a significance test. Wang *et al.* (Wang et al., 2003) modeled the background, shadow, and edge information as Gaussian distributions which are updated adaptively. A Bayesian framework

is then utilized to describe the relationships among the segmentation label, background intensity, and edge information. Markov random field (MRF) is used to improve the spatial connectivity of the segmented regions. Nicolas *et al.* (Martel-Brisson, N. & Zaccarin, A., 2005) introduce Gaussian mixture model (GMM) for the detection of moving cast shadows. The proposed algorithm consists of identification the distributions that could represent shadows, modification the learning rates of the distributions to allow them to converge within the GMM, and build of a GMM for moving shadows by using identified distributions. Mikic *et al.* (Mikic et al., 2000) model the shadow pixel as a Gaussian distribution with ($\mu_{S,R}$, $\mu_{S,G}$, $\mu_{S,B}$, $\sigma_{S,R}$, $\sigma_{S,G}$, $\sigma_{S,B}$) being the mean and variance, while the illuminated pixel is also model as a Gaussian distribution with ($\mu_{L,R}$, $\mu_{L,G}$, $\mu_{L,B}$, $\sigma_{L,R}$, $\sigma_{L,G}$, $\sigma_{L,B}$) being the mean and variance. Let D=diag(d_R,d_G,d_B) being the camera response for the same point when it is shadowed. Therefore, we have the following relationships.

$$\begin{aligned} \mu_{S,R} &= d_R\mu_{L,R}, \mu_{S,G} = d_G\mu_{L,G}, \mu_{S,B} = d_B\mu_{L,B}; \\ \sigma_{S,R} &= d_R\sigma_{L,R}, \sigma_{S,G} = d_G\sigma_{L,G}, \sigma_{S,B} = d_B\sigma_{L,B}; \end{aligned} \tag{15}$$

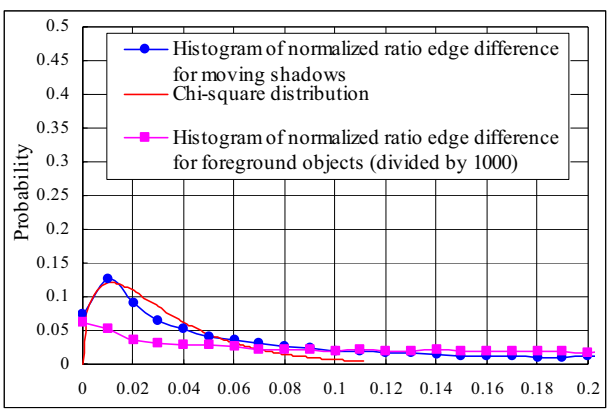


Fig. 12. Histogram of the normalized ratio edge difference for moving cast shadows and foreground, and comparison with Chi-square distribution.

The distribution of foreground objects is assumed to be uniform distribution. A maximum posteriori probability (MAP) is then used to classify the pixel into background(C_1), shadow(C_2), and foreground(C_3) according to its color vector v:

$$p(C_i \mid v) = \frac{p(v \mid C_i) \cdot p(C_i)}{\sum_{j=1,2,3} p(v \mid C_j) \cdot p(C_j)}; \tag{16}$$

In (Zhang et al., 2006), the distribution of the normalized background difference of ratio edge in shaded background area is also analyzed and is approximated to be a chi-square distribution. Therefore, a significance test can be used for automatic shadow detection. The distribution of $R_D(x,y)$ in Eq.(12) is depicted for moving shadows and foreground objects in Fig. 12. It can be found that ratio edge difference of moving shadows has much different distribution compared with that of foreground objects. The distribution of $R_D(x,y)$ of moving

shadows is also compared with Chi-square distribution in Fig. 12 and we can see that a good fitting can be reached.

7. Conclusion

In this chapter, we have provided a brief overview of the works about moving cast shadow detection. The state-of-the-art methods have been categories into color model, textural model, and geometric model according to the information and model utilized, which have been disscussed systemically. Furthermore, all kinds of statistical models have been employed to tackle the problem, which are also analyzed in detail. From the results, we can see that different method is fit for different situation and it is very hard to get a method in common use. Therefore, the future work may be the fusion of different information by statistical model to realize robust shadow detection.

8. References

Chien, S-Y., Ma, S-Y. & Chen L-G. (2002). Efficient moving object segmentation algorithm using background registration technique. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 577-586.

Cucchiara, R., Grana, C., Piccardi, M. & Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* vol. 25, no. 10, pp. 1337-1342.

Cucchiara, R., Grana, C., Piccardi, M., Prati, A. & Sirotti, S. (2001). Improving Shadow Suppression in Moving Object Detection with HSV Color Information. *Proceeding of IEEE International Conference on Intelligent Transportation Systems*, pp. 334-339.

Fung, G. S. K., Yung, N. H. C., Pang, G. K. H. & Lai, A. H. S. (2002). Effective moving cast shadow detection for monocular color traffic image sequences. *Optical Engineering*, vol. 41, no. 6, pp. 1425-1440.

Haritaoglu, I., Harwood, D. & Davis, L. S. (2000). W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions Pattern Analysis and Machine Intelligence.* vol. 22, no. 8, pp. 809-830.

Horprasert, T., Harwood, D. & Davis, L.S. (1999). A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection. Proceeding of *IEEE International Conference on Computer vision FRAME-RATE Workshop*.

Hsieh, J-W., Hu, W-F., Chang, C-J. & Chen, Y-S. (2003). Shadow elimination for effective moving object detection by Gaussian shadow modeling. *Image and Vision Computing*, vol. 21, pp. 505-516.

Hsieh, J-W., Yu, S-H., Chen, Y-S. & Hu, W-F. (2004). A shadow elimination method for vehicle analysis. Proceeding of *IEEE International Conference on Pattern Recognition*, vol. 4, pp. 372-375.

Hsieh, J-W., Yu, S-H., Chen, Y-S. & Hu, W.-F. (2006) Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Transactions on Intelligent Transportation Systems,* vol. 7, no. 2, pp. 175-187.

Leone, A., Distante, C. & Buccolieri, F. (2006). A shadow elimination approach in video-surveillance context. *Pattern Recognition Letters*, vol. 27, no. 5, pp. 345-355.

Martel-Brisson, N. & Zaccarin, A. (2005). Moving cast shadow detection from a Gaussian mixture shadow model. Proceeding of *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 643-648.

McKenna, J. S., Jabri, S., Duric, Z., Rosenfeld, A. & Wechsler H. (2000). Tracking Groups of People. *Computer Vision and Image Understanding*, vol. 80, pp. 42-56.

Mikic, I., Cosman, P. C., Kogut, G. T. & Trivedi, M. M. (2000). Moving shadow and object detection in traffic scenes. Proceeding of *IEEE International Conference on Pattern Recognition*. vol. 1, pp. 321-324.

Nadimi S. & Bhanu B. (2004). Physical models for moving shadow and object detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1079-1087.

Prati, A., Mikic, I., Trivedi, M. M. & Cucchiara R. (2003). Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 918-923.

Salvador, E., Cavallaro, A. & Ebrahimi, T. (2004). Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, vol. 95, pp. 238-259.

Siala, K., Chakchouk, M., Chaieb, F. & Besbes, O. (2004). Moving shadow detection with support vector domain description in the color ratios space. Proceeding of *IEEE International Conference on Pattern Recognition*. vol. 4, pp. 384-387.

Stander, J., Mech, R. & Ostermann, J. (1999). Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia*, vol. 1, pp. 65-76.

Thongkamwitoon, T., Aramvith, S. & Chalidabhongse, T. H. (2004). An adaptive real-time background subtraction and moving shadows detection. Proceeding of *IEEE International Conference on Multimedia and Expo*. vol. 2, pp. 1459-1462.

Toth, D., Stuke, I., Wagner, A. & Aach, T. (2004). Detection of moving shadows using mean shift clustering and a significance test. Proceeding of *IEEE International Conference on Pattern Recognition*, vol. 4, pp. 260-263.

Wang, Y., Tan, T. & Loe, K-F. (2003). A probabilistic method for foreground and shadow segmentation. Proceeding of *IEEE International Conference on Image Processing*, vol. 3, pp. 937-940.

Xu, D., Li, X., Liu, Z. & Yuan, Y. (2005). Cast shadow detection in video segmentation. *Pattern Recognition Letters*, vol. 26, pp. 91-99.

Yoneyama, A., Yeh, C. H. & Kuo, C-C. J. (2003). Moving cast shadow elimination for robust vehicle extraction based on 2D joint vehicle/shadow models. Proceeding of *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 229-236.

Yoneyama, A., Yeh, C-H. & Kuo, C-C. J. (2005). Robust vehicle and traffic information extraction for highway surveillance. *EURASIP Journal on Applied Signal Processing*. vol. 2005, pp. 2305-2321.

Zhang, W., Fang, X. Z. & Xu, Y. (2006). Detection of moving cast shadows using image orthogonal transform. Proceeding of *IEEE International Conference on Pattern Recognition,* vol. 1, pp. 626-629.

Zhang, W., Fang, X. Z. & Yang, X. K. (2006). Moving cast shadows detection based on ratio edge. Proceeding of *IEEE International Conference on Pattern Recognition*, vol. 4, pp. 73-76.

4

Reaction-Diffusion Algorithm for Vision Systems

Atsushi Nomura*, Makoto Ichikawa**, Rismon H. Sianipar*, and Hidetoshi Miike*
*Yamaguchi University, **Chiba University
Japan

1. Introduction

Vision systems require fundamental algorithms of image processing and vision computing. Algorithms of edge detection, grouping and stereo disparity detection are typical examples. Marr and his collaborators proposed several effective algorithms of edge detection and stereo disparity detection, in particular, in a computational approach (Marr, 1982).

Marr and Hildreth had previously proposed an edge detection algorithm (Marr and Hildreth, 1980), in which edge points were defined as those having a high brightness gradient over space. They utilized the Gaussian filter combined with the Laplacian one; the Gaussian filter removes noise components, and the Laplacian filter senses a brightness gradient. They additionally proposed an alternative algorithm that utilizes difference of two Gaussian filters having two different space constants of excitation and inhibition. Both of these two algorithms utilize the Gaussian filter; the output of the filter is equivalent to the solution of the diffusion equation. Their proposal highly attracted other researchers' attention, resulting in the development of many edge detection algorithms starting from the Gaussian filter or the diffusion equation, in which, for example, anisotropy was introduced into the diffusion equation (Perona & Malik, 1990).

Regarding stereo disparity detection, Marr and Poggio proposed "the cooperative algorithm" (Marr and Poggio, 1976; Marr et al., 1978). Stereo cameras project a target point located in a three-dimensional world onto two points on their left and right image planes; stereo disparity refers to the difference of the two points. A stereo disparity map helps to reconstruct the three-dimensional world and thus has many applications in vision systems. To construct a reliable stereo disparity map, Marr and Poggio made two important constraints, one of which is that spatial adjacent points on a stereo disparity map must have similar disparity levels. This constraint allows us to propagate disparity information in a spatial local area. Therefore, the cooperative algorithm for the stereo disparity detection utilizes an information propagation mechanism, which roughly refers to the mechanism of diffusion. Using this cooperative algorithm, other researchers have proposed several methods of stereo disparity detection (Zitnick & Kanade, 2000).

There are several interesting approaches to image processing and vision computing not only in the field of computer science, but also in the natural sciences. Kuhnert et al. demonstrated that a chemical reaction system solves the typical image processing tasks of edge detection

and segmentation (Kuhnert, 1986; Kuhnert et al., 1989). The mathematical model of the chemical reaction system cited in their demonstration is a type of reaction-diffusion system, which consists of non-linear reaction terms combined with diffusion equations. Their successful demonstration showing the capability of the reaction-diffusion system to process images strongly motivated us to further develop our image processing and vision computing algorithms by incorporating the reaction-diffusion system. In addition, Asai and his collaborators have been proposing reaction-diffusion devices for image processing (Asai et al., 2005; Adamatzkey et al., 2005). They realized their algorithms by utilizing large-scale integrated circuits and applied them to realistic applications. Reaction-diffusion systems can be found in a variety of natural systems (Murray, 1989). The FitzHugh-Nagumo equations are typical reaction-diffusion equations; they simulate an information transmission phenomenon along a nerve axon (FitzHugh, 1961; Nagumo et al., 1962).
This chapter presents a class of algorithms for typical image processing and vision computing such as edge detection, grouping and stereo disparity detection, all of which are fundamental functions needed for vision systems. The algorithms utilize reaction-diffusion equations, so we call this class of algorithms "reaction-diffusion algorithms". In particular, we utilize the FitzHugh-Nagumo type reaction-diffusion equations, since we are interested in biological systems and also in the human early visual processing mechanism. Previous algorithms, such as those proposed by Marr et al., utilize the Gaussian filter or a diffusion equation. In contrast to those algorithms, our reaction-diffusion algorithm utilizes two diffusion equations coupled with non-linear reaction terms. Under a certain ratio of the two diffusion coefficients, the non-linear reaction terms play an important role in solving the problem of unexpected blurring caused by simple diffusion-based algorithms.

2. Reaction-Diffusion System

The reaction-diffusion system with the two variables (u,v) consists of

$$\partial_t u = D_u \nabla^2 u + f(u,v)\,, \quad \partial_t v = D_v \nabla^2 v + g(u,v)\,. \tag{1}$$

The operator ∂_t denotes the temporal partial derivative $\partial/\partial t$. The Laplacian operator ∇^2 denotes $\partial^2/\partial x^2+\partial^2/\partial y^2$ in the two-dimensional coordinate system of (x,y); D_u is the diffusion coefficient of the variable u and D_v is that of v. The functions $f(u,v)$ and $g(u,v)$ are reaction terms, which depend on particular phenomena. The reaction terms of the FitzHugh-Nagumo equations (FitzHugh, 1961; Nagumo et al., 1962) are

$$f(u,v) = \frac{1}{\varepsilon}\left[u(u-a)(1-u)-v\right], \quad g(u,v) = u - bv\,, \tag{2}$$

where a and b are constants and ε is a positive small constant ($0<\varepsilon<<1$).
Let us focus on the set of the ordinary differential equations $\mathrm{d}u/\mathrm{d}t=[u(u-a)(1-u)-v]/\varepsilon$ and $\mathrm{d}v/\mathrm{d}t=u-bv$ in order to understand the basic behaviour of the FitzHugh-Nagumo equations. The set of the equations has two different types of system behaviour, the mono-stable system and the bi-stable one, depending on the parameter values of a and b. For example, when a=0.25 and b=1.0, the system becomes mono-stable. In this case, any solution starting from any point on a phase plot finally converges to the stable point A, as time proceeds [see Figs. 1(a) and 1(b)]. When a=0.25 and b=10, solutions converge to either of the two stable points A or C, and that is the bi-stable system [see Figs. 1(a) and 1(c)]. The variable u is an

activator, and the variable v is an inhibitor. When $u>a$ and $v=0$ at an initial state, because of $du/dt>0$, the variable u increases spontaneously; this is the self-activation process. After u reaches 1.0, the variable v also begins to increase. The increasing process of the variable v inhibits the variable u from increasing; this is the self-inhibition process. When $u<a$ at an initial state, a solution directory converges to the stable point A. When starting from the initial condition of $v=0$, the system works as a time-dependent threshold function, in which the parameter a is its threshold value. Therefore, a set of solutions (u,v) traces a trajectory indicated by arrows in the mono-stable system. In the bi-stable system, a set of solutions starting from $u>a$ and $v=0$ remains at the stable equilibrium point C.

Let us return to the full reaction-diffusion system. When the two diffusion coefficients of the activator variable u and the inhibitor variable v are in the condition of $D_u>D_v$, the reaction-diffusion system self-organizes the temporally evolving spatial pattern which propagates in space. However, when the two diffusion coefficients are in the condition of $D_u<<D_v$, the system self-organizes a static pattern (Turing, 1952; Kondo & Asai, 1995). By choosing appropriate parameter values and finite differences for the discrete version of the FitzHugh-Nagumo type reaction-diffusion equations under $D_u<<D_v$, we obtain spatial static patterns (Ebihara et al., 2003a; Nomura et al., 2003). Figure 2 shows numerical results obtained by the reaction-diffusion system of Eqs. (1) and (2), in which an initial condition for u has the binary digit of 0 or 1 randomly distributed in the centre part of the one-dimensional space x. The mono-stable system self-organizes the two impulses standing at the edge points; that is, the impulses divide the one-dimensional space into the centre part and the remaining flat parts. The bi-stable system also divides the space into such parts. These results show that the reaction-diffusion system has the ability to detect edge points and segments from the binary data. Note that the spatial distributions shown in Figs. 2(b) and 2(c) are not transient but almost static.

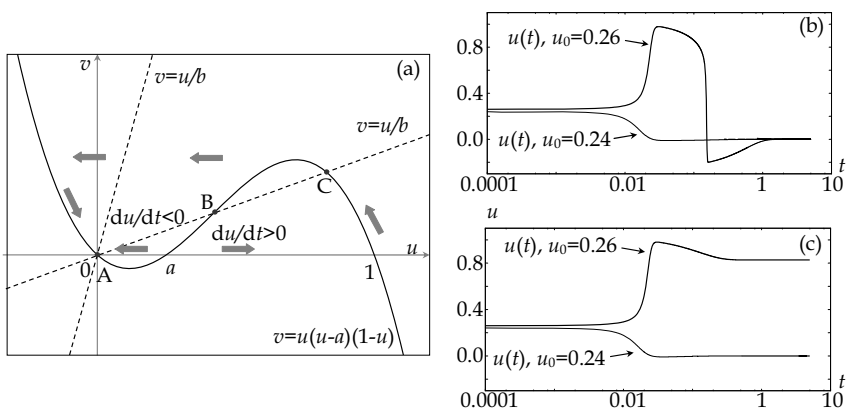


Figure 1. System behaviour of the FitzHugh-Nagumo equations $du/dt=[u(u-a)(1-u)-v]/\varepsilon$ and $dv/dt=u-bv$. (a) Phase plot for the equations. A and C are stable equilibrium points; B is an unstable point. (b) Temporal development of $u(t)$ for the mono-stable system (a=0.25, b=1.0). (c) Temporal development of $u(t)$ for the bi-stable system (a=0.25, b=10). Both of (b) and (c) show how the solutions starting from the two different initial conditions (u_0=0.24 and u_0=0.26) temporally change; the initial condition for $v(t)$ is zero and $\varepsilon=10^{-3}$ for both.

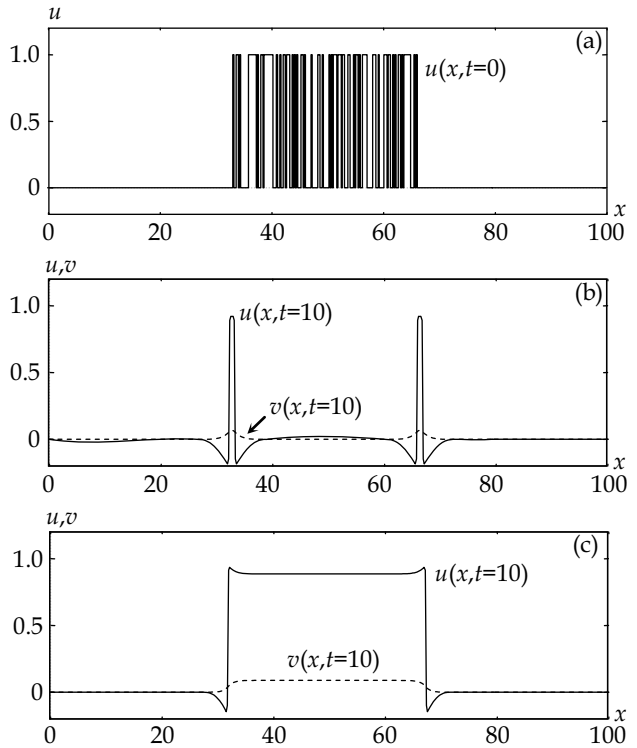


Figure 2. One-dimensional numerical results of the FitzHugh-Nagumo type reaction-diffusion equations of Eqs. (1) and (2). (a) Initial condition of $u(x,t=0)$; (b) spatial distributions of $u(x,t=10)$ and $v(x,t=10)$ for the mono-stable system with a=0.01 and b=1.0; (c) those for the bi-stable system with a=0.01 and b=10. The initial conditions for v in both (b) and (c) are $v(x,t=0)=0$. The other parameter values are D_u=1.0, D_v=10 and $\varepsilon=1.0\times10^{-3}$.

3. Previous Algorithms

Marr and his collaborators proposed algorithms for edge detection and stereo disparity detection. This section presents the previously proposed algorithms by Marr et al. and a more recently developed algorithm for stereo disparity detection.

3.1 Edge detection

Marr and Hildreth proposed an edge detection algorithm (Marr & Hildreth, 1980) in which edge points were generally defined as those having a high spatial gradient in image brightness distribution. Their algorithm consists of the following three steps to detect edge points from an image brightness distribution function denoted by $I(x)$ in the one-dimensional space x. The first step is to compute the convolution of the image function and the Gaussian function as follows:

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{3}$$

The parameter σ is the constant representing spatial spread. The convolution of the image function and the Gaussian function reduces noise contained in the image function. Natural images have noisy signals, which could produce pseudo edge points. Therefore, before detecting edge points, by applying a Gaussian filter to the image function, we obtain its smoothed function: $G*I(x)$ [see Figs. 3(a) and 3(b)], in which the symbol * denotes the convolution operator. The first-order derivative operator ∇ applied to $G*I(x)$ provides the distribution shown in Fig. 3(c). The second-order derivative operator ∇^2 for $G*I(x)$ provides the spatial distribution that is across the zero level at the edge point. That is, the zero-crossing point corresponds to the edge point. Thus, we can detect edge points by finding zero-crossing points in the distribution of $\nabla^2(G*I)$. This is the well-known edge detection algorithm that utilizes the 'Laplacian of Gaussian' (LoG) filter.

Another edge detection algorithm utilizes two Gaussian filters with different space constants. One of the filters has a small space constant denoted by σ_e, and the other has a large space constant denoted by σ_i. We can detect edge points by finding zero-crossing points in the output of the filter consisting of the difference of the two Gaussian filters (DOG). The DOG filter is expressed as

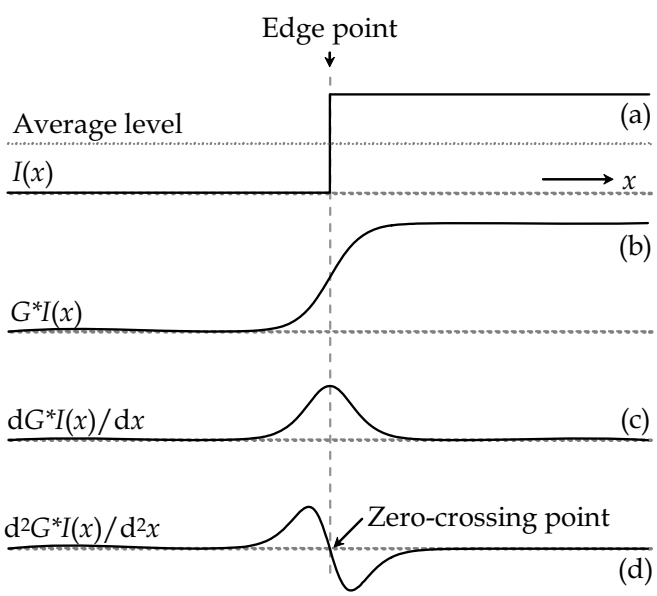


Figure 3. Edge detection algorithm proposed by Marr and Hildreth in the one-dimensional space x. (a) Image brightness distribution function $I(x)$ and its average level on brightness; (b) output of the Gaussian filter applied to $I(x)$; (c) first-order derivative of $G*I(x)$; (d) second-order derivative of $G*I(x)$. The zero-crossing point in the second derivative (d) corresponds to the edge point of the original $I(x)$.

$$\mathrm{DOG} = G(x;\sigma_e) - G(x;\sigma_i)\ . \tag{4}$$

Marr and Hildreth pointed out that the constant σ_i is larger than σ_e and the optimal ratio of the two constants (σ_e/σ_i) is 1.6 or 1.7.
Let us consider the following simple diffusion equation having the spatial and temporal distribution function $u(x,t)$ and the diffusion coefficient D_u :

$$\partial_t u = D_u \nabla^2 u\ . \tag{5}$$

When the diffusion equation has the initial condition of a spatial function $u_0(x)$, the next convolution of the function $u_0(x)$ and the Gaussian function $G_t(x,t;D_u)$ becomes the solution of the diffusion equation at time t,

$$u(x,t) = u_0(x) * G_t(x,t;D_u)\ , \tag{6}$$

where

$$G_t(x,t;D_u) = \frac{1}{2\sqrt{\pi D_u t}} \exp\left(-\frac{x^2}{4D_u t}\right). \tag{7}$$

Therefore, when the diffusion equation has the initial condition of an image function $I(x)$, it provides the solution equivalent to the Gaussian filter output for the image function. If an image pool stores the solution $u(x,t)$ of the diffusion equation during a short time Δt, edge points are detected from zero-crossing points in $[u(x,t\text{-}\Delta t)\text{-}u(x,t)]$ (Sunayama et al., 2000) derived from the simple diffusion equation of Eq. (5), since the space constant of the Gaussian function described in Eq. (7) depends on time t. Another algorithm for edge detection utilizes the next two simple diffusion equations having the two variables (u,v) and the initial conditions of $u(x,t\text{=}0)\text{=}v(x,t\text{=}0)\text{=}I(x)$. Under the condition of $D_u<D_v$, zero-crossing points in the difference distribution $(u\text{-}v)$ correspond to edge points.

$$\partial_t u = D_u \nabla^2 u\ ,\ \partial_t v = D_v \nabla^2 v \tag{8}$$

3.2 Stereo disparity detection

Figure 4 shows the arrangement of stereo cameras and an object in a three-dimensional world, and a basic idea for stereo disparity detection. The left camera projects the object point onto a position (x_L,y) on its image plane $I_L(x,y)$; the right one does it onto (x_R,y) on $I_R(x,y)$. The difference between the two positions is the stereo disparity $d\text{=}x_L\text{-}x_R$, which can be used to obtain the depth of the object (Gonzalez & Woods, 1992). If we can find the correspondence between the two points (x_R,y) and (x_L,y) from only the two image brightness distribution functions $I_L(x,y)$ and $I_R(x,y)$, we can obtain the stereo disparity, that is, the depth of the object point. To find the stereo correspondence, we overlap the two image distributions $I_L(x,y)$ and $I_R(x,y)$ at every possible disparity level d in $\Psi_d=\{d_0,d_1,\ldots,d_{N\text{-}1}\}$, in which N denotes the number of possible disparity levels. If the object has a disparity level d, a cross-correlation map $C_d(x,y)$ computed for $I_L(x,y)$ and $I_R(x+d,y)$ has a high correlation value nearly equal to 1.0 at the object position. By finding the highest value of $C_d(x,y)$ for all of the possible disparity levels, we can obtain a disparity map.

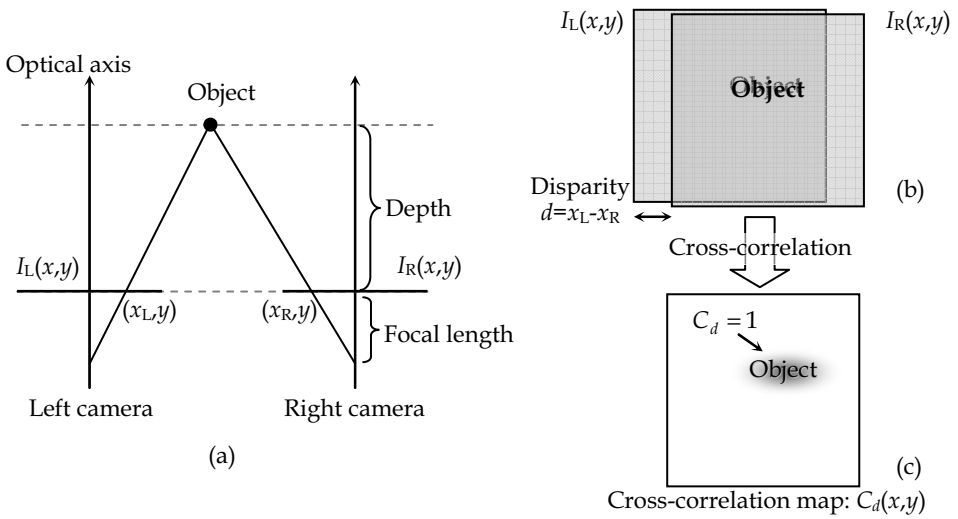


Figure 4. Stereo vision geometry and a cross-correlation map. (a) Stereo cameras and an object in a three-dimensional world. An object point is projected onto the image plane $I_L(x,y)$ of the left camera and onto the image plane $I_R(x,y)$ of the right one. (b) Stereo images overlapped with a stereo disparity level d. (c) Cross-correlation map $C_d(x,y)$ obtained between the stereo images with a disparity level d. When $d=x_L-x_R$, C_d becomes 1.

Following the above idea of detecting a stereo disparity map, we first compute a cross-correlation map $C_d(x,y)$ between stereo images. For a binary stereo image pair such as the random-dot stereograms (Julesz, 1960), a simple logic operation provides the cross-correlation map as

$$C_d(x,y) = \neg[I_L(x,y) \oplus I_R(x+d,y)], \tag{9}$$

where $\neg[\cdot\oplus\cdot]$ is the XNOR logic operation that gives 1 for a matched pair and 0 for an unmatched one (Nomura et al., 1999). For real stereo images, the normalized cross-correlation function computed in a spatial local area L_S,

$$C_d(x,y) = \frac{1}{\max\{s_L, s_R\}} \sum_{(x',y')\in L_S} \left[I_L(x+x',y+y') - \overline{I_L(x,y)}\right] \times \left[I_R(x+x'+d,y+y') - \overline{I_R(x+d,y)}\right] \tag{10}$$

provides similarity between the stereo images, where s_L denotes the standard deviation of I_L in L_S surrounding the point (x,y) and s_R denotes the standard deviation of I_R in L_S surrounding $(x+d,y)$; $\overline{I_L}$ and $\overline{I_R}$ are the averages of I_L and I_R in L_S as shown in

$$s(x,y) = \left\{ \frac{1}{N_{L_S}} \sum_{(x',y')\in L_S} \left[I(x+x',y+y') - \overline{I(x,y)}\right]^2 \right\}^{1/2}, \quad \overline{I(x,y)} = \frac{1}{N_{L_S}} \sum_{(x',y')\in L_S} I(x+x',y+y'), \tag{11}$$

where N_{L_S} is the number of points in L_S. The cross-correlation map $C_d(x,y)$ is provided for use in the next main step of stereo disparity detection. Several stereo algorithms utilize other

types of similarity measures such as the sum of absolute differences (SAD) between stereo images (Brown et al., 2003).
It is difficult to solve the stereo correspondence problem using only the cross-correlation map. The cross-correlation value C_d becomes 1 for a matched pair between stereo images. However, in real situations, images have many similar brightness patterns, which cause many miss matched pairs in $C_d(x,y)$. Therefore, there is much uncertainty in finding correct match pairs in the cross-correlation map. To solve the uncertainty and detect a reliable stereo disparity map, we need additional information or constraint, as described next.

Marr and Poggio proposed a cooperative stereo algorithm. They imposed two important constraints on stereo disparity distribution, a continuity constraint and a uniqueness constraint. The continuity constraint states that the stereo disparity distribution varies smoothly over a stereo disparity map or in a local spatial area called "local support". This is generally true except for object boundaries. The other uniqueness constraint states that a point on a stereo disparity map has only one disparity level. This is also true except for a transparent object. According to these two constraints and the biologically motivated idea of a cell network, they formulated the next update function for the cell state $S_d^t(x,y)$ of the disparity level d and the position (x,y) at the t-th iteration step, as follows:

$$S_d^{t+1}(x,y) = \sigma\left(\sum_{(x',y',d')\in\Omega_e} S_{d+d'}^t(x+x',y+y') - \lambda \sum_{(x',y',d')\in\Omega_i(x,y,d)} S_{d'}^t(x',y') + C_d(x,y), T \right). \tag{12}$$

In Eq. (12), $\sigma(S,T)$ denotes the threshold function, in which the parameter T is the threshold value for S. When $S<T$, $\sigma(S,T)$ provides zero; when $S\geq T$, it provides 1. The symbol Ω_e denotes the spatial local support area for the continuity constraint, and Ω_i denotes the inhibition area for the uniqueness constraint [see Fig. 1 in the article (Zitnick and Kanade, 2000)]. The parameter λ is a positive inhibition constant. After iterations needed for convergence of the update function of Eq. (12), we obtain the disparity map $M(x,y,t)$ at the t-th step by finding the maximum value of $S_d^t(x,y)$ for all of the possible disparity levels at a particular position (x,y),

$$M(x,y,t) = \arg\max_{d\in\Psi_d} S_d^t(x,y)\,. \tag{13}$$

The original cooperative algorithm works well for random-dot stereograms. However, the algorithm does not work for real stereo images.
Zitnick and Kanade improved the cooperative algorithm (Zitnick & Kanade, 2000). Their algorithm achieves good performance not only for random-dot stereograms but also for real stereo images, compared with the original cooperative algorithm. In addition, their algorithm can detect an occlusion area, which was not taken into account by the original cooperative algorithm. The update function proposed by Zitnick and Kanade is

$$S_d^{t+1}(x,y) = C_d(x,y) \times \left[\frac{R_d^t(x,y)}{\sum_{(x'',y'',d'')\in\Omega_i(x,y,d)} R_{d''}^t(x'',y'')} \right]^{\alpha}, \tag{14}$$

where α is a constant for convergence of the update function, and R_d^t is

$$R_d^t(x,y) = \sum_{(x',y',d')\in\Omega_e} S_{d+d'}^t(x+x',y+y') . \tag{15}$$

The algorithm finally detects the disparity map, including the occlusion area, by

$$M(x,y,t) = \begin{cases} \arg\max_{d\in\Psi_d} S_d^t(x,y) & \text{if} \quad \max_{d\in\Psi_d} S_d^t \geq T, \\ d_\infty & \text{otherwise.} \end{cases} \tag{16}$$

If the maximum value of S_d^t at the point (x,y) is less than T, the algorithm classifies the point (x,y) as the occlusion denoted by d_∞. The threshold value $T\leq 0$ switches off the algorithm of the occlusion area detection.

4. Reaction-Diffusion Algorithm

This section presents the reaction-diffusion algorithm for edge detection, grouping and stereo disparity detection by means of the FitzHugh-Nagumo type reaction-diffusion system presented in section 2. The reaction-diffusion system consists of partial-differential equations; thus, this section additionally presents numerical schemes required for the computation of the equations.

4.1 Edge detection

The one-dimensional numerical result of Fig. 2(b) lets us recognize that the FitzHugh-Nagumo type reaction-diffusion system has the ability of edge detection. It self-organizes impulses at edge points, if the initial condition is binary data. To utilize the edge detection algorithm on multi-valued image, we modify the FitzHugh-Nagumo equations. Let us recall the situation of Fig. 3 showing the original image brightness distribution $I(x,y)$ and its average brightness level. The image brightness distribution $I(x,y)$ is across its average level at the edge position. In Eq. (2), the parameter value a is the threshold value for the initial condition, as stated in the description for the ordinary differential equation and also as shown in Fig. 1. Therefore, when substituting the average level of $I(x,y)$ for the parameter value a of the FitzHugh-Nagumo equations, we can expect to realize the edge detection function. A simple diffusion equation provides a local average value of an initial condition and its local area is spreading as time proceeds. Thus, we estimate the average level or the threshold level of a with an additional diffusion equation starting from the initial condition of $I(x,y)$. The overall set of equations for edge detection in the reaction-diffusion algorithm is the following:

$$\partial_t u = D_u \nabla^2 u + f(u,v,a) , \quad \partial_t v = D_v \nabla^2 v + g(u,v) , \quad \partial_t a = D_a \nabla^2 a , \tag{17}$$

$$f(u,v,a) = \frac{1}{\varepsilon}\left[u(u-a)(1-u) - v\right], \quad g(u,v) = u - bv . \tag{18}$$

Note that Eq. (17) handles the parameter a as a spatial and temporal variable $a=a(x,y,t)$, as opposed to the constant parameter a of the original FitzHugh-Nagumo equations. The

diffusion coefficients D_u, D_v and D_a should satisfy the relation $D_u << D_v << D_a$. The Turing-like condition $D_u << D_v$ is from section 2, and the condition $D_v << D_a$ is for the computation of the local average level. Initial conditions for (u,v,a) are given as

$$u(x,y,t=0) = a(x,y,t=0) = a_0 \times I(x,y)\ ,\ v(x,y,t=0) = 0\ , \tag{19}$$

where $I(x,y)$ is a normalized image brightness distribution ranging from 0 to 1; a_0 is a constant. The Neumann boundary condition governs the four sides of the rectangular image region of u, v and a such as

$$\partial_x u\big|_{\text{Left,Right}} = 0\ ,\ \partial_y u\big|_{\text{Top,Bottom}} = 0\ . \tag{20}$$

4.2 Grouping

The human vision system has a grouping mechanism for processing visual stimuli. For example, when the system is exposed to the visual stimulus of an image that consists of several different features such as orientation, it will perceive several groups corresponding to particular orientation features (Beck, 1966). That is, for example, the human vision system can reconstruct the group map of Fig. 5(a) from the visual stimulus of Fig. 5(b). This is the grouping mechanism. We believe that the grouping mechanism underlies several human visual functions such as stereo disparity detection.

In accordance with the reaction-diffusion algorithm, we present a model that can reconstruct a grouping map from a visual stimulus. Figure 6 shows the overall flow diagram of visual processing for the grouping mechanism (Nomura et al., 2004). During the first stage, several orientation-selective filters detect the orientation feature distributions $s_n(x,y)$ from the input image $I(x,y)$ of a visual stimulus. For example, Fig. 7 shows the outputs s_n (n=0,1,2) of the three different orientation-selective filters applied to the input image of Fig. 5(b). Then, during the next stage, the distributions $s_n(x,y)$ are fed to the multi-sets of reaction-diffusion equations; each set has the two variables (u_n,v_n) and is slightly modified from the original FitzHugh-Nagumo equations. As time proceeds, the multi-sets of equations spontaneously self-organize groups of orientation features. Finally, the algorithm reconstructs a group map from the solutions u_n. Note that the multi-sets of the reaction-diffusion equations are mutually and inhibitedly linked through the activator variables u_n.

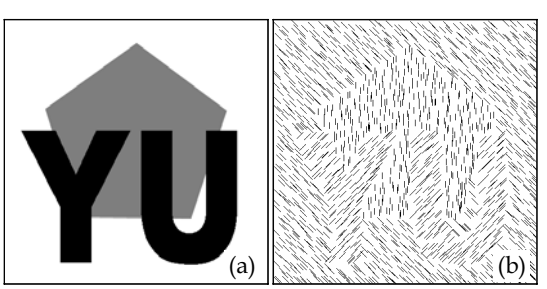


Figure 5. Group map and visual stimulus for the explanation of the grouping mechanism in the human vision system. (a) Original group map with three groups. (b) Visual stimulus having the three different features of line orientation. The image size is 400×400 pixels.

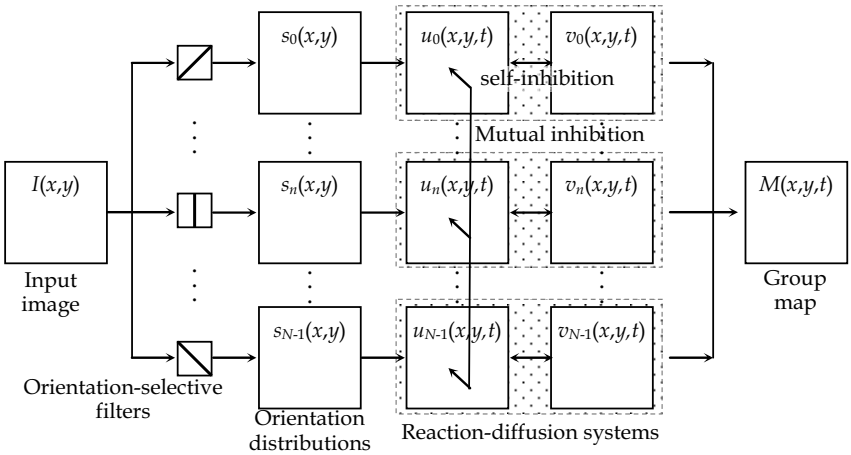
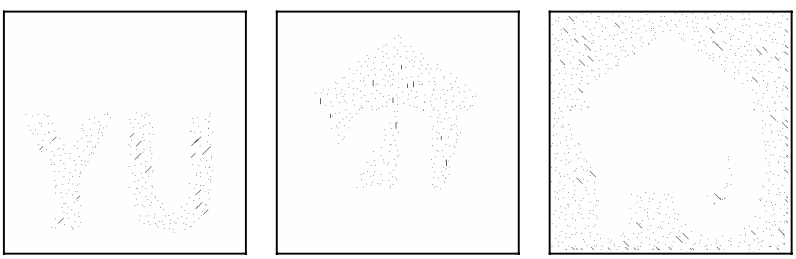


Figure 6. Flow diagram of the grouping mechanism. From the input image $I(x,y)$, orientation-selective filters provide feature distributions $s_n(x,y)$ for $n \in \{0,1,\ldots,N\text{-}1\}$, in which N denotes the number of groups. The distributions are fed to the multi-sets of the reaction-diffusion equations having the two variables (u_n,v_n). Integration of u_n provides a group map $M(x,y,t)$.



(a) (b) (c)

Figure 7. Outputs of three orientation-selection filters. Black dots indicate the existence of orientation features of (a) $\pi/4$, (b) $2\pi/4$ and (c) $3\pi/4$. The filters are realized with a matching procedure between an input image and a template pattern of an oriented short line.

To model the grouping mechanism we must consider two important constraints. One of them is that a particular point on an image is a member of only one group and is not classified into two or more groups; that is the uniqueness constraint. The other constraint is that spatial adjacent points are likely to be members of the same group; that is the continuity constraint. A particular point except for boundary areas of groups should satisfy these two constraints. By recalling that the parameter a is the threshold value, we formulate the set of equations governing the n-th group as follows:

$$\partial_t u_n = D_u \nabla^2 u_n + f(u_n, v_n, U_n) + \mu s_n, \quad \partial_t v_n = D_v \nabla^2 v_n + g(u_n, v_n), \tag{21}$$

$$f(u_n, v_n, U_n) = \frac{1}{\varepsilon}\left[u_n(u_n - a(U_n))(1-u_n) - v_n\right], \quad g(u_n, v_n) = u_n - bv_n, \tag{22}$$

where μ is a constant. The original reaction term $f(u,v)$ has a constant parameter a. In contrast to this, the modified version of the reaction term in Eq. (22) depends on the state of another set U_n, as follows:

$$a(U_n) = a_0 + U_n\,,\ U_n = \max_{n' \in \{0,1,\cdots,n-1,n+1,\cdots,N-1\}} u_{n'}(x,y,t)\,, \tag{23}$$

where a_0 is a constant and N is the number of groups. Let us consider the situation in which the variable $u_{n'}(x,y)$ of the n'-th group becomes large. This represents that the point (x,y) is already classified as the n'-th group. In that case, the n-th set must be inhibited to have a high value of u_n. Equation (23) works as the mutual inhibition mechanism by increasing the threshold value of other sets, and the multi-sets of equations exclusively become the excited state having the high value of u_n. Thus, Eq. (23) realizes the uniqueness constraint. The continuity constraint is built into the reaction-diffusion system, since the system originally has the spatial information propagation effect for its adjacent area. After convergence, the final step is to reconstruct a group map $M(x,y,t)$ by finding the maximum value of u_n at a particular point at time t as follows:

$$M(x,y,t) = \arg\max_{n \in \{0,1,\cdots,N-1\}} u_n(x,y,t)\,. \tag{24}$$

4.3 Stereo disparity detection

The previous cooperative algorithm proposed by Marr and Poggio has the uniqueness constraint and the continuity constraint, both of which are very similar to the constraints made by the reaction-diffusion algorithm modelling the grouping mechanism. Thus, we can expect that the reaction-diffusion algorithm described with Eqs. (21) and (22) is also applicable to the stereo disparity detection; a cross-correlation map $C_d(x,y)$ obtained by Eq. (9) or Eq. (10) substitutes for $s_n(x,y)$ in Eq. (21), and the disparity level d corresponds to the number n (Nomura et al., 2005). The local support area in the stereo disparity detection is not only over space but also across the disparity direction, as proposed by Zitnick and Kanade. Thus, it is necessary to modify the function $a(U_d)$ in Eq. (23) representing the uniqueness constraint, by taking into account the distance between the current disparity level d and the level having the largest value of $u_{d'}$ in the inhibition area Ω_i. One possible formulation having the two constants of a_0 and a_1, and the switching function tanh(·) is

$$a(U_d) = a_0 + U_d \times \frac{1}{2}\left[1 + \tanh\left(|d| - a_1\right)\right],\ U_d = \max_{(x',y',d') \in \Omega_i} u_{d'}(x',y',t)\,,\ |d| = \left| d - \arg\max_{(x',y',d') \in \Omega_i} u_{d'}(x',y',t) \right|. \tag{25}$$

Finally, Eq. (24) provides a stereo disparity map $M(x,y,t)$, in the same manner.

4.4 Numerical computation for reaction-diffusion equations

The realization of the reaction-diffusion algorithm on a computer system requires numerical computation of partial differential equations. The finite difference method is applicable to the computation. For example, the partial derivatives $\partial_t u$, $\partial_x u$ and $\partial_y u$ at (x,y,t) are approximately evaluated with the finite differences of δt, δx and δy in time and two-dimensional space as

$$\partial_t u(x,y,t) \cong \frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\delta t}, \tag{26}$$

$$\partial_x u(x,y,t) \cong h\frac{u_{i+1,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i-1,j}^{k+1}}{\delta x^2} + (1-h)\frac{u_{i+1,j}^{k} - 2u_{i,j}^{k} + u_{i-1,j}^{k}}{\delta x^2}, \tag{27}$$

$$\partial_y u(x,y,t) \cong h\frac{u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}}{\delta y^2} + (1-h)\frac{u_{i,j+1}^{k} - 2u_{i,j}^{k} + u_{i,j-1}^{k}}{\delta y^2}, \tag{28}$$

where $u_{i,j}^{k}$ denotes $u(i\delta x, j\delta y, k\delta t)$ in the discrete coordinate system. The first terms on the right side of Eqs. (27) and (28) are the implicit terms evaluated at (k+1), and the second terms are the explicit terms evaluated at k. The parameter h denotes a ratio between the explicit term and the implicit one in each of the equations; when h=0.5, the system of Eqs. (26)-(28) becomes the Crank-Nicolson scheme (Press et al., 1988). Thus, the discrete version of Eq. (1) becomes

$$\begin{aligned} &-hD_{uy}u_{i,j-1}^{k+1} - hD_{ux}u_{i-1,j}^{k+1} + \left[1 + 2h\left(D_{ux} + D_{uy}\right)\right]u_{i,j}^{k+1} - hD_{ux}u_{i+1,j}^{k+1} - hD_{uy}u_{i,j+1}^{k+1} \\ &\quad = (1-h)D_{uy}u_{i,j-1}^{k} + (1-h)D_{ux}u_{i-1,j}^{k} + \left[1 + 2(1-h)\left(D_{ux} + D_{uy}\right)\right]u_{i,j}^{k} \\ &\qquad + (1-h)D_{ux}u_{i+1,j}^{k} + (1-h)D_{uy}u_{i,j+1}^{k} + \delta t f(u_{i,j}^{k}, v_{i,j}^{k}), \end{aligned} \tag{29}$$

where

$$D_{ux} = D_u \delta t / \delta x^2, \; D_{uy} = D_u \delta t / \delta y^2. \tag{30}$$

By applying Eq. (29) to a particular point on an image, we obtain a set of linear equations. For example, the Gauss-Seidel method iteratively solves the set of equations (Press et al., 1988).

The previous study done by the authors and their collaborators implies that the choices of the spatial finite differences are very important, and it suggested that rather large finite differences would be better for the edge detection algorithm (Ebihara et al., 2003a).

5. Experimental Results

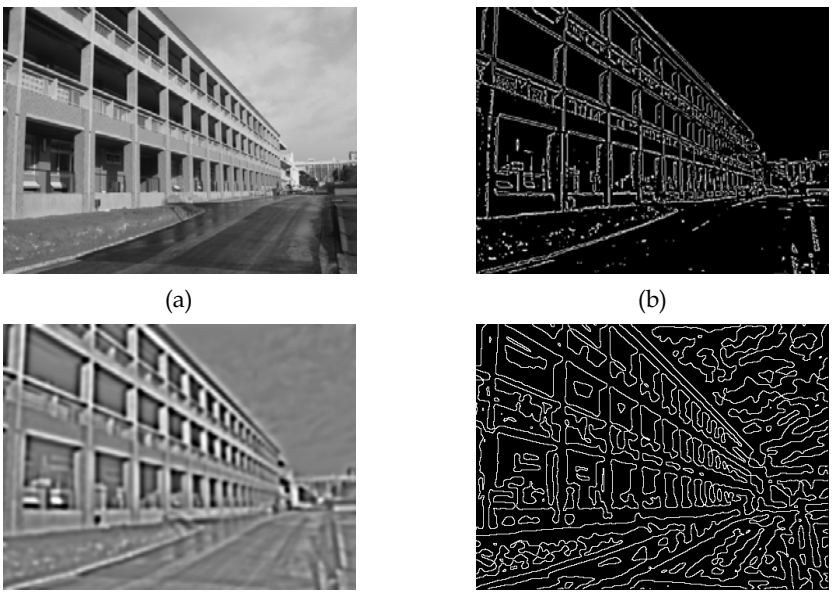
This section presents experimental results for the performance comparison of the reaction-diffusion algorithm and other competitive algorithms for edge detection, grouping and stereo disparity detection. Table 1 summarizes the algorithms, their parameter values utilized here and references to the results.

	Algorithm and model equation(s)	Parameter values	Results
Edge detection	Reaction-diffusion algorithm: Eqs. (17)-(19)	$\delta x=\delta y=1/2$, $\delta t=1/1000$, $D_u=1.0$, $D_v=5.0$, $D_a=100$, $a_0=0.25$, $b=1.0$, $\varepsilon=10^{-3}$	Fig. 8
	DOG filter realized with two diffusion equations: Eq. (8)	$\delta x=\delta y=1/2$, $\delta t=1/1000$, $D_u=1.0$, $D_v=2.56$	
Grouping	Reaction-diffusion algorithm: Eqs. (21)-(24)	$\delta x=\delta y=1/10$, $\delta t=1/1000$, $D_u=1.0$, $D_v=3.0$ or 1.0, $a_0=0.15$, $b=10$, $\varepsilon=10^{-2}$, $\mu=100$	Figs. 9,10 Table 2
	Single diffusion algorithm: Eqs. (21), (24) with $f(u_n,v_n,U_n)=0$	$\delta x=\delta y=1/10$, $\delta t=1/1000$, $D_u=1.0$, $\mu=100$	
Stereo disparity detection	Reaction-diffusion algorithm: Eqs. (21), (22), (24), (25)	$\delta x=\delta y=1/5$, $\delta t=1/100$, $D_u=1.0$, $D_v=3.0$, $a_0=0.13$, $a_1=1.5$, $b=10$, $\varepsilon=10^{-2}$, $\mu=3.0$	Figs. 11-13 Table 3
	Cooperative algorithm: Eqs. (14)-(16)	$\alpha=2.0$, $T=0$, $\Omega_e=5\times5\times3$, $C_d=0.08$ if $C_d<0.08$	

Table 1. Algorithms, their parameter values utilized in the experiments and references to the figures and tables showing their results. The ratio h needed for the finite difference method was fixed at $h=0.5$. For stereo disparity detection, both the reaction-diffusion algorithm and the cooperative algorithm utilize a cross-correlation map $C_d(x,y)$ evaluated by Eq. (10), in which the spatial local area L_S consists of the target point and its 4 nearest points, that is, $L_S=\{(x,y)\,|\,(0,0),(1,0),(0,1),(-1,0),(0,-1)\}$. Neither of the stereo algorithms has sub-pixel accuracy on disparity. The authors realized the computer programs of all the algorithms including the competitive ones by themselves.

5.1 Edge detection

Figure 8 shows edge detection results obtained for an image of the outdoor scene shown in Fig. 8(a). Figure 8(b) shows the distribution $u(x,y,t=10)$ obtained by the reaction-diffusion algorithm; this distribution directly expresses edge points. Figure 8(c) shows the difference of the two solutions u and v governed by the two diffusion equations having small and large diffusion coefficients ($D_u<D_v$) ; the difference corresponds to that of the DOG filter proposed by Marr and Hildreth [see Eq. (8) and section 3]. The zero-crossing points in Fig. 8(c) correspond to the edge points as shown in Fig. 8(d). The reaction-diffusion algorithm detects and preserves sharp corners, in contrast to the DOG filter. This is the main feature of the reaction-diffusion algorithm applied to edge detection.



(a) (b)

(c) (d)

Figure 8. Results of edge detection for a real image. (a) Original image, 450×340 pixels and 256 brightness levels. (b) Solution $u(x,y,t=10)$ obtained by the reaction-diffusion algorithm. (c) Difference of two solutions $[u(x,y,t=1.0)-v(x,y,t=1.0)]$ in Eq. (8). (d) Zero-crossing points obtained from (c). White lines and dots correspond to edge points in (b) and (d). See Table 1 for the algorithms and parameter values utilized here.

5.2 Grouping

The reaction-diffusion algorithm and its competitive single diffusion algorithm were applied to the filter outputs in Fig. 7 derived from Fig. 5(b). Figure 9(a) shows the result of the reaction-diffusion algorithm having the large inhibitory diffusion coefficient D_v=3.0; Fig. 9(b) shows the result with D_v=1.0 being equal to D_u. For comparison, Fig. 9(c) shows the result obtained by the single diffusion algorithm, which is derived from the reaction-diffusion algorithm with $f(u_n,v_n,U_n)=0$ in Eq. (21). The next error measure evaluates the difference between the true group map $M_t(x,y)$ and an obtained one $M_c(x,y,t)$:

$$E(t)=\frac{1}{N_F}\sum_{(x,y)\in F}\sigma\left(\left|M_t(x,y)-M_c(x,y,t)\right|,1.0\right)\times 100\ \ (\%) \tag{31}$$

The set F contains all of the points on an image plane, and N_F is the number of points in F. Figure 10 and Table 2 show the results of error evaluation for the algorithms. These results show the similar minimum errors (Table 2). The single diffusion algorithm achieves the minimum error at t=0.4; however, after that, the error is rapidly increasing monotonically. The reaction-diffusion algorithm with D_v=3.0 achieves the best evaluation of the minimum error at t=1.6. After that, the error is slightly increasing and finally it converges with good evaluation. The result using the reaction-diffusion algorithm with D_v=3.0 is better than that with D_v=1.0.

(a) (b) (c)

Figure 9. Results of grouping obtained by the reaction-diffusion algorithm with (a) D_v=3.0 at t=1.6, (b) D_v=1.0 at t=1.1 and (c) by the single diffusion algorithm at t=0.4. See Table 1 for the algorithms and parameter values utilized here. Figure 5 shows the original image of the visual stimulus, and Fig. 7 shows the outputs of orientation-selective filters.

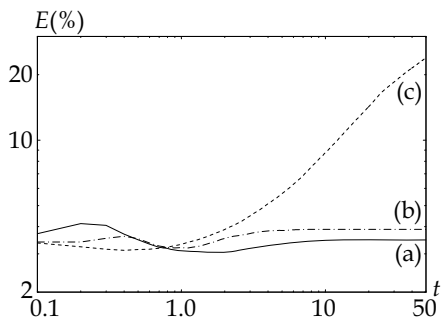


Figure 10. Temporal error changes evaluated for the results of the grouping process by the error measure $E(t)$ of Eq. (31). (a) Reaction-diffusion algorithm with D_v=3.0; (b) reaction-diffusion algorithm with D_v=1.0; (c) single diffusion algorithm. See Table 1 for the algorithms and parameter values. See Fig. 9 for the group maps at the minimum error.

Algorithm	Reaction-diffusion algorithm		Single diffusion algorithm
	D_v=3.0	D_v=1.0	
Minimum error	$E(t$=1.6)=3.05 (%)	$E(t$=1.1)=3.19 (%)	$E(t$=0.4)=3.12 (%)
Final error	$E(t$=50) =3.48 (%)	$E(t$=50) =3.89 (%)	$E(t$=50) =23.9 (%)

Table 2. Error comparison among the results of the reaction-diffusion algorithm and the single diffusion algorithm by the error measure $E(t)$ of Eq. (31). See Table 1 for the algorithms and their parameter values and Fig. 9 for the group maps at the minimum error.

5.3 Stereo disparity detection

The reaction-diffusion algorithm and the previous cooperative algorithm proposed by Zitnick and Kanade were applied to the well-known stereo images of TSUKUBA and VENUS for performance evaluation of stereo algorithms. Both the image pairs and their true disparity maps are available via the website http://www.middlebury.edu/stereo. Figures 11 and 12 show the stereo images, the true disparity map, a cross-correlation map and disparity maps obtained by the two algorithms.

Two kinds of error measures, E_{RMS} and E_{BMP} (Scharstein and Szeliski, 2002) evaluate the obtained disparity maps. The error measure E_{RMS} evaluates the root-mean-square error for an obtained stereo disparity map, as follows:

$$E_{\mathrm{RMS}}(t) = \left\{ \frac{1}{N_{F_{\neg\mathrm{O}}}} \sum_{(x,y)\in F_{\neg\mathrm{O}}} \left[M_{\mathrm{t}}(x,y) - M_{\mathrm{c}}(x,y,t) \right]^2 \right\}^{1/2}. \tag{32}$$

The set $F_{\neg\mathrm{O}}$ contains all of the points detected on an image plane except for the occlusion area and border; $N_{F_{\neg\mathrm{O}}}$ is the number of points in $F_{\neg\mathrm{O}}$. Thus, the error measure evaluates how much an obtained disparity map $M_{\mathrm{c}}(x,y,t)$ differs from the true one $M_{\mathrm{t}}(x,y)$. The error measure E_{BMP} evaluates the ratio of the number of correct match points to that of detected points $N_{F_{\neg\mathrm{O}}}$, as follows:

$$E_{\mathrm{BMP}}(t) = \frac{1}{N_{F_{\neg\mathrm{O}}}} \sum_{(x,y)\in F_{\neg\mathrm{O}}} \sigma\left(\left| M_{\mathrm{t}}(x,y) - M_{\mathrm{c}}(x,y,t) \right|, \delta d \right) \times 100 \ \ (\%). \tag{33}$$

The parameter δd denotes the threshold value for the judgement of bad match or a correct one; it was fixed at δd=1.0 pixel throughout the present experiments.

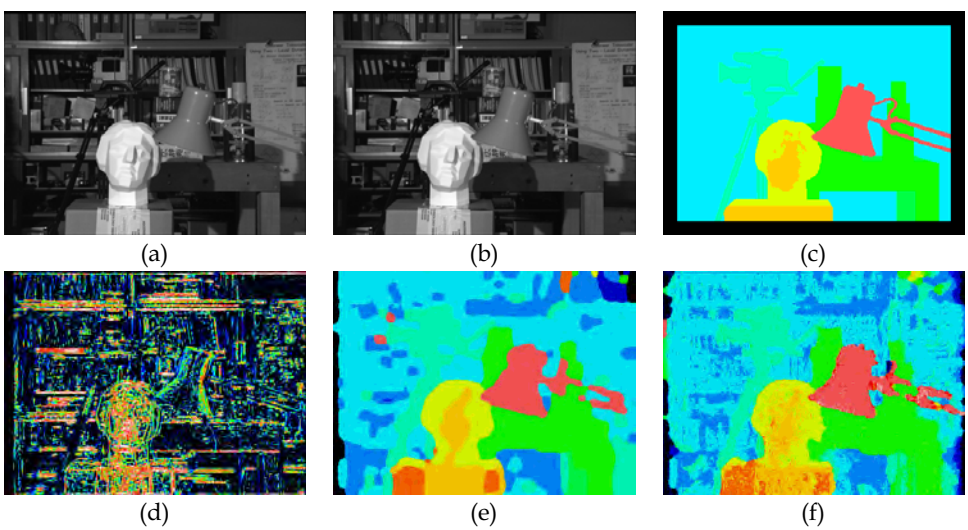


Figure 11. Stereo disparity detection for the image pair of TSUKUBA. (a) Left and (b) right images and (c) true disparity map $M_{\mathrm{t}}(x,y)$. The image size is 384×288 pixels, and possible disparity levels are Ψ_d={0,1,…,15} pixels. (d) Cross-correlation map $C_d(x,y)$ at the disparity level d=11 pixels. Disparity maps $M_{\mathrm{c}}(x,y,t)$ obtained by (e) the reaction-diffusion algorithm at t=50 and (f) the cooperative algorithm at t=100. See Table 1 for the algorithms and parameter values utilized here. The stereo image pair is available via the website http://www.middlebury.edu/stereo (Scharstein and Szeliski, 2002).

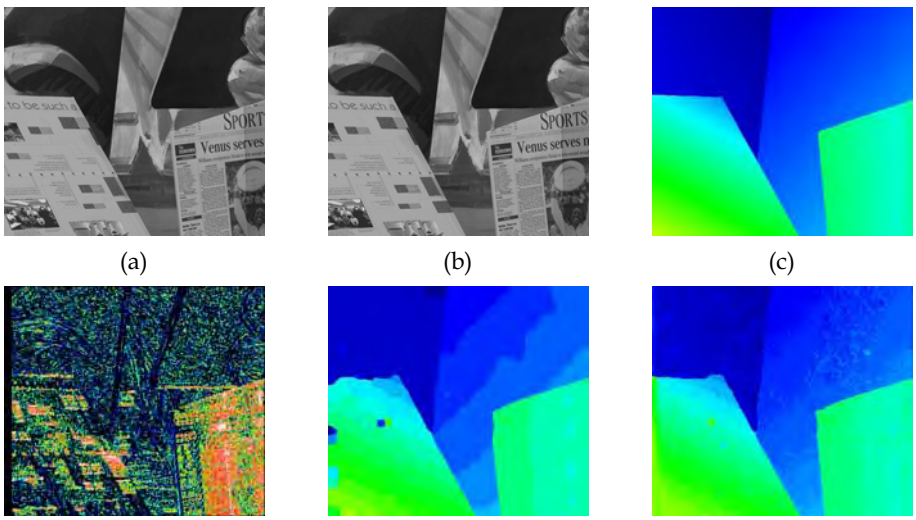


Figure 12. Stereo disparity detection for the image pair of VENUS. (a) Left and (b) right images and (c) true disparity map $M_t(x,y)$. The image size is 434×383 pixels, and possible disparity levels are Ψ_d={0,1,···,19} pixels. (d) Cross-correlation map $C_d(x,y)$ obtained at the disparity level d=12 pixels. Disparity maps $M_c(x,y,t)$ obtained by (e) the reaction-diffusion algorithm at t=50 and (f) the cooperative algorithm at t=100. See Table 1 for the algorithms and parameter values utilized here. The stereo image pair is available via the website http://www.middlebury.edu/stereo (Scharstein and Szeliski, 2002).

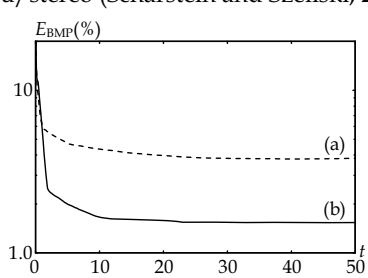


Figure 13. Temporal error changes evaluated for the disparity maps obtained by the reaction-diffusion algorithm. The bad-match-percentage error measure E_{BMP} of Eq. (33) was applied to the disparity maps detected from (a) TSUKUBA (Fig. 11) and (b) VENUS (Fig. 12).

Algorithm	Reaction-diffusion algorithm		Cooperative algorithm	
TSUKUBA	E_{RMS}=1.23 (pixel)	E_{BMP}=3.89 (%)	E_{RMS}=1.09 (pixel)	E_{BMP}=3.60 (%)
VENUS	E_{RMS}=0.677 (pixel)	E_{BMP}=1.54 (%)	E_{RMS}=0.691 (pixel)	E_{BMP}=4.32 (%)

Table 3. Performance comparison between the reaction-diffusion algorithm and the previous cooperative algorithm for the stereo image pairs of TSUKUBA and VENUS. See Table 1 for the algorithms and parameter values, and Figs. 11 and 12 for the disparity maps. See Eq. (32) for the error measure E_{RMS} and Eq. (33) for E_{BMP}.

Figure 13 shows the temporal changes of the bad-match-percentage error measure E_{BMP} for the results of the reaction-diffusion algorithm. The temporal changes show the convergence of the reaction-diffusion algorithm. In addition, for quantitative comparison, Table 3 shows the error measures evaluated for the results of the two algorithms. From these results, we see that the reaction-diffusion algorithm achieves good performance in the detection of stereo disparity.

6. Conclusion

This chapter presented the reaction-diffusion algorithm for vision systems. After a brief explanation of the reaction-diffusion system, we presented a class of algorithms for edge detection, grouping and stereo disparity detection by utilizing the FitzHugh-Nagumo type reaction-diffusion equations; all of the algorithms are necessary for the realization of vision systems. Previous algorithms, in particular, those proposed by Marr and his collaborators, utilize the Gaussian filter; the output of the filter is equivalent to the solution of the diffusion equation. In contrast to this, the reaction-diffusion algorithm has non-linear reaction terms coupled with diffusion equations. The non-linearity of the algorithm and the Turing-like condition can help to achieve good performance in edge detection, grouping and stereo disparity detection. Recently, the authors found a key mechanism in the stochastic resonance for performance improvement (Ebihara et al., 2003b). Thus, we conclude this chapter by noticing that further performance improvement will be possible with the use of the stochastic resonance in the reaction-diffusion algorithm.

7. References

Adamatzky, A.; Costello, B. D. L. & Asai, T. (2005). *Reaction-Diffusion Computers*, Elsevier, Amsterdam

Asai, T.; Costello, B. D. L. & Adamatzky, A. (2005). Silicon implementation of a chemical reaction-diffusion processor for computation of Voronoi diagram. *International Journal of Bifurcation and Chaos*, Vol. 15, pp. 3307-3320

Beck, J. (1966). Effect of orientation and of shape similarity on perceptual grouping. *Perception & Psychophysics*, Vol. 1, pp. 300-302

Brown, M. Z.; Burschka, D. & Hager, G. D. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, pp. 993-1008

Ebihara, M.; Mahara, H.; Sakurai, T.; Nomura, A. & Miike, H. (2003a). Image processing by a discrete reaction-diffusion system, *Proceedings of the 3rd IASTED International Conference on Visualization, Imaging, and Image Processing*, pp. 448-453, Benalmádena, Spain, September 2003

Ebihara, M.; Mahara, H.; Sakurai, T.; Nomura, A.; Osa, A. & Miike, H. (2003b). Segmentation and edge detection of noisy image and low contrast image based on a reaction-diffusion model. *The Journal of the Institute of Image Electronics Engineers of Japan*, Vol. 32, pp. 378–385 [in Japanese]

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, Vol. 1, pp. 445-466

Gonzalez, R. C. & Woods, R. E. (1992). *Digital Image Processing*, Addison-Wesley Publishing Company, Reading

Julesz, B. (1960). Binocular depth perception of computer-generated patterns. *The Bell System Technical Journal*, Vol. 39, pp. 1125-1162

Kondo, S. & Asai, R. (1995). A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*. *Nature*, Vol. 376, pp. 765-768

Kuhnert, L. (1986). A new optical photochemical memory device in a light-sensitive chemical active medium. *Nature*, Vol. 319, pp. 393-394

Kuhnert, L.; Agladze, K. I. & Krinsky, V. I. (1989). Image processing using light-sensitive chemical waves. *Nature*, Vol. 337, pp. 244-247

Marr, D. & Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, Vol. 194, pp. 283-287

Marr, D.; Palm, G. & Poggio, T. (1978). Analysis of a cooperative stereo algorithm. *Biological Cybernetics*, Vol. 28, pp. 223-239

Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, Vol. 207, pp. 187-217

Marr, D. (1982). *Vision*, W. H. Freeman and Company, New York

Murray, J. D. (1989). *Mathematical Biology*, Springer-Verlag, Berlin

Nagumo, J.; Arimoto, S. & Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the I.R.E.*, Vol. 50, pp. 2061-2070

Nomura, A.; Ichikawa, M. & Miike, H. (1999). Solving random-dot stereograms with a reaction-diffusion model under the Turing instability, *Proceedings of the 10th International DAAAM Symposium*, pp. 385-386, Wien, Austria, October 1999

Nomura, A.; Ichikawa, M.; Miike, H.; Ebihara, M.; Mahara, H. & Sakurai, T. (2003). Realizing visual functions with the reaction-diffusion mechanism. *Journal of the Physical Society of Japan*, Vol. 72, pp. 2385-2395

Nomura, A.; Ichikawa, M. & Miike, H. (2004). Realizing the grouping process with the reaction-diffusion model. *IPSJ Transactions on Computer Vision and Image Media*, Vol. 45 (SIG 8/CVIM-9), pp. 26–39 [in Japanese]

Nomura, A.; Ichikawa, M. & Miike, H. (2005). Stereo vision system with the grouping process of multiple reaction-diffusion models. *Lecture Notes in Computer Science* 3522, Part I, pp. 137-144

Perona, P. & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 629-639

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T. & Flannery, B. P. (1988). *Numerical Recipes in C*, Cambridge University Press, Cambridge

Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, Vol. 47, pp. 7-42

Sunayama, T.; Ikebe, M.; Asai, T. & Amemiya, Y. (2000). Cellular νMOS circuits performing edge detection with difference-of-Gaussian filters. *Japanese Journal of Applied Physics*, Vol. 39, pp. 2278-2286

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, Vol. 237, pp. 37-72

Zitnick, C. L. & Kanade, T. (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 675-684

5

A Parallel Framework for Image Segmentation Using Region Based Techniques

Juan C. Pichel, David E. Singh[1]
and Francisco F. Rivera[2]
[1]*Computer Science Dept., Universidad Carlos III de Madrid*
[2]*Electronic and Computer Science Dept., Universidade de Santiago de Compostela*
Spain

1. Introduction

Segmentation is the partitioning of an image into multiple regions (sets of pixels) according to a given criterion. The goal of segmentation is typically to locate objects of interest within the image. A wide variety of methods and algorithms are available to deal with the problem of the segmentation of images (Fu and Mui, 1981; Haralick and Shapiro, 1985; Pal and Pal, 1993). These methods can be broadly classified into four categories (Zhu and Yuille, 1996):

- Edge-based techniques.
- Region-based techniques.
- Deformable models.
- Global optimization approaches.

The edge-based techniques are based on information about the boundaries of the image. Therefore, they try to locate the points in which abrupt changes occur in the levels of some property of the image, typically brightness (Canny, 1986; Rosenfeld and Kak, 1982). On the other hand, those methods that use spatial information of the image (e.g. color or texture) to produce the segmented image fit into the region-based techniques (Chen et al., 1992; Sahoo et al., 1988). These methods depend on the consistency of some relevant property in the different regions of the image. The deformable models are based on curves or surfaces defined within an image that moves due to the influence of certain forces. They can be classified into various groups, principally snakes, deformable templates and active contours (Blake and Isard, 1998; Kass et al., 1988). All of these techniques avoid the use of a global criterion when segmenting the image, which is contrary to the global optimization approaches (Geman and Geman, 1984; Kanungo et al., 1994).

In this work a unified framework for image segmentation is proposed. The technique consists of two stages: a parallel seeded region growing algorithm (PSRG) and a region merging heuristic (RM). In Figure 1 the functional scheme of the proposed algorithm is shown. In the first step, different segmentations, performed in parallel, of the same input image are obtained. Each of these segmentations, which from now on will be called *partial segmentations*, are also generated in parallel using different number of processors. This way, the region growing algorithm uses a two level parallelism. Next, a region merging heuristic is applied to the oversegmented image created as result of combining the different initial

segmentations. The merging process is guided using only information about the behavior of each pixel in the initial segmentations (without external parameters). In order to guide the merging stage we introduce a magnitude called repulsing force between neighboring regions that measures the tendency of them to remain separated in the oversegmented image. In order to stop the merging process an evaluation function of the segmented images was used. In addition the algorithm has been validated using several real images with different sizes and characteristics, and it has been tested on a HP Superdome cluster.

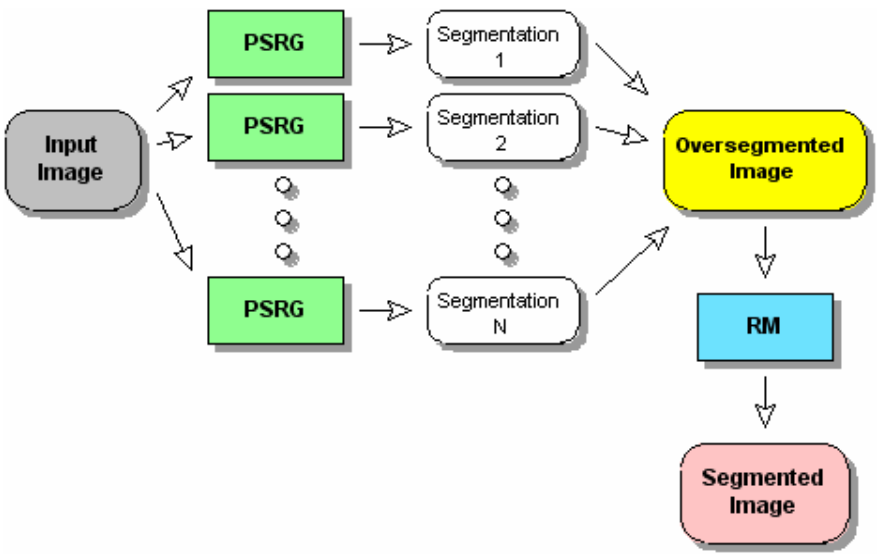


Fig. 1. Scheme of the proposed segmentation algorithm

2. Parallel Seeded Region Growing (PSRG) algorithm

Our proposal was inspired by the region growing algorithm introduced by Mehnert and Jackway (1997), referred as SRG (Seeded Region Growing algorithm) from now on. One of the main benefits of this algorithm is that it solves the dependencies imposed by the previous proposal by Adams and Bischof (1994) in the order to access the pixels in the image.

The SRG algorithm starts with a set of pixels in the image to be segmented, called seeds. These seeds are the starting point to determine the regions in the image. Pixels that are neighbor of the seeds are candidates to be included in the corresponding region. Some established similarity criterion is used to decide which of them are finally added to the corresponding region. This process is repeated sequentially in an incremental way, and the regions begin to "grow". In each iteration, the pixels that are candidates to be included in at least one region are stored in a Neighbour Holding Queue (NHQ). In this algorithm, the similarity δ between these pixels and its neighbour region is defined as the difference between the grey intensity of the pixel and the average value of intensities of the pixels that currently define the region.

In the SRG algorithm, the candidates are stored in a list of LIFO queues defined by consecutive and disjoint intervals of similarity, in such a way that pixels in the i-th queue

are those that present similarities in the interval [δ_i^{min},δ_i^{max}]. Queues are ordered according to δ_i^{min}, defining an ascending Priority Queue (PQ). In each iteration, only pixels in the First Queue (FQ), that is the queue with lowest δ_i^{min}, are considered, and in the SRG algorithm, all of them are immediately assigned to the corresponding region.
To avoid dependencies in the SRG algorithm, the average intensities of the regions are not updated in each assignment. Only after all the pixels in the FQ are assigned, this average is updated. Note that at this point new pixels can be included in the NHQ as well as in the PQ queues.

2.1. Implementation of the parallel algorithm

The proposal of a parallel algorithm based on the RGS is referred as PSRG (Parallel Seeded Region Growing algorithm). The main idea behind this proposal is to assign a different set of seeds or regions to each process, in such a way that the corresponding regions grow independently. Each process works with a subset of the regions. Regions assigned to a particular process are called local. The growing process in the regions of each process is not completely independent of the rest, but each process must take into account the state of the regions assigned to the remaining processes. Note that with this approach, the segmentation could not be the same as in the sequential case, and that it depends on the distribution of regions among processes.
In our approach, we consider second order neighbourhood for the pixels (Wang and Wang, 2004), however the algorithm can be adapted to any other case. The implementation of the parallel code uses the standard Message Passing Interface (MPI) library (Gropp et al., 1994). The main stages of the whole algorithm are described next.

2.1.1. Initial distribution of seeds

This stage can be considered as a preprocessing routine. Apart from initialling the parameters and variables, i.e. the position of the seeds, its main objective is to distribute the seeds (regions) among the available processes. As we will see in next stages, this distribution is important because the number of overlaps directly depends on it. *Overlaps* are the pixels that are simultaneously assigned to different regions. Note that regions in the same process can not produce overlaps. Therefore, in order to reduce overlaps, regions that are going to be neighbours must be assigned to the same process. This situation can not be foreseen, but in many cases a good approach can be obtained if we consider the distance among seeds, in such a way that seeds that are near to each other have a large probability of producing neighboring regions. We used a version of the Prim´s algorithm (Gibbons, 1984) on the position of the seeds in the image to reorder and distribute the regions. In addition, the regions are equally distributed among processes to achieve good load balance.

2.1.2. Parallel Region Growing

In this stage the SRG algorithm is applied to the set of regions in each process with the addition of two types of communications among processes: one to detect overlaps, and other to control the growing speed to avoid artificial growing of some regions. Two parameters T_1 and T_2 are introduced to determine the interval between pairs of communications of each type in terms of number of iterations of the SRG algorithm. This stage is referred as PRG algorithm. Next, both types of communications are introduced with detail.

2.1.2.1. Communications to detect and solve overlaps

After T_1 iterations a new communication to deal with overlaps is performed. For each process the objective is to know the pixels already assigned by other processes. Therefore, overlaps (pixels assigned to different regions in different processes) among regions are detected. These pixels are labelled as *borderline pixels*, and their neighbours are not considered as candidates in the following iterations.

This communication can be efficiently implemented by a reduction operation. Each processor labels the pixels already assigned to any local region as 1, and labels 0 the rest of them. After the reduction (summation) of the labels among processors for the whole image, the pixels with labels higher that 1 are considered overlaps. Note that only two bits are needed to label the pixels.

The number of iterations between communications is a parameter that affects the performance of the parallel code. We propose to consider as a initial value a estimation of the number of pixels to be processes before the first overlap:

$$T_1 = \frac{N}{R} \cdot D_{min} \tag{1}$$

Where R is the number of regions, P is the number of processors and D_{min} is the minimum euclidean distance in number of pixels among seeds assigned to different processes.

After this first value, T_1 changes dynamically according to the number of pixels detected as overlaps in the previous communication. We propose to use the values given by:

$$T_1 = \alpha \cdot \frac{\beta}{\Delta K} \tag{2}$$

Where α is a parameter that characterizes the cost of the communications in the particular system, β is the agreeable maximum number of pixels in the overlap areas between pairs of communications (this parameter can be tuned by the user), and ΔK is the number of new overlap pixels since the previous communication.

2.1.2.2. Communications to control the growing speed

In our parallel algorithm, each process has its own PQ and FQ, and the values of similarity they consider in a particular iteration can highly differ from one process to another. Therefore it is necessary to include some action to avoid unfair grows produced by local FQs with lower values of δ than in other FQs.

To deal with this problem, we propose to use a reduction operation to evaluate de maximum and minimum values of δ used in the FQs, δ^{max} and δ^{min} respectively. In such a way that a new parameter ϕ is defined to specify the agreeable interval of similarities to be processed in each iteration given by the size L:

$$L = (\delta^{max} - \delta^{min})\phi \qquad 0 \le \phi \le 1 \tag{3}$$

Therefore, if the similarity of a particular pixel in a local PQ is lesser than δ^{min} + L, then it is processed, otherwise it is not assigned to the associated region. The number of iterations between reductions is defined by T_2 (established by the user). Note that this communication

presents a lower cost that the previous one, because it involves just two values instead of information about all the pixels in the image.

2.1.3. Redistribution of seeds

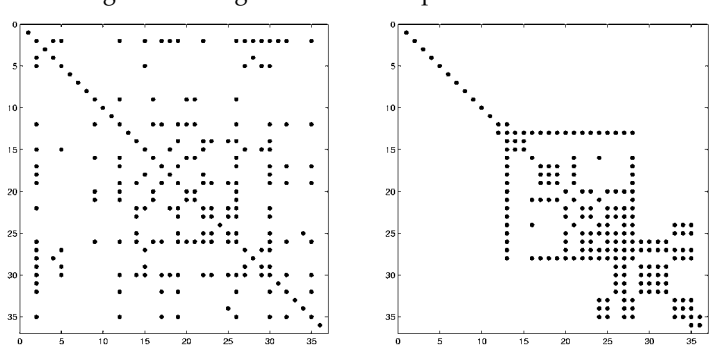
In this stage, a new distribution of regions among processes is performed in order to assign overlap pixels to regions. The objective is to minimize the number of overlap pixels after the execution of the PRG algorithm. This new distribution just involves the overlap area of the image. This stage consists of three steps:

1. Finding overlap pixels.
2. Obtaining the new redistribution of regions among processes.
3. Finding the optimum number of processes needed and establishing communications to perform the redistribution.

The idea behind this stage is to assign those regions that share overlap pixels to the same process, in such a way that a new execution of PRG can assign these pixels locally. Next we analyze the above three steps with more detail.

After the PRG algorithm was executed, each process has a number of regions defined by a set of pixels, and no other process has these pixels assigned to any of its regions. In fact, all these regions are limited by a border line defined by the overlap pixels or the frame of the image. So, a reduction operation involving all the processes is carried out. The objective of these communications is every process to know the overlap pixels and the regions that have them as part of its borderline.

Two regions are said to be close to each other if both have as neighbour at least one overlap area. In this step close regions are detected by a parallel flooding algorithm. As a result of this, a so called adjancency matrix M is obtained. This matrix is defines as: M[i][j]=0 if regions *i* and *j* are not close to each other, and M[i][j]=1 otherwise. Note that M is a symmetric matrix. Then, the Cuthill-McKee algorithm (Saad, 1996) is applied to reorder the matrix in such a way that the nonzero entries are moved near the diagonal. Figure 2 shows the effect of applying this algorithm. This reordered matrix can be partitioned into contiguous blocks that are distributed among the processes. The result of this distribution is that groups of close regions are asigned to the same process.



(a) (b)

Fig. 2. (a) Original adjacency matrix, (b) reordered matrix using the Cuthill-McKee algorithm.

2.1.3.3. Obtaining the number of processes and establishing communications

In this step, the most appropriate number of processes is obtained. Note that regions that are not close to any other are not relevant for this stage, because they can not grow, so they are not being condidered in this step. In order to obtain a good load balance, the same number of regions are assigned to the processes Figure 3 shows an example of this partition for 3 processes. The entries that are inside one of these partitions are solved in this step, however the others (marked as shared entries in the figure) will not be solved now (they represent the shared overlap areas). The time needed to finish this step is limited by the process that has more entries in the partition of M. This number is denoted as E_{max}. In addition, shared entries will be processed in the final stage, if their number is E_{shared}, then the cost of both processes can be modelled by the linear expression $K = A \cdot E_{max} + B \cdot E_{shared}$. Parameters A and B are used to weight the relative cost among of this step and the final stage. In our experiments, we found that adequate values for them are: $A=1.5$ and $B=1$. Therefore, to obtain the optimum number of processes, K should be minimized. Finally, the regions are distributed among the selected processes, and the PRG algorithm is executed.

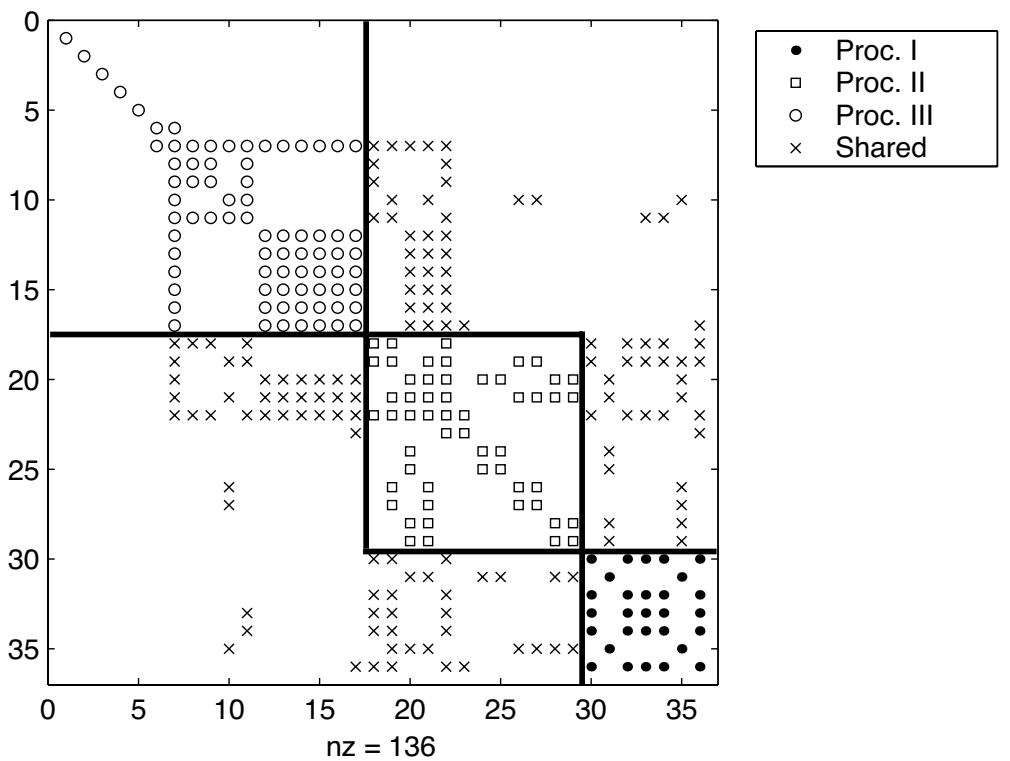


Fig. 3. Example of the partitioning of an adjacency matrix.

2.1.4. Final stage

In this stage, the shared overlap areas are solved sequentially by the RGS algorithm, and therefore the last overlap pixels are finally assigned to regions. Note that after this stage,

some groups of pixels, called *islands*, could be disconnected from the seed. These islands are easily detected by a flooding algorithm and finally added to their best neighbour region according to the similarity criterion.

3. Obtaining the oversegmented image

The PSRG algorithm presents two problems. On one hand, it has a great dependence with the initial position of the seeds. Moreover, the number of seeds is the number of regions of the final segmented image and therefore, we need *a priori* knowledge of the image to obtain a good segmentation. On the other hand, this type of segmentation algorithms (region based ones) force all the pixels of the image to belong to a region in the segmented image, so there will be pixels that belong to regions with low similarity levels.

We deal with these problems creating an oversegmented image from different executions of the PSRG algorithm with seeds placed randomly and introducing the concept of shadow zone. Later, this oversegmented image will be processed by a region merging method (detail in Section 4) to obtain the final segmented image. Note that this proposal presents a two-level parallelism: a coarse-grain one defined by the parallel execution of several PSRGs, and a fine-grain one defined by the parallel nature of the PSRG algorithm.
The MPI library was used to implement the parallel code of this proposal. This library allows the definition of groups of processes that fits with our two-level parallelism model, in such a way that the number of processes to solve each partial segmentation and the number of partial segmentations can be easily established. In other words, if N partial segmentations are executed with P processes each one, then the total number of processes is NxP.

3.1. Generation of seeds

Each partial segmentation is obtained from a different set of seeds. These seeds can be obtained randomly if there is no a priori information about the image. After that, each segmentation is obtained by executing the PSRG algorithm with P processors.

3.2. Shadow zones

As we have mentioned above, one of the drawbacks of the SRG and PSRG algorithms is that the number of regions is exactly determined by the number of seeds. Generally, the objective of the segmentation algorithms based on regions is the labeling of all the pixels of the image. In many cases, in the final stages of the process, this situation causes pixels to be included in regions from which they have very low similarity levels, thereby creating regions with low homogeneity. To avoid this effect we propose the inclusion of a specific threshold (ε) as a possible solution. This threshold would be such that those pixels with a low degree of similarity with respect to the target region are not included in it, remaining unlabeled. The set of unlabeled pixels will be called shadowed zones. Therefore, a pixel, after the partial segmentation, can be labeled and thus belongs to a region, or it can be included in a shadowed zone.
Figure 4 shows the effect of applying PSRG with different values of ε on the Lena image and using 30 seeds. In particular Figure 4(b) shows the segmentation produced by PSRG without threshold, and Figures 4(c), 4(d) and 4(e) show the result when ε is more and more restrictive. Note that when ε is low, the image present more details that when ε is high.

However, when ε is too low, the number of shadow zones can be so high that their execution in the following stages is less efficient.
It is important to note that the shadow pixels are not processed by PSRG algorithm. Any way, they will be taken into account in the creation of the oversegmented image and by the merging process.

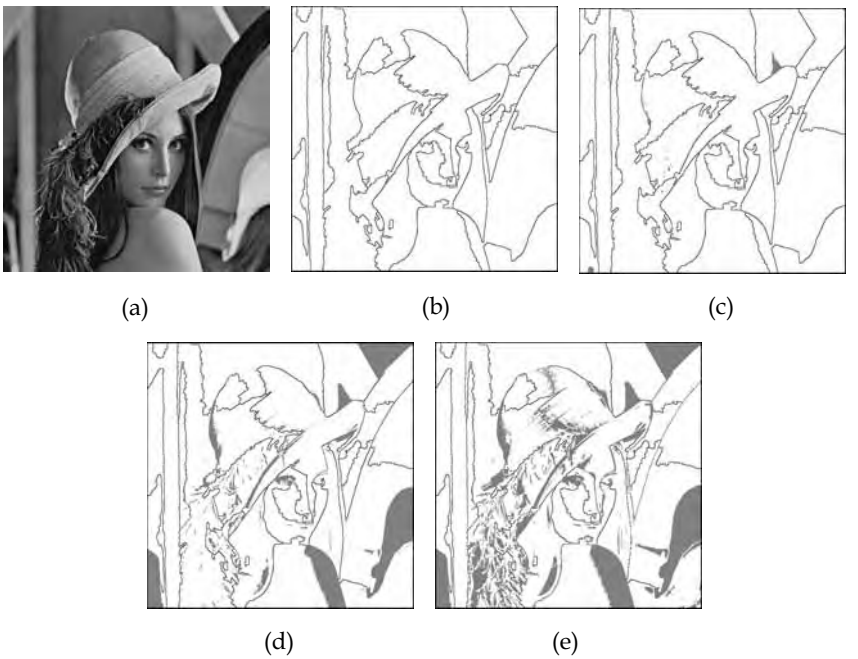


Fig. 4. Partial segmentations of the image Lena: (a) Original image, (b) ε = 255, (c) ε = 100, (d) ε = 50 and (e) ε = 35.

Figure 5 shows two partial segmentations produced by two different sets of seeds for the PSRG algorithm with no threshold ($\varepsilon = 255$) (Figures 5(a) and 5(b)) and the PSRG algorithm with $\varepsilon = 35$ (Figures 5(c) and 5(d)). As expected, note that the partial segmentations obtained by PSRG with $\varepsilon = 35$ are more similar to each other than the ones obtained by PSRG with $\varepsilon = 255$. We can conclude that using PSRG with shadow zones minimize the dependence of the final segmentation from the initial position of the seeds.
Our proposal generates an oversegmented image as a combination of various partial segmentations in which shadowed areas can exist. There are other segmentation techniques that create an oversegmented image such as watershed algorithms (Haris et al., 1998). As we detail later, we need to collect, from different partial segmentations, the information that will guide the merging process. In our algorithm an operation for intersecting all the partial segmentations is performed in such a way that those pixels that belong to the same region in all the partial segmentations remain united in one of the regions of the oversegmented image. Figure 6 shows a simple example of the creation of an oversegmented image formed, in this case, from three partial segmentations. The first partial segmentation presents a shadowed zone and two regions, whilst in each of the other two segmentations there are

three regions and no shadowed zones. In this example the oversegmented image consists of five regions.

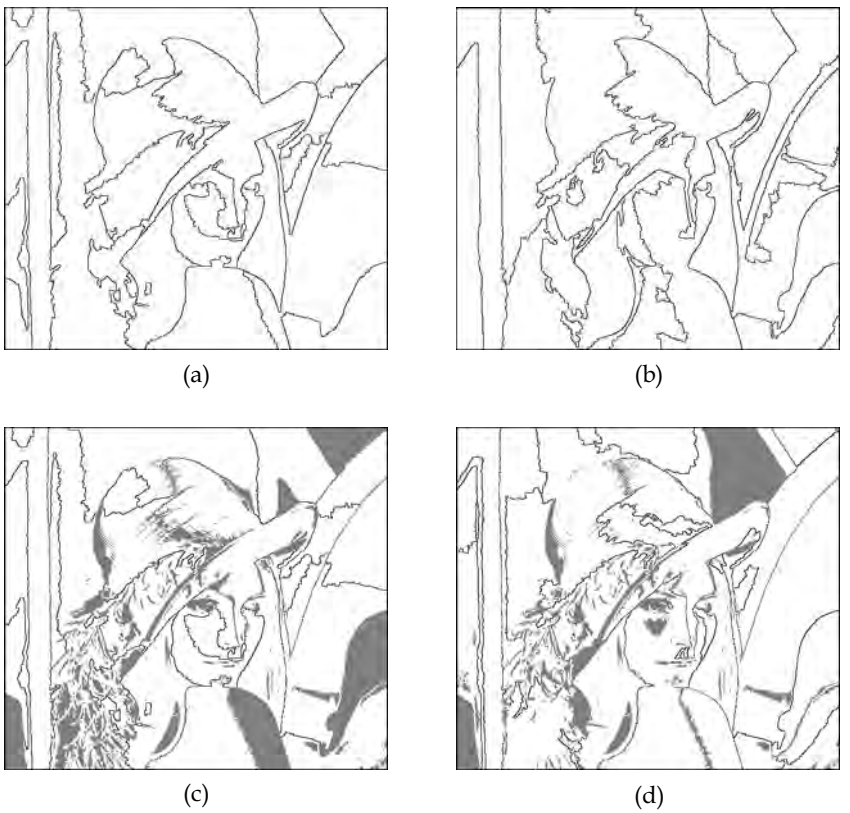


Fig. 5. Partial segmentations of the image Lena using different seeds: (a) and (b) ε = 255, (c) and (d) ε = 35.

In Figure 7 two oversegmented images obtained using four partial segmentations of Lena are shown. Note that when the threshold ε decreases, the number of regions of the oversegmented image grows. This behaviour is due to the existence of many shadowed zones in the partial segmentations obtained using the PSRG algorithm. This way, if shadowed zones exist in some of the partial segmentations, they are considered as any other region in the generation of the oversegmented image, although later, in the merging algorithm, they will be treated differently.

Finally, Figure 8 shows the evolution of the number of regions of the oversegmented images created from 2, 3 and 4 partial segmentations of the Lena image using different number of seeds. The results show that when using a higher number of seeds, the number of regions of the oversegmented images increases. This behavior is observed as well for an increasing number of partial segmentations.

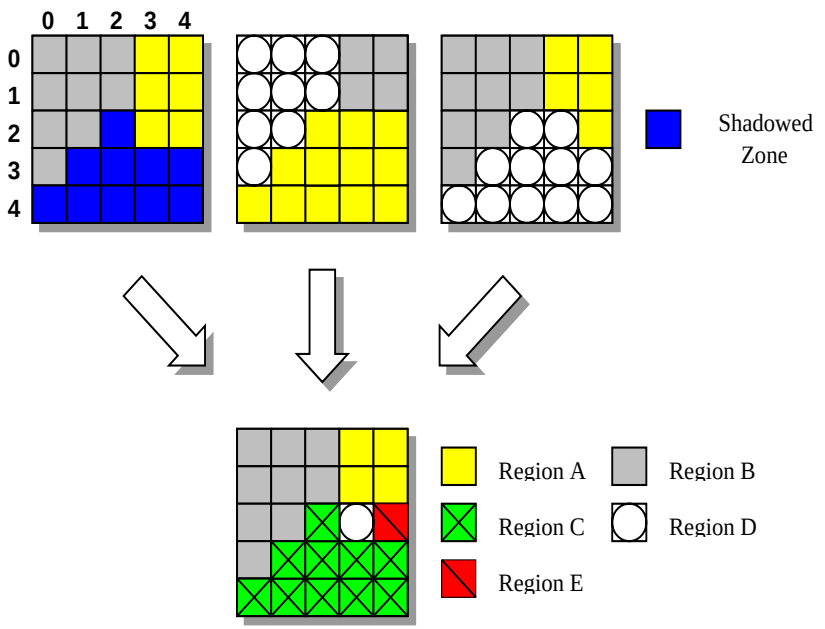
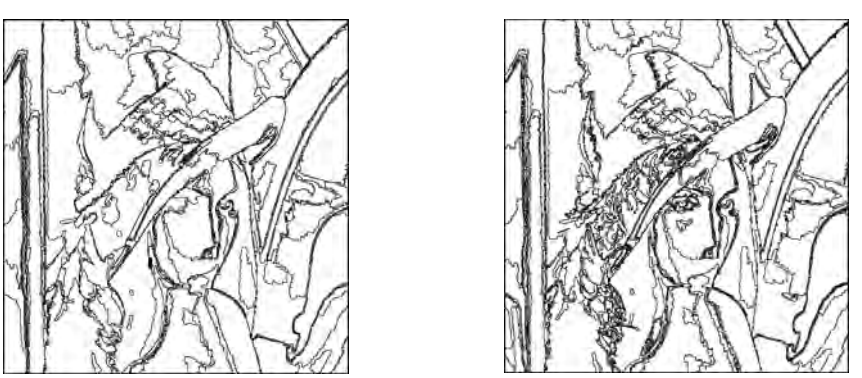


Fig. 6. Example of the generation of an oversegmented image from three partial segmentations of 5×5 pixels with three regions each one.



(a) (b)

Fig. 7: Oversegmented image created using four partial segmentations of the image Lena: (a) ε = 255 and (b) ε = 50.

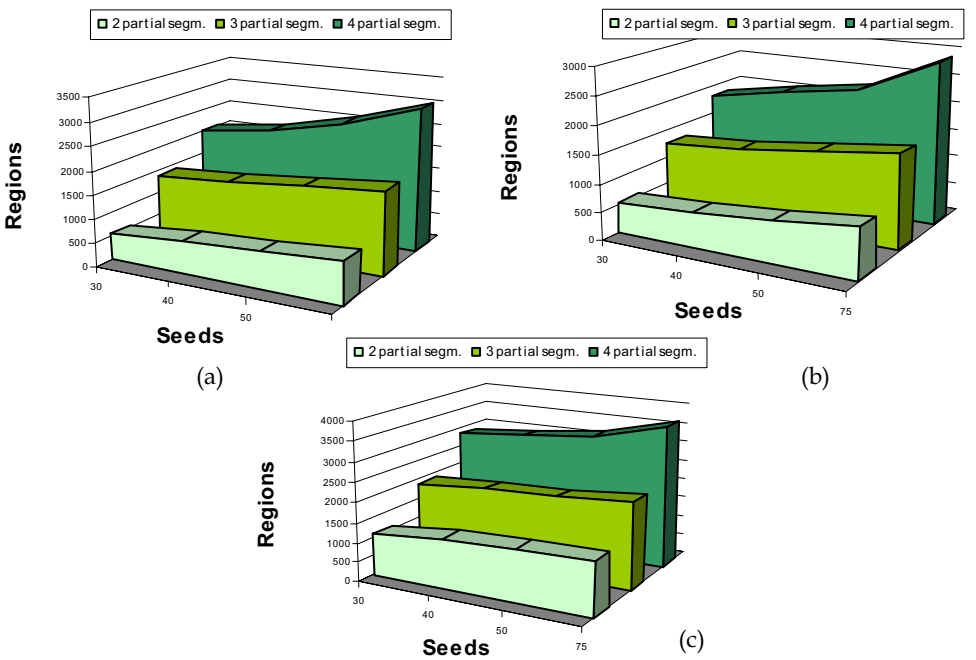


Fig. 8. Average number of regions of the oversegmented image using Lena as input image: (a) ε = 255, (b) ε = 100 and (c) ε = 50.

4. Region Merging (RM) algorithm

In a previous work (Pichel et al., 2006) a new region merging algorithm was introduced. The main contribution of this proposal is that all the relevant information to obtain the final segmented image is obtained exclusively from the different partial segmentations, both for creating an oversegmented image and for applying the subsequent region-merging algorithm. In this paper, the partial segmentations are performed using the PSRG algorithm, and later the RM algorithm is applied. This strategy does not take into account local characteristics such as size, shade of average grey intensities, etc. Based on the results obtained for all the pixels of the image in each of the partial segmentations, global conclusions are obtained with respect to them. Without losing generality, the merging algorithm uses a second-order neighborhood scheme, so that up to eight neighbors are defined for each pixel.

The RM algorithm uses a force of repulsion between two neighboring pixels i and j that measures the tendency of these pixels to be or not in the same region. This force is given through the following equation:

$$f_{ij} = -C_1 n_1(i,j) - C_2 n_2(i,j) + C_3 n_3(i,j) + C_4 n_4(i,j) \tag{4}$$

where C_K are parameters to weight each of the situations in which two neighboring pixels could be found, and n_K is the number of times this situation occurs in the initial segmentations. The four situations are the following: both belong to same labeled region (n_1), both belong to different labeled regions (n_2), both belong to shadowed zones (n_3), and one belongs to a region and the other to a shadowed zone (n_4).
Therefore, two different components can be identified in Equation 4: on one hand an attractive component given by term $-C_1n_1(i,j)- C_2n_2(i,j)$ that measures the tendency of these pixels to belong to the same region, and on the other, a repulsive component given by term $C_3n_3(i,j)+C_4n_4(i,j)$ that measures the opposite. According with this equation, the lesser the force f_{ij} the greater is the tendency for these pixels to belong to the same region. Additionally, we have that the pixels that belong to the same region always verify $f_{ij}<0$.
In (Pichel et al., 2006) an analysis to determine the way these parameters are related was performed. We can summarize the relationships between the parameters as: $C_4 > C_1 > C_3 > C_2$. Moreover, several definitions were introduced. Let R be a region in the oversegmented image, and let S be a neighboring or adjacent region to it. We define the set of pixels of R neighbors of S as:

$$V_{R,S} = \{ i \in R \mid \exists j \in S, \text{such that i and j are neighbors}\} \tag{5}$$

S and R are neighbors if $V_{R,S} \neq \varnothing$.
For each i in $V_{R,S}$, we define the associated set of its neighbors belonging to S, as:

$$U^i_{R,S} = \{ j \in S \mid \text{ i and j are neighbors, and } i \in V_{R,S}\} \tag{6}$$

Then, the force of repulsion between neighboring regions is defined as the average of the forces of repulsion of all the neighboring pixels belonging to each one of those regions:

$$F_{R,S} = \frac{\sum_{i \in V_{R,S}} \left(\sum_{j \in U^i_{R,S}} f_{ij} \right)}{\sum_{i \in V_{R,S}} \| U^i_{R,S} \|} \tag{7}$$

where $\| \bullet \|$ is the cardinality operator.
The structure of the data used for representing the partitions of the oversegmented image is a region adjacency graph (RAG) (Haris et al., 1998). The RAG of a segmentation of K regions is defined as a weighted undirected graph, $G=(V,E)$, where $V=\{1,2,...,K\}$ is the set of nodes and $E \subset V \times V$ is the set of edges. Each region is represented by a node, and between two nodes $R,S \in V$ there is an edge (R,S) if the regions are neighbors. A weight is assigned to each edge of the RAG, so that those nodes joined by the lesser (or greater) weighted edge, depending on its definition, will be the regions that are candidates for merging. In our case, the function used to assign weights to the edges is the force of repulsion F_{RS} given by the Equation 7 and therefore, the regions that are candidates for merging are those joined by the edge with least weight.
Using the RAG as input, an iterative heuristic to deal with the problem of merging is proposed, so that one merging is performed in each of the iterations, based on the weight of

the edges. In each iteration of the RM algorithm, the pair of regions that have the smallest weight are merged. A data structure adequate for storing the weights is a queue, which can be implemented using a heap (Knuth, 1973). All the edges of the RAG are stored in the heap according to the weights, so that the first edge always has the smallest weight. Given the RAG of an initial partition of *K* regions denoted as (K-RAG), and a heap of its edges, the RAG of the partition *K-n* is obtained using the merging algorithm described in the following pseudo-code:

```
DO i = 0, n-1
    Find minimum cost edge in (K-i)-RAG
    Merge the selected pair of regions to get the (k-i-1)-RAG
    Update the heap
END DO
```

The *K-RAG* corresponds to the initial partition of the image, which in our case, is the oversegmented image. Subsequently, an iterative process is applied, in which, in the *i*-th iteration, the two regions with the smallest eight are merged. Once they have been merged, the list of edges is updated, and the *(K-i-1)-RAG* is obtained.

It is inferred that *n* iterations are needed to obtain the *(K-n)-RAG*. Therefore, one of the problems posed by this strategy is to establish the best value of *n*, i.e., the best number of regions of the final segmented image. Different alternatives can be used. For example, using the property of the growing value of the first term of the heap, a certain threshold can be selected to stop the iterations when the value of the edge at the top of the heap exceeds it. The main drawback of this approach is that it is not evident to determine a good threshold *a priori*. Another approach is the use of a method that allows a numerical evaluation of the segmentation for choosing the best threshold according to some selected validation criterion.

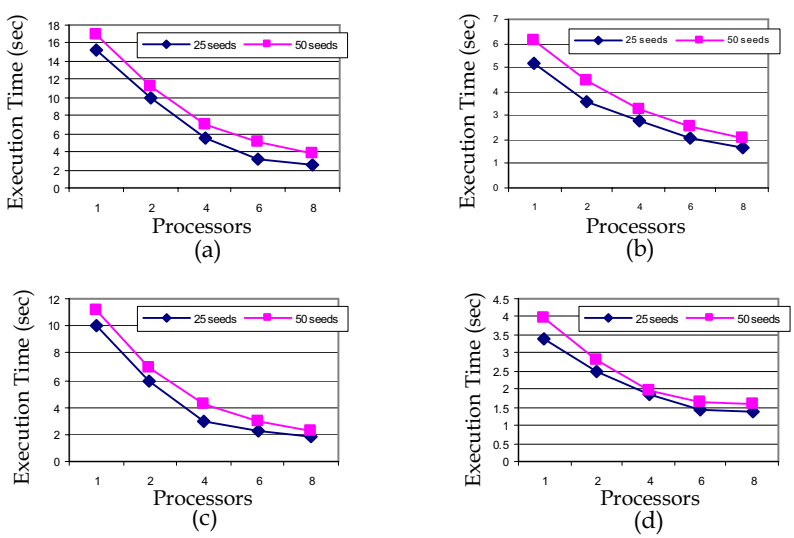


Fig. 9. Average execution times of the PSRG algorithm using different parameters for the image Lena (512×512 pixels): (a) $\phi = 0.01$, $\varepsilon = 255$, (b) $\phi = 0.05$, $\varepsilon = 255$, (c) $\phi = 0.01$, $\varepsilon = 25$ and (d) $\phi = 0.05$, $\varepsilon = 25$.

5. Results

In this section the results obtained by our proposal are shown. We have focused on two different aspects: the execution time required by the algorithm to obtain a final segmented image and the quality of the segmentation provided by the algorithm.

5.1. Execution Times

Our algorithm has been tested on a HP Superdome cluster with 128 1.5 GHz Itanium2 processors and 384 GBytes of memory. As example, in Figure 9 the execution times of the PSRG algorithm using an image of 512×512 pixels (Lena) are shown. The graphics point out that the code presents a good scalability, obtaining speedups up to 6.2 when using 8 processors per segmentation. Note that the execution times when using ε are lower than those obtained when the threshold is not applied (ε = 255). This behavior is due to, as we have commented before, the higher number of pixels to be added to the regions. In turn when ϕ increases, the execution time of the PSRG algorithm also increases.

Processors per partial segmentation

Execution Time (sec)	**1**	**2**	**4**	**6**	**8**
PSRG	9.9	5.8	3.1	2.3	1.9
RM	7.1	7.1	7.1	7.1	7.1
PSRG+RM	17.0	12.9	10.2	9.4	9.0

Table 1. Average execution times of the segmentation algorithm for the image Lena using a different number of processors per partial segmentation.

Finally, we have measured the execution times of the global segmentation system including the parallel algorithm (PSRG) and the sequential one (RM). The results are shown in Table 1. In the example, four partial segmentations were performed in order to create the oversegmented image, with ε = 100 and ϕ = 0.05.

5.2. Evaluation of the segmentation

As case of study we use function Q both to objectively evaluate the quality of the algorithm, as well as to adjust the value of the parameters of the weight function of the edges of the RAG. The evaluation function Q was proposed by Borsotti et al.(1998) for color images, which is a variant of that proposed by Liu and Yang (1994). One of its main advantages is that do not require any external parameter. Specifically, function Q is expressed by:

$$Q = \frac{1}{10000(N \times M)} \sqrt{R} \sum_{i=1}^{R} \left[\frac{e_i^2}{1 + \log A_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right] \tag{8}$$

where N×M is the size of the image, R is the number of regions, A_i and e_i are the area in number of pixels and the quadratic error of the color values of the i-th region, respectively. Also, $R(A_i)$ represents the number of regions that have an area equal to A_i. The smaller the value of Q, the better the segmentation of the image. For images in grey levels, we have

adapted the normalization term of the previous equation to obtain values within a similar range to those obtained by Q in color images, and the definition of e_i corresponds to the mean value of the grey level of the i-th region.

In (Pichel et al., 2006) an exhaustive study was performed to a broad set of test images in order to determine the values of the weight parameters of the force of repulsion. Finally, the following values were proposed: C_1=1, C_3 =0.2, C_3=0.8 and C_4=2.

In order to illustrate the behaviour of our proposal in a more precise way, in Figures 10, 11 and 12 the values of function Q compared to the number of regions of the oversegmented image using a different number of partial segmentations are shown. Shadowed zones, defined by different thresholds, have been used in all the tests. From the behavior Q, we can infer that when the number of regions is equal to the number that each image really has, Q presents its minimum value.

This behaviour is absolutely clear in the case of the image Test1 (Figure 10). This is a synthetic image and consists of 5 homogeneous regions. For this image, as we can observe in the figure, only two partial segmentations are needed to obtain a correct final segmented image. Nevertheless, the situation is different when the input image is a real one like Lena and Peppers (Figures 11 and 12). Note that, in these cases, a clear global minimum of Q does not exist, so we cannot decide on which is the best segmentation with this criterion. The information that we can extract is that an interval of values of Q exists, shown in the figures with an arrow. In this interval the most adequate segmentations can be found. The segmented images displayed correspond with a local minimum of function Q when using six partial segmentations.

Therefore, based on these results we conclude that for the segmentation of real images, Q is very useful for determining the set of the most adequate segmentations, but in most of the cases it is not going to be sufficiently discriminating to select just one. But note that using our proposal high quality segmentations are obtained.

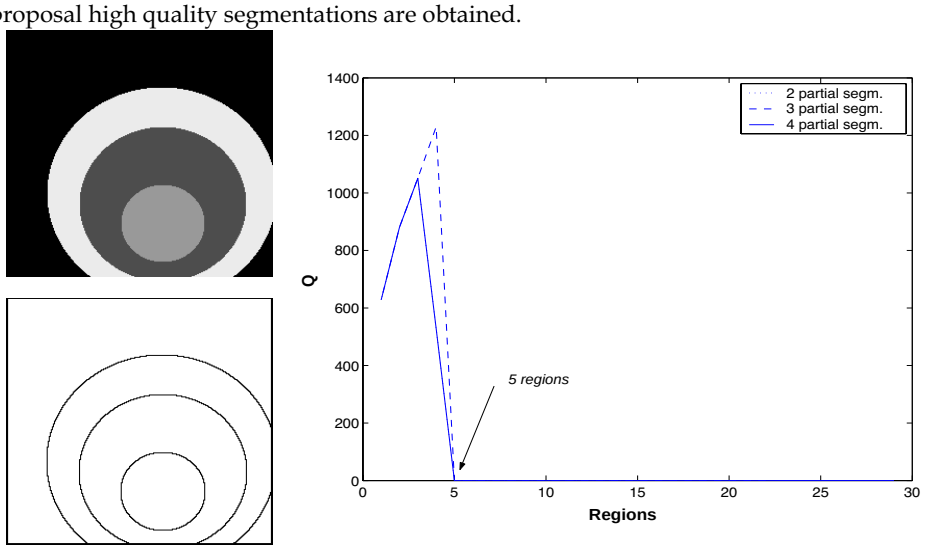


Fig. 10. Segmentation of the image Test1: original image, result of the segmentation and Q compared to the number of regions.

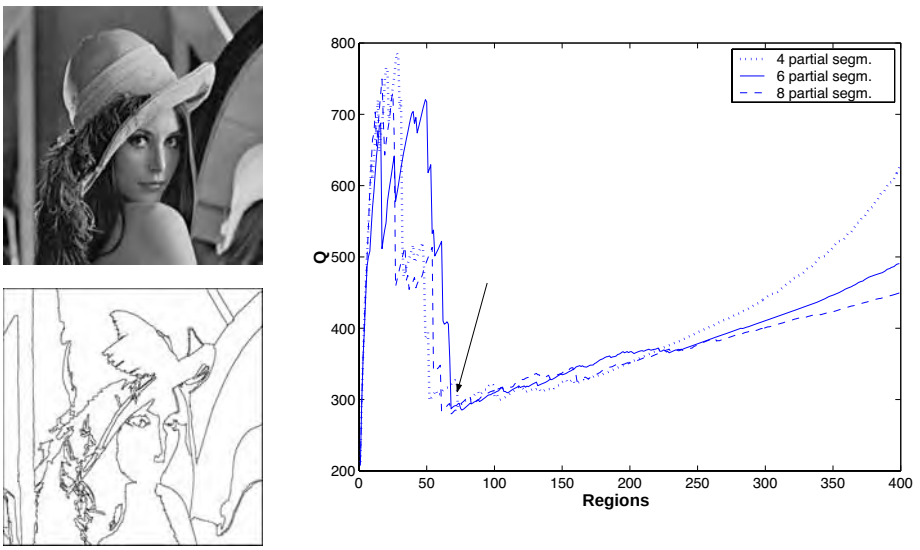


Fig. 11. Segmentation of the image Lena: original image, result of the segmentation and Q compared to the number of regions.

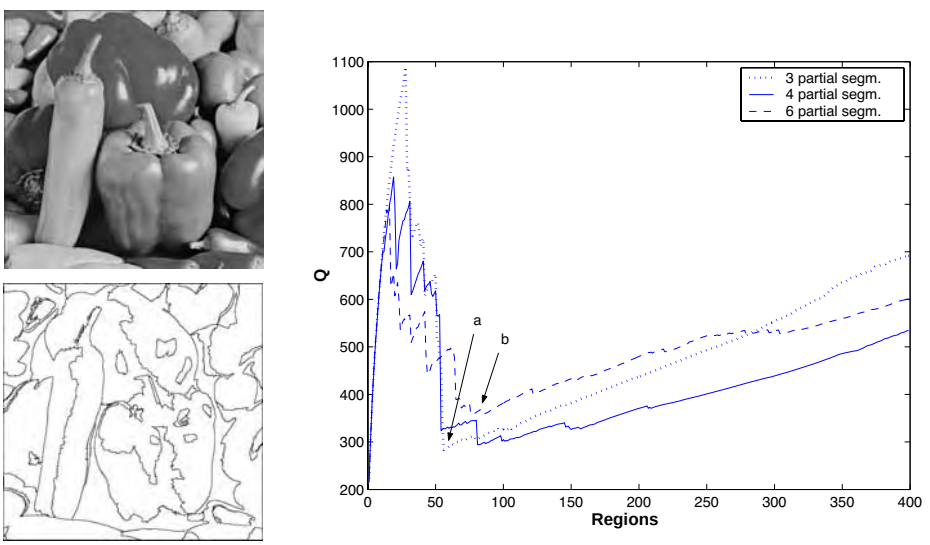


Fig. 12. Segmentation of the image Peppers: original image, result of the segmentation and Q compared to the number of regions.

6. Conclusions

In this work a parallel framework for image segmentation using region based techniques is presented. The algorithm is based on performing several segmentations of the same image using a parallel region-based algorithm. Moreover these segmentations are also obtained in parallel. This way, our proposal presents a two-level parallel layout. Next, an oversegmented image that collects all the information from the previous segmentations is created. A region-merging algorithm, developed previously by the authors, is then applied to this oversegmented image. A relevant aspect is that the information obtained from the partial segmentations will, in fact, guide the merging process, in such a way that the actual characteristics of each region or pixel are not taken into account.

The merging algorithm uses the concept of force of repulsion between neighboring pixels that indicates quantitatively their tendency to form part of different regions. The force of repulsion considers several situations in which any two neighboring pixels can be found in all the partial segmentations that are used to create the oversegmented image, including the shadowed zones. The shadowed zones are groups of pixels that differ in their intensity level a certain threshold from the region in which they could be included. Introducing this concept in the region-based algorithms, regions with low levels of homogeneity are avoided, improving the quality of the whole process. Note that, given that the shadowed zones are not treated by the algorithm, the information that can be extracted from these zones is minimum. As stopping criterion of the merging algorithm, we use a function to evaluate the quality of the segmentation.
The algorithm has been validated using several artificial and real images demonstrating the benefits of our proposal, and it was tested on a HP Superdome cluster.

7. Acknowledgements

This work was supported in part by the Ministry of Science of Spain through the TIN2004-07797-C02 project. The authors also thank the supercomputing facilities provided by CESGA.

8. References

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Anal. Machine Intell.* Vol. 6, No. 16, pp. 641–647.

Blake, A. and Isard, M. (1998). *Active Contours*. Springer.

Borsotti, M.; Campadelli, P. and Schettini, R. (1998). Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*. Vol. 19, pp. 741–747.

Canny, J.F. (1986). A computational approach to edge-detection. *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 8, pp. 679–698.

Chen, S.; Lin, W. and Chen, C. (1992). Split-and-merge image segmentation based on localized feature analysis and statistical tests. *CVGIP: Graph. Models Image Process*. Vol. 53, No. 5, pp. 457–475.

Fu, K. and Mui, J. (1981). A survey on image segmentation. *Pattern Recognition*, Vol. 13, No. 1, pp. 3–16.

Geman, D. and Geman, S. (1984). Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell*. Vol. 6, pp. 721–741.

Gibbons, A. (1984). *Algorithmic graph theory*. Cambridge University Press..

Gropp, W.; Lusk, E., Skjellum, A. (1994). *Using MPI Portable Parallel Programming with the Message Passing Interface*. The MIT Press.

Haralick, R. and Shapiro, L. (1985). Survey, image segmentation techniques. *Comput. Vision Graphics Image Process*. Vol. 29, No. 1, pp. 100–132.

Haris, K.; Efstratiadis, N.; Maglaveras, N. and Katsaggelos, A. K. (1998). Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. Image Process*. Vol. 7, No. 12, pp. 1684–1699.

Kanungo, T.; Dom, B.; Niblack, W. and Steele, D. (1994). A fast algorithm for MDL-based multi-band image segmentation. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 609–616.

Kass, M.; Witkin, A. and Terzopoulos, D. (1988). Snakes: Active contour models. *International J. Computer Vision*, Vol. 1, No. 4, pp. 321–331.

Knuth, D. E. (1973). *The Art of Computer Programming: Sorting and Searching*. Addison-Wesley.

Liu, J. and Yang, Y. 1994. Multiresolution color image segmentation. *IEEE Transactions on Pattern Anal. Machine Intell*. Vol. 16, No. 7, pp. 689–700.

Mehnert, A. and Jackway, O. (1997). An improved seeded region growing algorithm. Pattern Recognition Letters . Vol. 27 , No. 10, pp. 1065–1071.

Pal, N. and Pal, S., (1993). A review on image segmentation techniques. *Pattern Recognition*, Vol. 26, No. 9, pp. 1277–1299.

Pichel, J. C.; Singh, D. E and Rivera, F. F. (2006). Image segmentation based on merging of sub-optimal segmentations. *Pattern Recognition Letters*, Vol. 27, No. 10, pp. 1105–1116.

Rosenfeld, A. and Kak, A. (1982). *Digital Picture Processing*. Academic Press, New York.

Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company.

Sahoo, P.; Soltani, S.; Wong, A. and Chen, Y. (1988). A survey of thresholding techniques. *Comput. Vision Graphics Image Process*. Vol. 41, No. 1, pp. 233–260.

Wang, X. and Wang, H. (2004). Evolutionary Gibbs sampler for image segmentation. *Proceedings of International Conference on Image Processing*, pp. 3479-3482.

Zhu, S.C. and Yuille, A. (1996). Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. Pattern Anal. Machine Intell*. Vol.18, No. 9, pp. 884–900.

6

A Real-Time solution to the image segmentation problem: CNN-Movels

Giancarlo Iannizzotto, Pietro Lanzafame, Francesco La Rosa
University of Messina (VisiLAB)
Italy

1. Introduction

2D Image segmentation has been a main issue in image analysis since the very early years. Traditional literature usually classifies segmentation approaches as *area-based* or *contour-based*. In the second class, among dozens of different approaches, *Active Contours* have recently gained more and more interest. Active contours (also known as *deformable models*) are open or closed curves that can accurately fit to the contours of objects featuring almost any kind of shape. These models are called *active* because they automatically respond to specific characteristics of the points of the image, by changing their shape consequently. For example, an active contour can respond to the *edgeness* values of the image points.
A particular type of active contour is the *snake*: it responds both to the characteristics of the points of the image (through the minimization of a quantity called external energy), and to specific internal laws ruling its shape and way of deformation, tending to minimize a quantity called *internal energy* (Kass et al., 1988; Lai & Chin, 1995).
It usually consist of elastic curves that, located over an image, evolve from their initial shapes and positions in order to adapt themselves to the notable characteristics of the scene. This evolution comes as a result of the combined action of external and internal forces. The external forces lead the snakes towards features of the image, whereas internal forces model the elasticity of the curves. In a parametric representation, a snake appears as a curve u(s)=(x(s),y(s)), s ϵ [0,1], with u(0)=u(1). Its internal energy is often defined as

$$E_i(u(s)) = \alpha|u_s(s)|^2 + \beta|u_{ss}(s)|^2 \tag{1}$$

A snake is made up of two factors: the membrane energy $\alpha|u_s(s)|^2$, which weights its resistance to stretching, and the thin-plate energy $\beta|u_{ss}(s)|^2$, that weights its resistance to bending. The terms $u_s(s)$ and $u_{ss}(s)$ represent the first and second derivatives respectively. The elasticity parameters α and β control the smoothness of the curve. The external energy is generally defined as a potential field P,

$$E_e(u) = \int_0^1 P(u(s))ds \tag{2}$$

This external potential is a combination of different terms based on the application and the characteristics of interest.
The total energy of the snake will be the sum of the external and internal energy terms along the curve u(s):

$$E_{snake}(u) = \int [E_i(u(s)) + E_e(u(s))] ds \tag{3}$$

The solution to the problem of detecting the contour is found in the minimization of this energy function.
Some other variations of the snake (*snake spline*) are represented by its parametric formulations much quicker and computationally less expensive (Flickner et al., 1996). Snakes, in high-noise conditions, can *lose contact* with their primary target and can stick to some local maxima of the internal/external energy. On the other hand, very interesting results were obtained for automatic segmentation, even in presence of nested contours, by using *level-set methods* (Malladi et al., 1995).
An interesting unifying approach to segmentation is described in (Malladi & Sethian, 1996), where a class of constrained clustering algorithms for boundary extraction (as a generalization of known algorithms) is introduced. With T-snakes (McInerney, 1997), also the conventional snake approach was extended to provide the ability of splitting and merging. These algorithms generally seem to suffer from an intrinsic high computational complexity and from an effect of *contours smoothing* which can be undesired.
In (Iannizzotto & Vita, 1996) and, later, in (Iannizzotto & Vita, 2000), a new kind of active contour was introduced: this is composed by a chain of autonomous agents (MOVing elements: MOVels), which move independently but in a collaborative fashion over the image, according to some very simple rules and some image features. The idea of exploiting both homogeneity and non-homogeneity as pixel feature for image segmentation appears very attractive to overcome (at least, partly) the problem of noise sensitivity. In (Jones & Metaxas, 1998) an attempt to combine active contours with deformable models is made, but the process is split into two distinct steps: first edge detection, accomplished by means of a similarity-based function; then, a curve fitting process is applied to the resulting binary image, by initializing a balloon-like deformable model (Cohen, 1991) inside each contour and letting it inflate and fit the contour itself. In (Zhu & Yuille, 1996) region-growing and balloon-based approaches are unified in a common framework relying, in order to perform energy minimization, on a competition-based technique. A segmentation algorithm is introduced, based on a basic competitive learning approach according to the classification given in (Theodoridis & Kotroumbas, 1999), integrated with a probabilistic, bayesian decision criterion instead of the common similarity distance, and with a region-merging extension. The described approach assumes that the probability distribution of the point features are gaussian: this is usually not true. In their paper, the authors actually point out this problem, while enforcing the generality of their results for *any* probability distribution. However, no evidence is provided of this generality, and large part of the theoretical results seems to hold only for gaussian distributions. Finally, substantial prior information is exploited and needed, as prior probability distribution for the bayesian decision approach.
Recently, a different approach was introduced, which exploits autonomous agents randomly spread throughout the image (Liu & Tang, 1999). An agent is positioned in an area which is non-homogeneous (in a sense which is defined in the paper), it moves toward

a homogeneous area. When it finds it, it breeds, producing new agents which will gradually cover this area. If an agent cannot find an homogeneous area, it doesn't breed and, after a given lifetime, it dies. The overall effect is that after a number of life-cycles, all the pixels in the image will be visited and classified, thus producing a segmentation. Moreover, since the main target of the agents is breeding, and breeding needs space for the offspring, in some sense the agents exhibit a competitive behaviour.

When strong CPU power consumption constraints must be met, and high computation speed is mandatory (real-time processing) advanced computing resources cannot be used and so it is preferable to adopt custom hardware.

An alternative approach to image processing is provided by the *Cellular Neural Network* (CNN) paradigm, introduced by Prof. L.O. Chua in 1988 (Chua & Yang, 1988a; 1988b). A CNN consists of a network of first order nonlinear circuits, locally interconnected by linear (resistive) connections. CNNs have been extensively used in image processing applications (Matsumoto & Yokohama, 1990) such as filtering, edge detection, character recognition (Szirànyi & Csicsvàri, 1993) and object recognition (Milanova & Buker, 2000). Thanks to their architecture they can be applied to inherently parallel problems in which traditional methods cannot achieve a high throughput (Manganaro et al., 1999).

Various approaches to implementing real-time segmentation techniques on CNNs have been proposed (Rekeczky, 1999; Kozek & Vilarino, 1999; Vilarino et al., 2003). In (Rekeczky, 1999) "bias controlled trigger-waves" are used to determine the edge of an object in a scene, without, however, solving the problem of searching for nested objects.

(Kozek & Vilarino, 1999) and (Vilarino et al., 2003) proposed an image segmentation strategy based on either a continous or discrete-time CNN architecture, capable of revealing any nested objects in a scene, but the level of accuracy of the edges extracted was not investigated.

In the past, a still image segmentation technique (Iannizzotto et al., 2003) was developed, based on an active contour obtained via single-layer CNNs. The contour initially laid on the frame of the image shrinks, deforms and multiplies until it matches the edges of each of the objects present in the scene. The shape of each object in the image is accurately extracted and nested objects, if any, are correctly detected. Again, this technique suffer from sensitivity to noise as in the most of edge-based methods; noise may create insignificant false edges or determine some "edge fragmentation".

The aim of this work is to re-formulate the algorithm proposed within (Iannizzotto et al., 2003) in order to step-over the weakness of this, and other similar, works. The technique accurately traces the edges of objects, nested at various levels, even in presence of false (or fragmented) edges.

The input to the system is a gray-scale image obtained by applying to the image a median filter and a gradient operator. Guided by statistical properties of *edgeness* of the image pixels, the chain adapts its shape to that of the objects in the image until it marks out their contours. The output is a set of closed chains of points, each representing the contour of a single object. In the following sections we will describe the techniques developed and present a set of experimental results, laying particular emphasis on the evaluation technique used. In the final section we will draw our conclusions on the work carried out and discuss future lines of research.

2. Cellular Neural Network

As stated in the introduction, a CNN consists of an array of non-linear, locally interconnected, first order circuits. As connections are local, each cell is connected only to the cells belonging to its neighbourhood, as it is shown in Fig.1.

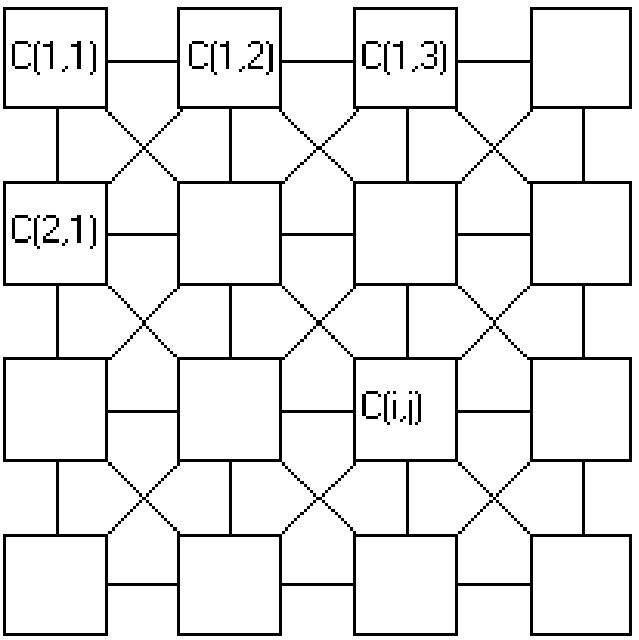


Fig. 1. Architecture of a CNN

If we call the generic cell in the *MxN* array as C_{ij} (the cell on the *i*-th row and the *j*-th column of the array), a formal definition of the neighbourhood of radius *r* of the cell C_{ij}, $N_r(i,j)$, is given by:

$$N_r(i,j) = \{C_{kl} : max\{|k-i|,|l-j|\} \le r, 1 \le k \le M, 1 \le l \le N\} \tag{4}$$

An *MxN* CNN, with *MxN* cells arranged in *M* rows and *N* columns, is entirely characterized by a set of *MxN* nonlinear differential equations, associated with each cell. The generic cell x_{ij} is described by the following relations:

$$\begin{aligned} C\frac{dv_{x_{ij}}(t)}{dt} &= -R^{-1}v_{x_{ij}}(t) + \sum_{kl\in N_r} A_{ij,kl}v_{y_{kl}}(t) + \sum_{kl\in N_r} B_{ij,kl}v_{u_{kl}}(t) + I_{ij} \\ &+ \sum_{kl\in N_r} A1_{ij,kl}(\Delta v_{yy}) + \sum_{kl\in N_r} B1_{ij,kl}(\Delta v_{uu}) + \sum_{kl\in N_r} D_{ij,kl}(\Delta v) \\ v_{y_{ij}}(t) &= f(v_{x_{ij}}(t)) = 0.5(|v_{x_{ij}}(t)+1| - |v_{x_{ij}}(t)-1|) \end{aligned} \tag{5}$$

where:

$$\Delta v_{yy} = v_{y_{kl}}(t) - v_{y_{ij}}(t)$$
$$\Delta v_{uu} = v_{u_{kl}} - v_{u_{ij}}$$
$$\Delta v = v_{u,x,y_{kl}}(t) - v_{u,x,y_{ij}}(t)$$
$$\left|v_{x_{ij}}(t)\right| \le 1, \left|v_{u_{ij}}\right| \le 1, \left|I_{ij}\right| \le v_{max}$$
$$1 \le i \le M, 1 \le j \le N \tag{6}$$

where $v_{x_{ij}}, v_{u_{ij}}, v_{y_{ij}}$ are respectively the state, input and output voltage of the CNN cell C_{ij}.

The state and output vary in time, whereas the input is kept constant. The indexes *ij* refer to the position of the cell in the 2D grid, while $kl \in N_r$ is a grid point in the neighborhood within the radius *r* of the cell *ij*. Matrices ***A***, ***B***, ***A1***, ***B1***, ***D***, called *templates*, describe the interaction of the cell with its neighbourhood and regulate the evolution of the CNN state and output vectors. Template connections can be realised by voltage-driven current generators.

$A_{ij,kl}$ is called linear feedback template, $B_{ij,kl}$ the linear control template, I_{ij} is a current bias in the cell. $A1_{ij,kl}$, $B1_{ij,kl}$ and $D_{ij,kl}$ are nonlinear templates respectively applied to Δv_{yy}, Δv_{uu} and Δv. $A1_{ij,kl}$ is called difference controlled nonlinear feedback template, $B1_{ij,kl}$ is the difference controlled nonlinear control template, $D_{ij,kl}$ is the generalized nonlinear generator. The output characteristic *f* adopted is a sigmoid-type piecewise-linear function.

CNNs are exploited for image processing by associating each pixel of the image to the input or initial state of a single cell. Subsequently, both the state and output of the CNN matrix evolve to reach an equilibrium state. The evolution of the CNN is governed by the choice of the template. A lot of templates have already been defined in order to perform basic image processing operations, like gradient computation, smoothing, hole detection, line deletion, isolated pixel extraction and deletion, and so on. Simple operations can be performed just by using the basic templates ***A***, ***B***, and the bias ***I***, whereas more complicated processing requires the use of the nonlinear templates ***A1***, ***B1***, and the generalized nonlinear generator ***D***. The proposed algorithm can be totally implemented onto a "*CNN Universal Machine*" (CNN-UM), an hardware structure able to implement CNNs (Chua & Roska, 1993).

The main advantage of using CNNs in image processing is related to the increasing of throughput due to the massive parallelism of the structure, joined to the similar way of signal processing, typical of CNNs. In fact they are able to perform a complete image processing analysis in time of order of 10^{-6} s (by using a CNN hardware implementation), this in form of sequences of simple tasks like array target segmentation, background intensity extraction, target detection and target intensity extraction.

Depending on the type of neurons that are basic elements of the network, it is possible to distinguish continuous-time CNN (CTCNN), discrete-time CNN (DTCNN) (oriented especially on binary image processing), CNN based on multi-valued neurons (CNN-MVN) and CNN based on universal binary neurons (CNN-UBN). CNN-MVN makes possible processing, which is defined by some multiple-valued threshold functions, and CNN-UBN allows processing defined not only by threshold, but also by arbitrary Boolean function.

3. Proposed Strategy

In the algorithm presented the input to the system (a *continous-time single layer CNN*) is a gray-scale image processed by applying to the original image median and gradient operators. Guided by statistical properties of *edgeness* of the image pixels, obtained during preprocessing phase, the chain adapts its shape to that of the objects in the image until it marks out their contours. A basic block diagram of this algorithm is shown in fig. 2.

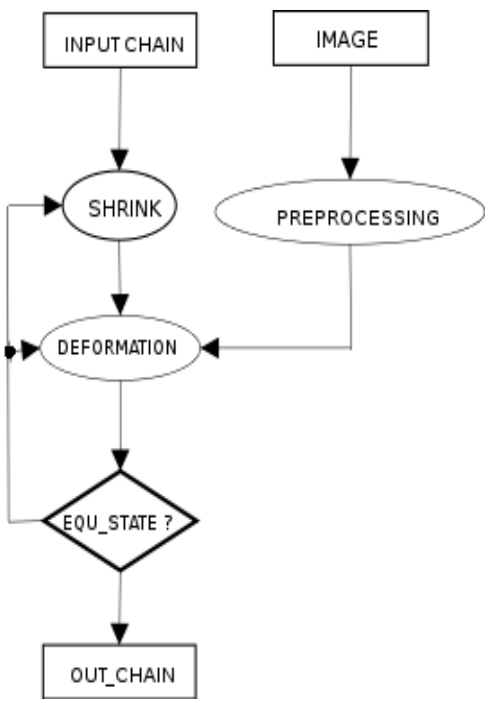


Fig. 2. Basic diagram of the algorithm

The original image is segmented via iterative shrinking and deformation of a chain, initially laid on the frame of the image. The chain shrinks across the whole image so as to reveal all the objects present, and deforms in order to adapt to the objects detected. The chain comprises a set of pixels, arranged over the image in such a way as to form a closed chain. The sequence of operations adopted to shrink the chains is shown in fig. 3.

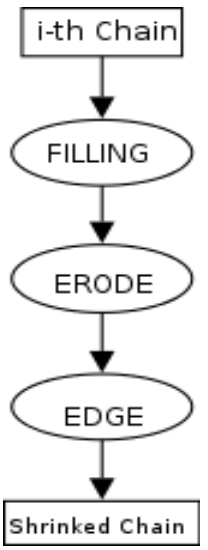


Fig. 3. Basic diagram of the shrinking process

The chain is initialised on the image to ensure that it contains all the objects in the scene. Once laid, the chain undergoes an iterative process of shrinking and deformation in order to adapt itself to the borders of the objects in the image. Shrinking occurs maintaining the shape of the input chain, while deformation is obtained by combining the information from statistical properties of edgeness (mean and standard deviation) of the image pixels and a binary image, result of preprocessing (see fig. 4) the original image. The iteration stops when a steady state is reached, i.e. the chain can't move any further.

The statistical properties (*mean and standard deviation* computed on 5x5 neighbourhood) are obtained applying templates suggested in (Moreira-Tamayos & Gyvez, 1999) on the edgeness image. "Mean" and "standard deviation" images, just obtained, are then combined through a weighted sum, as shown in fig. 5, by means of standard sum and product operation.

When a point of the chain meet a pixel of the image which features a very high value of *edgeness* this means that this point belongs to some object's border, so the chain should stick to this point. The point of the chain will therefore be "disabled". At each step and for each point in the chain (MOVel) is computed a functional, if its value exceeds a fixed threshold the MOVel is disabled. This functional depend on neighbour edge points number and on pixel statistical properties. The functional is computed and thresholding is applied in the same step by means of templates shown in eq. 7.

This operation return a map of points of the chain that have to be disabled. Each extracted point presents three characteristics:

- high edgeness
- belong to a region with high average edgeness
- edgeness similar to that of its neighbours

In fig. 6 we can see input (Feature-1), bias (Feature-2), mask (i-th chain) and output (disabled points) of the operation just described. The threshold (implicit) depend on adopted parameters (see eq. 7).

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

$$B = \begin{vmatrix} d & d & d & d & d \\ d & c & c & c & d \\ d & c & 0 & c & d \\ d & c & c & c & d \\ d & d & d & d & d \end{vmatrix} \tag{7}$$

$$I = Feature1$$

During its evolution, the chain may contain separate, not nested, objects to be detected. If, during iteration, non-adjacent points of the chain overlap, the chain splits into two chains, which continue to evolve independently of each other. To detect the presence of nested objects, if any, a *daughter* chain is generated inside each contour obtained, and the operations mentioned above are repeated on this chain. The *daughter* chains evolve until they reach the contours being sought or, if they do not contain any objects, implode and disappear. The search for nested objects is resumed whenever moving chains reach their steady state. It ends when all the moving chains have imploded.

This means that all the objects in the scene have been detected, making any further search useless.

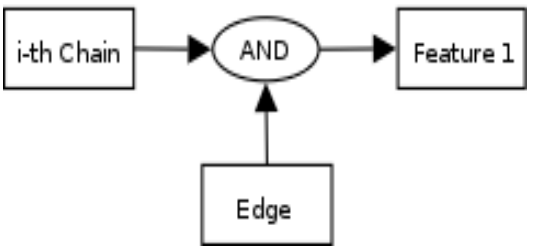


Fig. 4. Feature-1 extraction

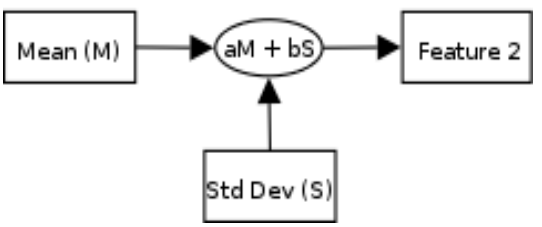


Fig. 5. Feature-2 extraction

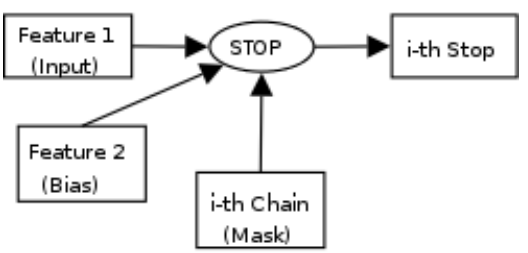


Fig. 6. Stopping phase

4. Accuracy Evaluation

Characterizing the performance of image segmentation approaches has been a persistent challenge. Performance analysis is important since segmentation algorithms often have limited accuracy and precision.

For some applications (e.g. medical images analysis), interactive drawing of the desired segmentation by domain experts has often been the only acceptable approach, and yet suffers from intra-expert and inter-expert variability. Automated algorithms have been

sought in order to remove the variability introduced by experts, but no single methodology for the assessment and validation of such algorithms has yet been widely adopted.
An automated algorithm is compared to the segmentations generated by a group of experts, and if the algorithm generates segmentations sufficiently similar to the experts it is regarded as an acceptable substitute for the experts.
The most appropriate way to carry out the comparison of an automated segmentation to a group of experts segmentations is so far unclear. A number of metrics have been proposed to compare segmentations, including volume measures, spatial overlap measures, such as Dice (Dice, 1945) and Jaccard similarities (Jaccard, 1912), and boundary measures, such as the Hausdorff measure (Huttenlocher et al., 1993). Agreement measures between different experts have also been explored for this purpose. Studies of rules to combine segmentations to form an estimate of the underlying true segmentation have as yet not demonstrated any one scheme to be much favourable to another.
We present here a new algorithm for estimating the ground truth segmentation from a group of experts segmentations. Then, we employ the estimated ground truth to assess and validate the results of our segmentation technique. To estimate a ground truth we use a technique known as Active Shape Model (*ASM*) with the aim to synthesize a model representative of a training set (segmentations generated by a group of experts).
For this technique (Cootes & Taylor, 1992) the shape of an object is represented by a set of n points, which may be in any dimension. Commonly the points are in two or three dimensions.
The training set typically comes from hand annotation of a set of training images through landmarking. Good choices for landmarks are points which can be consistently located from one image to another: in two dimensions points could be placed at clear corners of object boundaries, "T" junctions between boundaries or easily located biological landmarks. This list would be augmented with points along boundaries which are arranged to be equally spaced between well defined landmark points. By analysing the variations in shape over the training set, a model is built which can mimic this variation.
If a shape is described by n points in d dimensions we represent the shape by a nd element vector formed by concatenating the elements of the individual point position vectors. For instance, in a 2-D image we can represent the n landmark points, $(x_i; y_i)$, for a single example as the $2n$ element vector, $\mathbf{x}$, where

$$x = \left(x_{1,} \ldots, x_n, y_{1,} \ldots, y_n\right)^T \tag{8}$$

Given s training examples, we generate s such vectors $\boldsymbol{x}_j$. These vectors form a distribution in the nd dimensional space in which they live. If we can model this distribution, we can compare the model obtained in such a way with the segmentation result of our system processing. In particular we seek a parameterized model of the form x = M(b), where b is a vector of parameters of the model.
Through Principal Component Analysis (PCA) we build a model of the object shape to segment obtaining also a dramatic reduction in size of the training set data. To obtain the model we execute the following steps:

1. Compute the mean of the data,

$$\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i \tag{9}$$

2. Compute the covariance of the data,

$$S = \frac{1}{s-1}\sum_{i=1}^{s}(x_i - \bar{x})(x_i - \bar{x})^T \tag{10}$$

3. Compute the eigenvectors, φ_i and corresponding eigenvalues λ_i of **S** (sorted so that $\lambda_i \leq \lambda_{i+1}$).

If **Φ** contains the *t* eigenvectors corresponding to the largest eigenvalues, then we can approximate any of the training set, **x** using

$$x \approx \bar{x} + \Phi b \tag{11}$$

where **Φ** = ($\varphi_1 | \varphi_2 | \ldots | \varphi_t$) and **b** is a *t* dimensional vector given by

$$b = \Phi^T (x - \bar{x}) \tag{12}$$

The vector **b** defines a set of parameters of a deformable model. By varying the elements of **b** we can vary the shape, **x** using Equation 11. The variance of the ith parameter, b_i, across the training set is given by λ_i. By applying limits of $\pm 3\sqrt{\lambda_i}$ to the parameter b_i we ensure that the shape generated is similar to those in the original training set.

The number of eigenvectors to retain, *t*, can be chosen so that the model represents some proportion (e.g. 98%) of the total variance of the data, or so that the residual terms can be considered noise.

To estimate the quality of our system applied on a still-image, the results obtained by a group of experts have been collected and then used as training set to build an ASM.

Once built the model, this is compared with the result of our system. The comparisons have been done using a normalized version (with respect to the number of selected landmarks) of the Mahalanobis distance.

Fig. 7 shows an image selected from the test set alongside the relative segmentation images. Fig. 8 and fig. 9 show respectively the segmentation image manually obtained by a human operator and the one produced by our algorithm. In this case the normalised error is equal to 0,2. This is an intermediate value between those obtained but, as direct comparison shows, the resulting segmentation is visually acceptable.

Fig. 7. An image selected from the test set

Fig. 8. Manual segmentation

Fig. 9. Automatic segmentation

5. Experimental Results

In order to show the validity of the proposed algorithm, we provide the results obtained on the same set of test images used in (Iannizzotto et al., 2003). Accurate measures were performed on a set of 20 gray-scale images, specially selected for their contents. As mentioned in section 4, the method chosen to evaluate the results obtained by the algorithm is based on a comparison between the segmentation image obtained automatically and the estimated ground truth obtained as previously described (see par. 4). In Fig. 10 a selection of the test images is shown. Figs. 11 and 12 respectively show the segmentation images obtained by a group of experts and those obtained by applying the algorithm being proposed. Fig. 13 is a plotting of the error produced by our algorithm against the processed image. As the graph in Fig. 13 shows, the error on the test images is bounded around an average value of 0.2 in comparison with 0.3 obtained applying the same validation

technique to the algorithm described in (Iannizzotto, 2003). These results show a reduction of the average error for the segmented images. It is due both to the used metric, which is less sensitive to impulsive noise, and to an actual improvement of the algorithm performances. In fact, the use of median operator has got rid of some peak in the error trend (Iannizzotto, 2003), caused by noise in the image, and the use of statistical features has made possible a "generalized" reduction of the error.

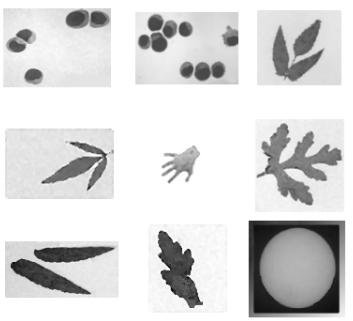


Fig. 10. A selection of the test images

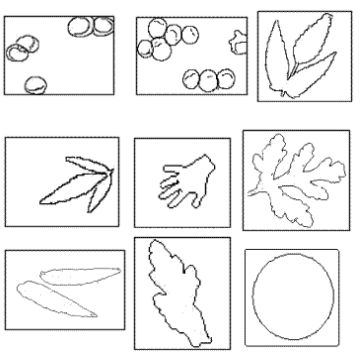


Fig. 11. Manual segmentation of the test images

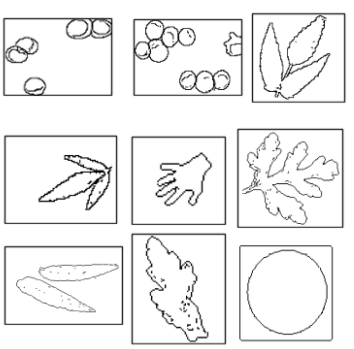


Fig. 12 Automatic segmentation of the test images

6. Competitive Approach

The technique we proposed, although effective in its results, is still affected by some parameter dependences: thresholds and weighting values used during computation. A possible solution, is an approach to image segmentation based on competing chains. Each chain acts as a competitive active contour which reacts to image features. Our work aims at producing a framework in which image segmentation is performed without any user input (namely, *unsupervised segmentation*) and with the minimum amount of prior information.

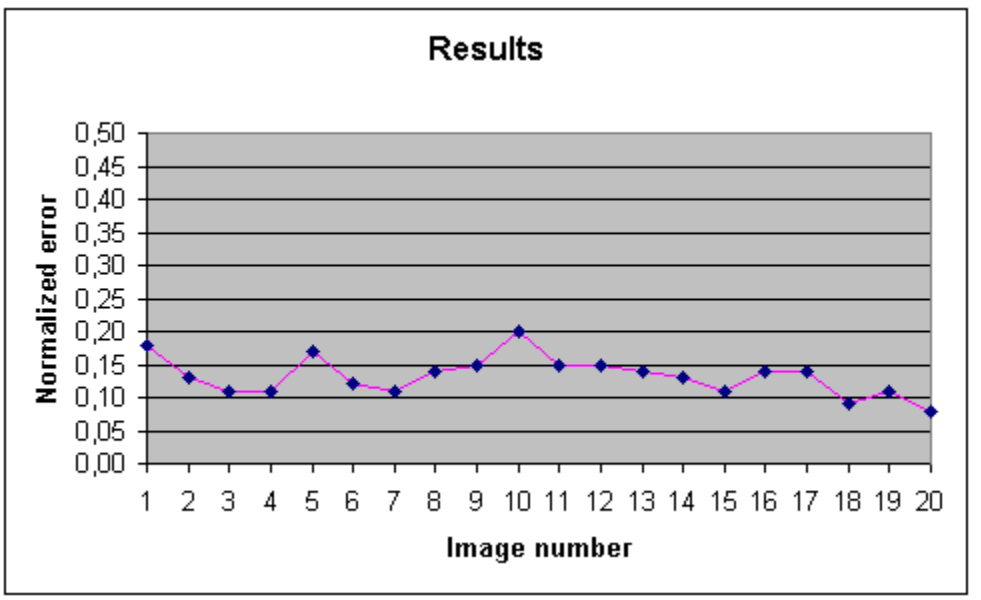


Fig. 13. Graph of results obtained

The competition-based approach will heavily reduce the influence of initialization on the final result (Zhu & Yuille, 1996).

In the following we outline the competition-based approach through a brief description of our algorithm.

At initialization time, K chains are generated and uniformly spread over the image. The number K depends on the size of the image and will usually be quite large to correctly segment the image independently of the initial position of the chains.

After the initialization, each chain grows in size until it meets another chain or a high edgeness contour. In the latter case, if the contour is closed and surrounds all the chain, the chain sticks to it, stopping its growth process and *breeds*, generating a new chain which will grow beyond the contour. This process allows the algorithm to detect multiple nested objects as *chain hierarchies*.

If the chain meets another chain, they start competing for the territory (i.e. an area in the image), and after a finite time a steady state will be reached. One of the two chains will probably "conquer" some part of the territory, until some line will be found, composed of

pixel which are equidistant from both the chains. This line will be the border between the two chains.
At each step the pixel gray-level mean, a feature representative of all pixels surrounded by the chain, is estimated. The chains compete for the territory based on similarity between their "mean" and the gray-level of the point "to conquer". It cannot happen that one of the chains totally defeats the other, since at least the original area of a chain (i.e. the one surrounded by the chain at initialization time) will always match better its "statistic". But if two chains lay on the same, uniform, area then they will have the same statistic: in this case, as soon as they meet, they *merge*, thus producing a larger chain with the same statistic. The sequence of operations adopted to let the chains grow is shown in fig. 14.

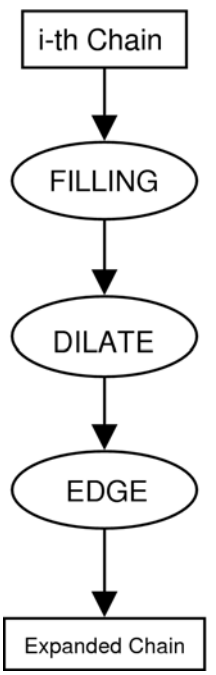


Fig. 14. Expansion phase

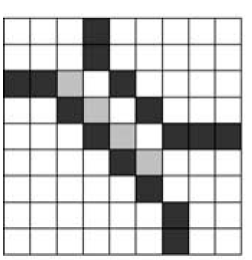


Fig. 15. Collision point detection

The detection of the collision point between chains is obtained using the approach proposed in (Vilarino et al., 2003).
An example of collision detection between two chains and merging chains, automatically handled by the algorithm, is shown in fig. 15.
Fig. 16 is a plotting of the error produced by our algorithm against the processed image. As the graph in Fig. 16 shows, the error on the test images is bounded around an average value of 0.18 in comparison with 0.2 obtained applying the same validation technique to the algorithm described in section 5.

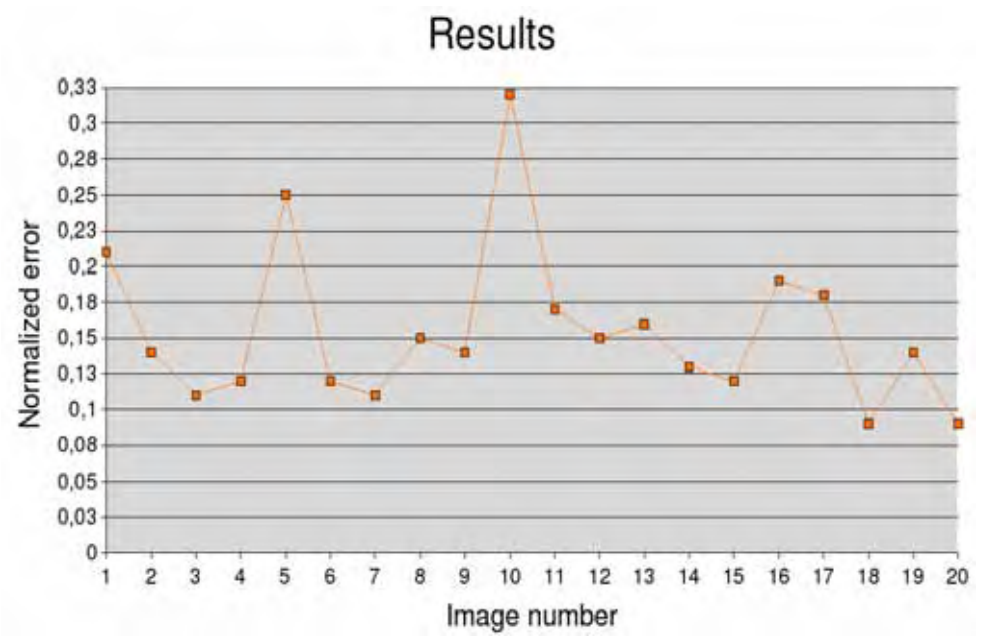


Fig. 16. Graph of results obtained

7. Conclusion

In this work we have described a re-formulation of a 2D still-image segmentation algorithm, implemented on a single-layer CNN, previously proposed (Iannizzotto, 2003). This algorithm is able to step-over limitation inherent to the class of active contours: sensitivity to insignificant false edges or "edge fragmentation". The approach features an iterative process of uniform shrinking and deformation of the active contour. Guided by statistical properties of *edgeness* of the image pixels, the chain adapts itself to the image contours. Undesirable smoothing of the edges of the objects are prevented by the absence of any particular rigidity constraints on the chain. The technique used for uniform shrinking, which automatically handles any splitting, allows the presence of any nested edges to be detected.

Experimental measures of the accuracy of the segmentation were carried out using a technique based on Active Shape Models. Finally, an alternative competition-based approach, used to reduce some parameter dependences, is outlined in section 6.

8. References

Abe, T. & Matsukawa, Y. (2000). A region extraction method using multiple active contour models,*Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR2000)*

Chua, L. & Yang, L. (1988). Cellular neural networks: Theory, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, (1988), pp. 1257–1272

Chua, L. & Yang, L. (1988). Cellular neural networks: Applications, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, (1988), pp. 1273–1290

Chua, L. & Roska, T. (1993). The cnn universal machine: An analogic array computer, *IEEE Trans. Circuits and Systems II*, Vol. 40, (1993), pp. 163–173

Cohen, L. (1991). On active contour models and balloons, *CVGIP: Image Understanding*, Vol. 53, No. 2, (1991), pp. 211–218

Dice, L. R. (1945). Measures of the amount of ecologic association between species, *Ecology*, Vol. 26, No. 3, (1945), pp. 297–302

Flickner, M.; Hafner, J.; Rodrìguez, E. J. & Sanz, J. L. C. (1996). Periodic quasi orthogonal spline bases and applications to least-squares curve fitting in digital images, *IEEE Transactions on Image Processing*, Vol. 5, No. 1, (Jan 1996), pp. 71–88

Frigui, H. & Krishnapuram, R. (1999). A robust competitive clustering al gorithm with applications in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, (May 1999),pp. 450–465

Huttenlocher, D.; Klanderman, G. & Rucklidge, A. (1993). Comparing images using the hausdorff distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, (September 1993),pp. 850–863

Iannizzotto, G. & Vita, L. (1996). A fast accurate method to segment and retrieve object contours in real images, *Proceedings of International Conf on Image Processing*, Lausanne, Switzerland

Iannizzotto G. & Vita L. (2000). Fast and accurate edge-based segmentation with no contour smoothing in 2-d real images, *IEEE Trans. on Image Processing*, July 2000

Iannizzotto, G.; La Rosa, F.; Rizzo, A. & Xibilia, M. (2003). 2d still-image segmentation with cnn-amoeba, *Proceedings of International Work-shop on Computer Architectures for Machine Perception*, New Orleans

Jaccard, P. (1912). The distribution of flora in the alpine zone, *New Phytologist*, Vol. 11, pp. 37–50

Jones, T. & Metaxas, D. (1998). Image segmentation based on the integration of pixel affinity and deformable models, *Proceedings of CVPR98*, pp. 330–337

Kass, M.; Witkin, A. & Terzopoulos, D. (1988). Snakes: Active contour models, *International Journal of Computer Vision*, Vol. 1, pp. 321–333

Kozek, T. & Vilarino, D. L. (1999). An active contour algorithm for continuous-time cellular neural networks, *Journal of VLSI Signal Pro- cessing Systems*, Vol. 23, No. 2-3, (1999), pp. 403–414

Lai, K. & Chin, R. (1995). Deformable contours: Modeling and extraction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 11, pp. 1084–1090, November 1995.

Liu, Y. & Tang, Y. (1999). Adaptive image segmentation with distributed still-image segmentation algorithm, implemented on a single behaviour-based agents, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 6, (June 1999), pp. 544–551

Malladi, R.; Sethian, J. A. & Vemuri, B. (1995). Shape modelling with front example of collision between two chains is shown. An example propagation, *IEEE Trans. on PAMI*, Vol. 17, No. 2, (February 1995)

Malladi, R. & Sethian, J. A. (1996). A unified approach to noise removal, image enhancement,and shape recovery, *IEEE Trans. on Image Processing*, Vol. 5, No. 11, (November 1996)

Manganaro, G.; Arena, P. & L. Fortuna (1999). *Cellular Neural Networks: Chaos, Complexity and VLSI processing*, Springer-Verlag

Matsumoto, T.; Yokohama, T.; Suzuki, H. & Furukawa, R. (1990). Several image processing examples by cnn, *Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, Budapest

McInerney, T. J. (1997). Topologically Adaptable Deformable Models. *Ph.D Thesis*, University of Toronto

Milanova, M. & Buker, U. (2000). Object recognition in image sequences with cellular neural networks, *Neurocomputing*, Vol. 31, No. 1-4, pp. 125–141

Moreira-Tamayos, O. & Gyvez, J. P. D. (1999). Subband coding and image compression using cnn, *International Journal of Circuit Theory and Applications*, No. 27, pp. 135–151

C. Rekeczky (1999). Active contour and skeleton models in continous-time cnn, *Proceedings of ECCTD '99*, Stresa, Italy

Szirànyi, T.; & Csicsvàri, J. (1993). High-speed character recognition using a dual cellular neural network architecture (cnnd), Analog and Digital Signal Processing, Vol. 40, No. 3, pp. 223–231

Theodoridis, S. & Kotroumbas, K. (1999). *Pattern Recognition*. Academic Press

Vilarino, D. L.; Cabello, D.; Pardo, X. M. & Brea, V. M. (2003). Cellular neural networks and active contours: a tool for image segmentation, *Image Vision Computing*, Vol. 21, No. 2, pp. 189–204

Zhu, S. & Yuille, A. (1996). Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,Vol.18, No. 9, (September 1996), pp. 884–900

7

Optimizing Mathematical Morphology for Image Segmentation and Vision-based Path Planning in Robotic Environments

Francisco A. Pujol, Mar Pujol & Ramón Rizo
University of Alicante
Spain

1. Introduction

Robotics advances have generated an increasing interest in new research projects and developments. Nowadays this science has several new applications characterized by working in non-structured dynamic environments. As a result, the research on this emerging area is growing, and, specially, vision algorithms are constantly being improved.
In many cases, navigation needs real-time answers. As robots often work in dynamic environments, it would be desirable that the system takes a decision and applies it before external conditions change.
Much work has been done on solving the problem of planning shortest paths between different locations within an environment (also known as a workspace) scattered with obstacles. For these solutions, the obstacles are usually considered as solid objects, and a collision-free path (of possibly shortest distance) must be found to navigate around them. However, not all path planning applications can be modelled as such a problem.
On the other hand, Mathematical Morphology (MM) is a useful tool in image analysis, commonly used to extract components of the image, like contours, skeletons and convex forms. Although there are some approaches that take into account topographical maps in order for a robot to navigate through a workspace, few approaches actually deal with Mathematical Morphology operations.
In this chapter we will focus on some of the research that we have completed in this field in the last few years. This way, two different robotic MM-based applications are discussed:

- Path planning, which is strongly influenced by the precision of the acquisition process. Thus, it can be modified both by the quality of the information obtained from the environment, and the attributes of the system and the environment in which it works. Here, we shall refer to vision-based path planning.
- Image segmentation, which is an essential part of any intelligent system, since it is necessary for further processing such as feature extraction or object and face recognition, among others.

The research work here described has obtained very good experimental results and would contribute to the development of practical recognition and path planning systems. The use of vision improves the system, since once the visual information has been interpreted in

order to provide a basic world representation, then the objects may be modelled to determine a free path.

2. Mathematical Morphology Overview

The term *Mathematical Morphology* commonly denotes a branch of Biology that deals with the shape and the structure of plants and animals. We use here the same word as a tool to extract image components that are useful in the representation and description of regions, such as contours, skeletons and convex forms.
The basis of Mathematical Morphology is the set theory. Sets in Mathematical Morphology represent the shapes of the objects in an image. The morphological operations are based, therefore, in geometric relations between the points of such sets.
This discipline focuses on the morphological transformations of images, i.e., erosion, dilation and its combinations, when some local operators, called structuring elements, are applied. The shape and the position of the origin of the structuring elements have a decisive influence on the final result of the morphological operator (Serra, 1992).
Mathematical Morphology describes objects as subsets of the Euclidean space. The fundamental structure in Mathematical Morphology is the complete reticulum (Serra, 1982), that is, a set $\Re$ where for all the elements $\{X_i\} \in \Re$, two fundamental laws exist: the supremum (sup) or the minimum upper level ($\vee\{X_i\}$) and the infimum (inf) or the maximum lower level ($\wedge\{X_i\}$). This structure explains the most common processes used in Mathematical Morphology. A binary image can be modelled as a set belonging to a boolean grid. A gray-level image is modelled as a function belonging to the set of the upper semi-continuous functions.
The two basic operations defined in Mathematical Morphology, erosion and dilation, are described below.

Set Erosion

The erosion of a set X using a symmetrical structuring element B is the locus of the centre of the structuring element B, when B is included in X (Serra, 1982). It can be written as:

$$\varepsilon_B(X) = \{ y, \forall\, b \in B, y + b \in X \} = \bigcap_{b \in B} (X + b) \tag{1}$$

Set Dilation

The dilation of a set X by a symmetrical structuring element B is the locus of the centre of the structuring element B, when B hit X (Serra, 1982). It can be written as:

$$\delta_B(X) = \{ x + b, x \in X,\ b \in B \} = \bigcup_{b \in B} (X + b) \tag{2}$$

The erosion and the dilation are dual operations: the dilation of a binary image is the complemented erosion of the complementary image. The erosion of a binary image is the complemented dilation of the complementary image.
From these basic operations, dilation and erosion, more complex transformations are constructed, as opening (dilate the result of an erosion) and, its dual operation, closing (erode the result of a dilation), to implement basic filters.

3. Path Planning Applications of Morphological Filtering

3.1 A General Review of Path Planning

Robot path planning has proven to be a hard problem. There is strong evidence that its solution requires exponential time in the number of dimensions of the configuration space, i.e., the number of degrees of freedom (DOF) of the robot. This result is remarkably stable: it still holds for specific robots, e.g., planar linkages consisting of links serially connected by revolute joints (Joseph & Plantiga, 1985), and sets of rectangles executing axis-parallel translations in a rectangular workspace (Hopcroft & Wilfong, 1986). Though general and complete algorithms have been proposed (Canny, 1988), their high complexity precludes any useful application. This negative result has led some researchers to seek heuristic algorithms. While several of such planners solve difficult problems, they often fail or take much more computation times than simpler ones. The fact that their behavior is not well characterized is a major drawback: they cannot be used as black boxes in larger robot control systems.

Collision-free path planning, which assumes perfect knowledge of the world and stationary obstacles, is only the most basic motion-planning problem in robotics. Clearly, we would ultimately like robot planners to deal with issues such as uncertainties, moving obstacles, movable objects, and dynamic constraints. But every extension of the basic problem adds to computational complexity. For instance, allowing moving obstacles makes the problem grow exponentially with the number of moving obstacles (Canny, 1988). Before we can effectively investigate such extensions in large configuration spaces, it seems that we must better understand how to practically solve basic path planning.

Path-planning applications are so diverse that it is infeasible to design a tailor-made algorithm for every possible robot. Instead, we need general path-planning algorithms not bound to the specifics of any particular robot. We believe that between the two extreme types of planners suggested above –complete and heuristic– there is a place for practically efficient general planners achieving a weaker form of completeness. In other words, we may perhaps trade a limited amount of completeness against a major gain in computing efficiency. Full completeness requires the planner to always answer a path-planning query correctly, in asymptotically bounded time. A weaker, but still interesting form of completeness is the following: if a solution path exists, the planner will find one in bounded time, with high probability. We call it probabilistic completeness. This weaker completeness becomes particularly interesting if we can show that the planner's running time grows slowly with the inverse of the failure probability that we are willing to tolerate.

3.2 Layout of Paths

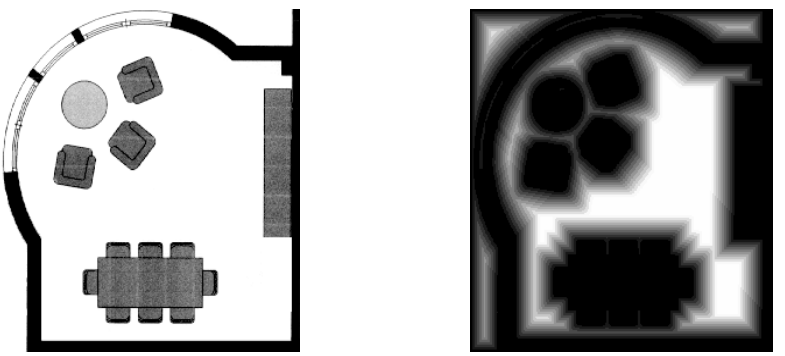
Let us consider an application of the morphological primitives that are described in section 2, a method for the accomplishment of maps for gray-tone images (that should have been captured by the robot camera) as a previous step to collision-free path planning. Supposing that the obstacles in the picture room are represented in dark tones, the point is that images that have to be processed should be represented in a proper way. In a real situation a perspective transformation ought to be made.

Let I be the original image and I' the result image, then the followed algorithm to draw up the land map of an image that shows a room, and using a dilation with a $n \times n$ SE, is the next one:

```
UI=mean(I)
threshold = UI
Binarize(I)
while (not  End condition) do
        Dilation(I, I')
        Change black tone to gray tone in I'
end while
```

Table 1. Land map algorithm

The result of this operation results in black objects surrounded by gray tones of greater intensity, until becoming into a white color, as we show in figure 1. White color is considered the tone in which the probability of collision with an object in a path followed by the robot is minimum. Consequently, in the 3D view (figure 2) the high zones would be a low risk for the robot.



(a) Picture of the room (b) Land map

Fig. 1. Land map for an image.

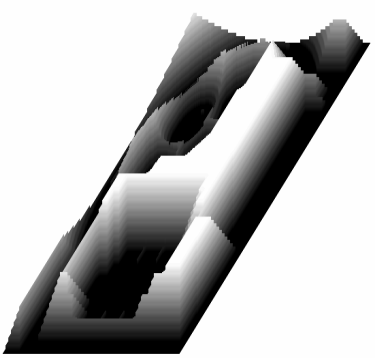


Fig. 2. 3D View of the land map.

Once determined the land map of a room, we want to show an example of high security trajectories that an autonomous vehicle could follow through it, to avoid the collision with the obstacles located in this workspace. So we are going to see some of the possible paths that the vehicle would decide to execute in figure 3, where o is the origin of the path and e is the end of the path. The algorithm is:

```
Select origin
while ((origin<>edge) OR (origin_value<>black)) do
Find maximum in neighborhood 3x3
if (maximum_value > origin_value)
origin = maximum
        else, if (maximum_value = origin_value)
origin = maximum
            else
                Follow path

end while
```

Table 2. Path planning algorithm

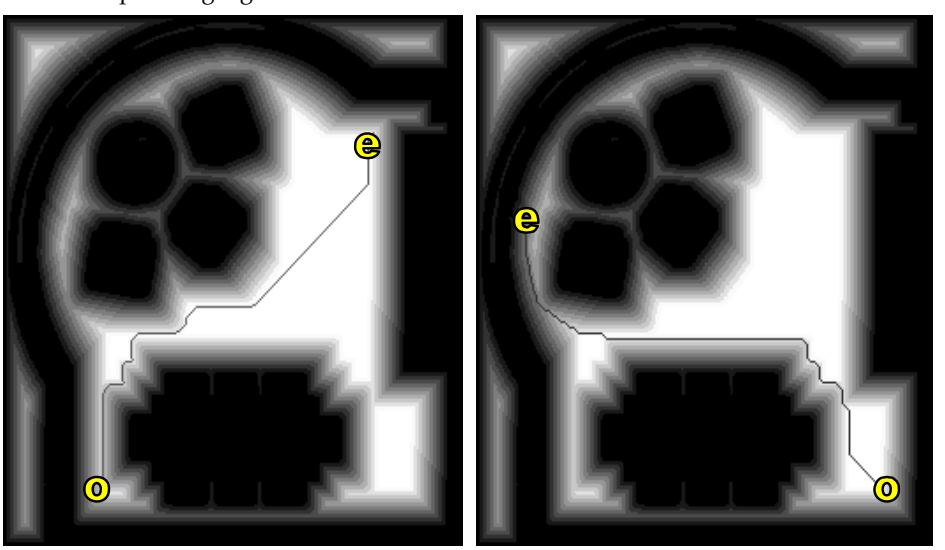


Fig. 3. Possible trajectories followed by a mobile robot.

3.3 Improving the Path Planning Algorithm

In this section we propose an improved Mathematical Morphology-based path planning algorithm. Let us consider that the vision-based system works with grey-scale images. First, a map that separates obstacles from free-space is obtained; this initial processing method can be described as:

1. Apply a Gaussian smoothing to the original image.
2. Associate a set of symbols for each pixel. These symbols are extracted from:
    a. The gradient of the image.
    b. The variance considering each pixel's neighborhood.
3. Merge the results by means of the creation of a new image, where lower intensity pixels represent a higher probability of being classified as an obstacle.
4. Binarize the image, where obstacle pixels are labelled as '0' and free-space pixels as '1'.
5. Repeat the following steps:
    a. Implement the SE-decomposed morphological dilation.
    b. Change black tones to a grey tone, increasing the grey intensity for each iteration.
6. Obtain a map where higher intensity pixels constitute the obstacle-free zones.

Table 3. Improved land map algorithm

As soon as this map is obtained, the path planning algorithm can be executed. From the starting point of the path, the algorithm chooses the pixel with the lowest probability of collision in a 3x3 neighbourhood; once selected, the first movement is performed. This choice is carried out considering two criteria:

- The closeness to the obstacles. It is preferred to move to positions with high intensity, as they have a low probability of collision.
- The accomplishment of the task. This criterion makes the robot follow its path towards the destination point in case there are several similar or equal intensity values in the neighbourhood.

Hence, the selection of the next pixel in the robot's path will be completed by using a normalized weight which ensures that the new pixel has the lowest probability of collision in the neighbourhood. To do this, we define a set of weights w_i that are estimated as:

$$w_i = e^{a*(dist_{old} - dist_{new})} \tag{3}$$

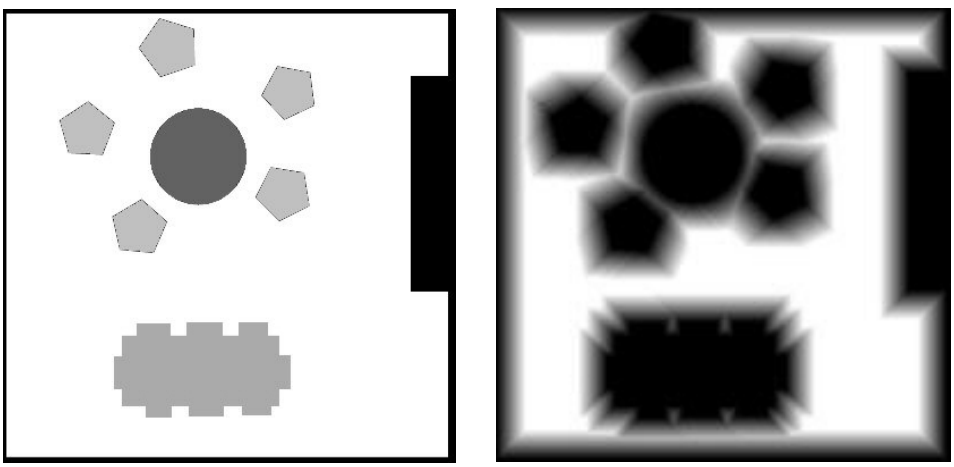
where $dist_{old}$ is the Euclidean distance from the current pixel to the destination one, $dist_{new}$ is the Euclidean distance from the selected new pixel to the destination one and a is a real constant, so that $0 \leq a \leq 1$.

As a consequence, the robot will move to the pixel with the highest weight w_i; this operation will be repeated until it arrives to the destination or there is some failure due to a collision with non-detected obstacles. Therefore, the higher factor a is, the bigger differences among the weights w_i exist; obviously, factor a is a critical element for the robot to follow an optimal path to the destination, as it provides the best weights w_i to complete the path planning task.

In relation to this, the following section analyzes practically how to achieve a fast, powerful operation in some example situations.

3.4 Experiments

Let us consider now the results of some experiments that will test the suitability of our model. First of all, Fig. 4 (a) shows a world created for the robot to wander throughout it. We assume that there is no perspective distortion; this situation occurs when the camera's optical axis is not perfectly perpendicular to the presentation surface. This distortion, which is part independent, can be compensated for using a homogeneous transformation.



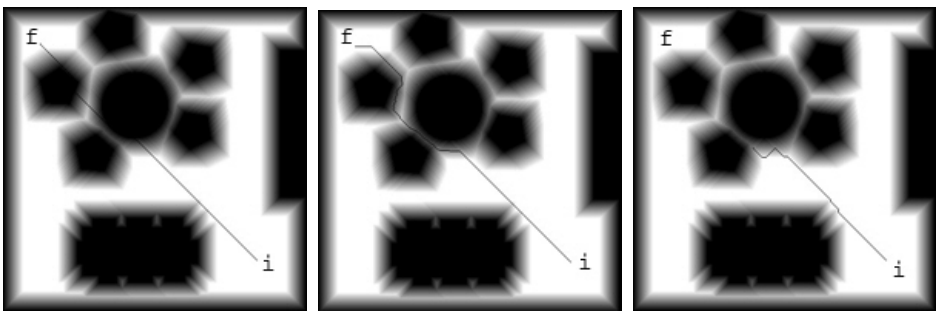
(a) (b)

Fig. 4. The environment for path planning: (a) World 1 (b) A morphological map

At this point, the algorithm developed before is applied; thus, the resulting map after the initial processing method, using a 3x3 square SE, is depicted in Fig.4 (b).

From this map, a trajectory is followed after defining its origin and destination pixels (i.e., locations in real world). To do this, the weights wi must be estimated using Eq. (3) and, additionally, factor a should be defined. In Fig. 5 some example paths for a 3x3 square SE (where factor a varies from 0.1 to 1.0) are shown. Note that i refers to the initial point of the path, and f indicates the final point of the path.

The analysis of these examples shows that if factor a has a high value (Fig. 5 (a)), the algorithm will select a path that easily reaches the destination point, although it is not collision-free as the approaching to the obstacles can be sometimes dangerous. This will lead to the incorporation of some other sensing capabilities to prevent from collision. On the contrary, when factor a has a low value (Fig. 5 (c)), the robot may stop before completing its task, since it is preferred not to move close to the obstacles. As a consequence, factor a has a better behaviour when it makes the robot follow an optimal or semi-optimal path that keeps it away from collision (in this example, a = 0.1, see Fig. 5 (b)).



(a) (b) (c)

Fig. 5. Some paths followed in the environment: (a) Factor a = 1.0 (b) Factor a = 0.1 (c) Factor a = 0.05.

Nevertheless, the choice of an appropriate factor will depend mainly on the map of the environment produced after the initial method. Moreover, there must be a training process to determine a well-suited factor before a real operation.

4. Image Segmentation with Mathematical Morphology

Designing an image segmentation scheme needs the consideration of two features of visual recognition: cost and uncertainty. Visual recognition is generally costly because the image data is large. Visual information contains uncertainty from many sources such as discretization. In general, the more observation a vision system performs, the more information is obtained, but the more cost is required. Thus, a trade-off must be considered between the cost of visual recognition and the effect of information to be obtained by recognition. In relation to this, some of the most popular approaches which provide low computation times and good information are the threshold techniques and the edge-based methods (Ouadfel & Batouche, 2003), (Pal, & Pal, 1993).

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects are exactly outside the range of the levels in the background. These thresholding algorithms are simple and give very good results, but deciding the threshold values is not easy. Specially, this is a really serious problem for an automated vision system, as the system should decide the threshold values taking its own decision.

On the other hand, edge-based methods (e.g., gradient operators) focus on contour detection. They involve finding the edges of objects in the image and using this edge information to achieve the complete boundaries for the main objects in the image. Edge detection has many problems, especially when working with noisy images, since it could even fragment the true edges.

To overcome these problems, we propose a method that combines both thresholding and gradient operators: the so-called Morphological Gradient Threshold (MGT) segmentation, as described in Table 4. It consists of 7 main steps, where the gradient and the Laplacian are calculated in terms of Mathematical Morphology operations and the optimal threshold value is selected by measuring the lowest distance between the ideal segmentation and a collection of MGT segmented images.

Step 1. Image smoothing.
Step 2. Global dilation and erosion.
Step 3. For every pixel, create a list of symbols by means of the Morphological gradient and the Morphological Laplacian.
Step 4. Creation of a pixel-symbol map.
Step 5. Binarization of the pixel-symbol map.
Step 6. Computation of a suitable measure to obtain the optimal threshold.
Step 7. Obtention of the MGT segmented image.

Table 4. MGT segmentation algorithm

4.1 Construction of a Pixel-Symbol Map

In every digital image there is a certain amount of white noise. To avoid the noise effects, which only consume computation time and affect the real image features, an initial filtering process has to be applied. There are many algorithms to accomplish this task; in our approach a Gaussian filter has been chosen, since it preserves many of the image features while its computational cost can be assumed in a real-time environment. For more information see (Basu & Su, 2001).

Once the noise is eliminated, the point is how to create the pixel-symbol map. To do this, let us consider first the computation of some derivative-based operations, i.e., the gradient and the Laplacian.

Edge detection is a main problem in image analysis. There are many approaches to obtain edges by means of the gradient of an image (e.g., Prewitt or Sobel operators). Among all of these methods we find the morphological gradient, which uses the Mathematical Morphology operators.

Therefore, one can define the morphological gradient of an image X by a structuring element (SE) B, $\rho_B(X)$, as:

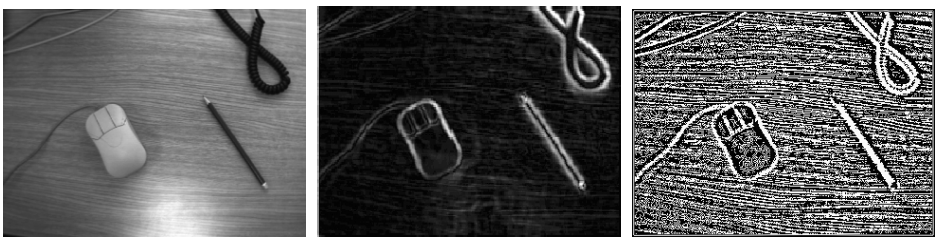
$$\rho_B(X) = \frac{1}{2}(\delta_B(X) - \varepsilon_B(X)) \tag{4}$$

where $\delta_B(X)$ and $\varepsilon_B(X)$ are, respectively, the dilation and the erosion of an image X by a SE B.

The following step is to calculate the second derivative, the Laplacian. Again, we have chosen a morphological implementation for the Laplacian, as we can use with costless time the previously pre-calculated erosion and dilation. Thus, the morphological Laplacian of an image X by a SE B, $\Lambda_B(X)$, is defined as:

$$\Lambda_B(X) = \frac{1}{2}(\delta_B(X) + \varepsilon_B(X)) - X \tag{5}$$

The results for a gray-scale image after these initial steps are shown in Fig. 6, where the SE B is a 3x3 square.



(a) (b) (c)

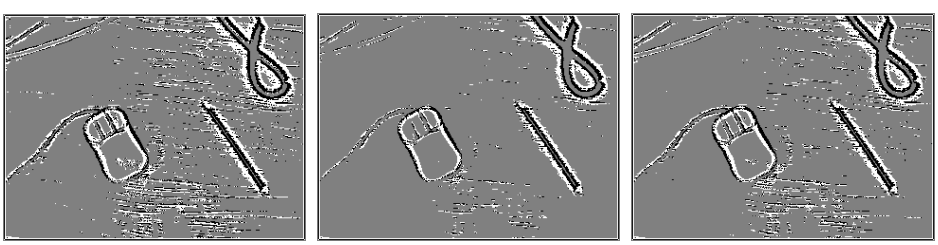
Fig. 6. A real image: (a) Original image. (b) $\rho_B(X)$. (c) $\Lambda_B(X)$.

The next task is building a map that characterizes properly the pixels for a good segmentation. Thus, the pixel-symbol map $m(x,y)$ is obtained as follows:

$$m(x,y) = \begin{cases} 128 & \text{if } \rho_B(x,y) < \text{MGT} \\ 255 & \text{if } \rho_B(x,y) \geq \text{MGT and } \Lambda_B(x,y) \geq 0 \\ 0 & \text{if } \rho_B(x,y) \geq \text{MGT and } \Lambda_B(x,y) < 0 \end{cases} \tag{6}$$

where MGT is the morphological gradient threshold and (x,y) is a pixel in X. The resulting image has three different gray-levels, according to if a pixel belongs to an object, to the background or to the borders.

The choice of the threshold value is one of the most difficult tasks, since the final result is high dependent on many factors, such as lighting conditions, objects texture or shading. Fig. 7 shows the results of the construction of the pixel-symbol map for the image in Fig. 6, with several different MGT values.



(a) (b) (c)

Fig. 7. The pixel-symbol map $m(x,y)$ with different MGT values: (a) Gradient mean. (b) MGT = 0.9 * max($\rho_B(X)$). (c) MGT = 0.8 * max($\rho_B(X)$).

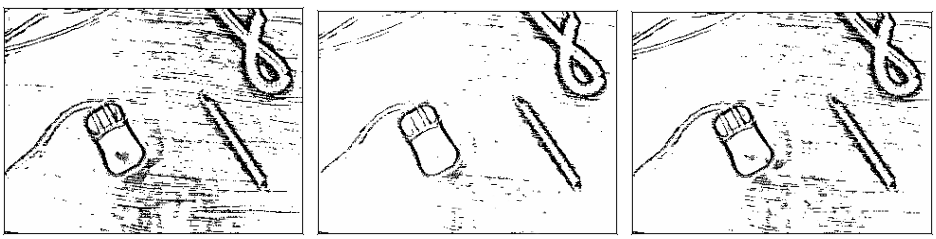
Though many practical systems utilize an experimentally obtained threshold, in this work we consider the use of an automated thresholding system. This method takes into account a binary image metrics to compare the segmentation results and, afterwards, to establish the quality level of the obtained segmentation, as it is described in the following section.

4.2 A Measure of the Quality of the Segmentation

A main problem in computer vision is to be able to compare the results using a proper metrics. This will quantify the differences between two images and, if binary images are used, the method would be both easily implementable and low computationally complex. In

our system we are interested in measuring the distance between image G (the map after gradient thresholding) and image A (the ideal segmentation). Thus, it will establish the optimal MGT value.

Hence, the map must be binarized first. To do this, we must recall that $m(x,y)$ has only 3 gray-levels (Eq. (6)): 0, 128 and 255. For simplicity, let us consider that the threshold is the same as in the construction of $m(x,y)$, i.e., the gradient threshold MGT. The results of this process are shown in Fig. 8.



(a) (b) (c)

Fig. 8. Binarization with different MGT values: (a) Gradient mean. (b) MGT = 0.9 * $\max(\rho_B(X))$. (c) MGT = 0.8 * $\max(\rho_B(X))$.

Next, a reliable measure to compare the obtained image segmentation with an ideal segmentation must be selected.

As proved in (Pujol et al., 2000), a good error measurement for binary images is $\Delta^p(A,G)$, defined as the pth order mean difference between the thresholded distance transforms of two images: A (the ideal segmentation) and G (the binary pixel-symbol map). Let us define first some previous terms:

- Let X denote the pixel raster.
- A binary image $A \subseteq X$ is a set $A = \{x \in X : A(x)=1\}$.

If $\sigma(x,y)$ is the distance between two pixels x and y, the shortest distance between a pixel $x \in X$ and $A \subseteq X$ is defined as:

$$d(x,A) = \inf \{ \sigma(x, a) : a \in A\} \tag{7}$$

Then, for $1 \leq \sigma \leq \infty$, we define:

$$\Delta^p(A,G) = \left[\frac{1}{N} \sum_{x \in X} \left| w(d(x,A)) - w(d(x,G)) \right|^p \right]^{1/p} \tag{8}$$

where N is the total number of pixels in X and $w(t) = \min(t, c)$, for $c>0$.

Intuitively, $\Delta^p(A,G)$ measures the suitability of an estimated image to be used instead of the real one.

Now, we can evaluate the goodness of our segmentation scheme.

4.3 Experiments

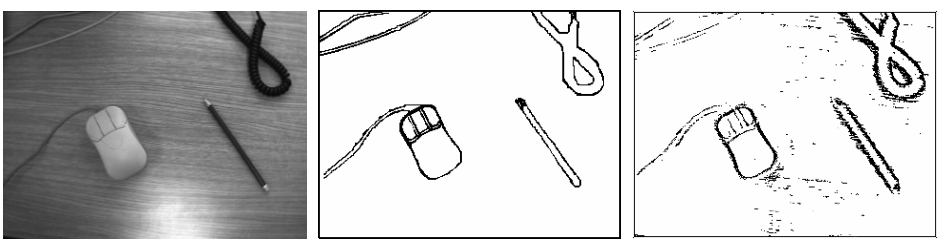
Let us show now the results of some experiments completed for our model. The tests have been performed with a set of real images, whose pixel-symbol maps have been calculated

for different MGT values. Then, after applying the binarization process, the distance $\Delta^p(A,G)$ has been computed.
Table 5 shows the results for the image in Fig. 8, where $p = 2$, $c = 5$.

MGT value	Distance $\Delta^p(A,G)$
MGT = 0.95 * max(ρ_B(X))	0.2722
MGT = 0.9 * max(ρ_B(X))	0.1988
MGT = 0.85 * max(ρ_B(X))	0.3412
MGT = 0.8 * max(ρ_B(X))	0.3704
MGT = 0.75 * max(ρ_B(X))	0.4966

Table 5. Results obtained after the segmentation process.

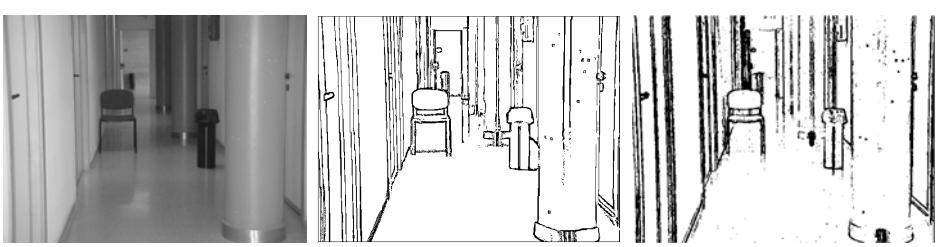
As shown, the lowest distance is obtained when MGT = 0.9 * max($\rho_B(X)$). Fig. 9 compares the ideal segmentation and the MGT segmentation with the lowest $\Delta^p(A,G)$ distance. Intuitively, if we compare the previous results in Fig. 8, the selected MGT value is quite similar to the ideal segmentation.



(a) (b) (c)

Fig. 9. (a) Original image. (b) Ideal segmentation. (c) MGT segmentation.

Let us consider now a more complex real image in order to confirm the accuracy of our technique to give an automated extraction of the threshold value with the best behavior. Fig. 10 and Table 6 show the results.



(a) (b) (c)

Fig. 10. (a) Original image. (b) Ideal segmentation. (c) MGT segmentation.

MGT value	Distance $\Delta^p(A,G)$
MGT = 0.95 * max(ρ_B(X))	0.5526
MGT = 0.9 * max(ρ_B(X))	0.3115
MGT = 0.85 * max(ρ_B(X))	0.2245
MGT = 0.8 * max(ρ_B(X))	0.2731
MGT = 0.75 * max(ρ_B(X))	0.3219

Table 6. Results obtained after the segmentation process.

The minimum distance is obtained again when MGT = 0.9 * max($\rho_B(X)$) and, as a consequence, we can conclude that the parameters used for this segmentation are near optimal, as they have a behavior very close to ideal segmentation.
Nevertheless, the threshold could be adaptively updated so as to assume the real conditions in which every image has been taken by the vision system.

5. Conclusion

In general terms, the path planning process for suitable morphological gradient threshold. To do this, global morphological operators have been used to compute the gradient and the Laplacian and, after a proper binarization, the distance between the ideal segmentation and the MGT segmentation has been computed. As a consequence, the gradient threshold with the lowest distance has been selected as the optimal threshold value. Experimental results show that our model is fast and robust and could be applied for real-time imaging.
As a future work, to fully appreciate the implications of incorporating a path planner into a robot system it is necessary to consider a real robot system. The use of simulations can give a good idea of the ability to solve the basic problem but it is also necessary to consider how the planner will receive input data and how the output path will be used to generate a trajectory and be implemented by a physical robot. This will make possible a more accurate designing method so that the robot internal hardware and software could be efficiently implemented.
Finally, the results of our research could be extended to object classification and recognition. It would be also an interesting task to consider new simulation experiments with different environments, such as image sequences obtained from a camera placed in a robot platform, where real-time constraints have a great influence a mobile robot is strongly influenced by the precision of the acquisition process. Thus, it can be modified both by the quality of the information obtained from the environment, and the attributes of the system and the environment in which it works.
In this chapter, we have developed a proposal of a model for the generation of a map in unknown environments. To do this, we have described a path planning technique for autonomous robots that uses morphological filtering. In this method, some high security paths for a robot to follow are computed; the experimentation shows that the prototype is robust and can be applied in real time for many robotic applications, since it is a very quick algorithm to compute free paths with high probability of no collision.

On the other hand, image segmentation is an essential issue since it is the first step for image understanding, and any other step, such as feature extraction and recognition, heavily depends on its results. In this chapter, we have also described a novel approach to image segmentation based on the selection of a in the final recognition results.

6. References

Basu, M. & Su, M. (2001), Image smoothing with exponential functions, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 15, No. 4, (June 2001), page numbers 735-752, ISSN 0218-0014.

Canny, J. F. (1988), *The Complexity of Robot Motion Planning*, MIT Press, ISBN 0-262-03136-1, Cambridge, MA.

Hopcroft, J. E. & Wilfong, G. T. (1986), Reducing Multiple Object Motion Planning to Graph Searching, *SIAM Journal on Computing*, Vol. 15, No. 3, (February 1986), page numbers 768-785, ISSN 0097-5397.

Joseph, D. A. & Plantiga, W. H. (1985), On the Complexity of Reachability and Motion Planning Questions, *Proceedings of the First ACM Symposium on Computational Geometry*, pp. 62-66, ISBN 0-89791-163-6, Baltimore, Maryland, United States, ACM.

Ouadfel, S. & Batouche, M. (2003), MRF-based image segmentation using ant colony system, *Electronic Letters on Computer Vision and Image Analysis*, Vol. 2, No. 2, (July 2003), page numbers 12-24, ISSN 1577-5097.

Pal, N.R. & Pal, S.K. (1993), A review on image segmentation techniques, *Pattern Recognition*, Vol. 26, No. 9, (September 1993), page numbers 1277-1294, ISSN 0031-3203.

Pujol, F., Pujol, M., Llorens, F., Rizo, R. & García, J. M. (2000), Selection of a suitable measurement to obtain a quality segmented image, Proceedings of the 5th Iberoamerican Symposium on Pattern Recognition, pp. 643-654, ISBN 972-97711-1-1, Lisbon, Portugal, September 2000, F. Muge, Moises P. and R. Caldas Pinto (Eds.).

Serra, J. (1982), *Image Analysis and Mathematical Morphology. Vol I*, Academic Press, ISBN: 012637242X.

Serra, J. (1992), *Image Analysis and Mathematical Morphology. Vol II. Theoretical Advances*, Academic Press, ISBN 0-12-637241-1.

8

Manipulative Action Recognition for Human-Robot Interaction

Zhe Li, Sven Wachsmuth, Jannik Fritsch[1] and Gerhard Sagerer
Applied Computer Science, Faculty of Technology, Bielefeld University
Germany

1. Introduction

Recently, human-robot interaction is receiving more and more interest in the robotics as well as in the computer vision research community. From the robotics perspective, robots that cooperate with humans are an interesting application field that is expected to have a high future market potential. A couple of global and also mid-sized companies have come up with quite sophisticated robotic platforms that are designed for human-robot interaction. The ultimate goal is to place some robotic assistant or companion in the regular home environment of people, who would be able to communicate with the robot in a human-like fashion. As a consequence, the "hearing" as well as the "seeing" -- as the most prominent and equally important modalities -- are becoming major research issues.
From the computer vision perspective, robot perception is more than an interesting application field. During the last decades, we can note a shift from solving isolated vision problems to modeling visual processing as an integral connected component in a cognitive system. This change in perspective pays tribute to important aspects of understanding dynamic visual scenes, such as attention, domain and task knowledge, spatio-temporal context as well as a functional view of object categorization.
The visual recognition of human actions is in the center of all these aspects and provides a bridge for a non-verbal as well as verbal communication between a human and the robot, which both are highly ambiguous. It enables the robot's anticipation of human actions leading to a pro-active robot behavior especially in passive, more observational situations. Furthermore, it draws attention to manipulated objects or places, embeds objects in functional as well as task contexts, and focuses on the spatio-temporal dynamics in the scene.
Recently, much work has been done in the area of gesture-based human-robot interaction (HRI) because of humans' intensive use of their hands. These approaches mostly deal with *symbolic, interactional*, or *referential gestures* that have a communicative meaning on their own (Nehaniv, 2005). In terms of Bobick's taxonomy of *movements, activities, and actions* (Bobick, 1998) they can be characterized as movements or, in more structured cases,

[1] J. Fritsch is now with the Honda Research Institute Europe GmbH in Offenbach, Germany.

activities. In this regard, object manipulations[2] are more complex because the hand trajectory needs to be interpreted in relation to the manipulated object. Due to Bobick this kind of context characterizes *actions*.

In this chapter, we aim at the vision-based recognition of simple actions that are defined by a non-deterministic sequence of object manipulations. As a manipulative gesture, this serves an important communicative function in human-robot interaction. First, the manipulation of an object draws the attention of the communication partner on the objects that are relevant for a performed task. Secondly, it serves the goal of a more pro-active behavior of the robot in passive, more observational situations. As Nehaniv states: "If the robot can recognize **what** humans are doing and to a limited extent **why** they are doing it, the robot may act appropriately" (Nehaniv, 2005). For example, in Fukuda's work a cooking support robot is developed (Fukuda et al., 2005). It can recognize human manipulations of objects by sensing the movements of the markers on the objects and give recommendations by speech or gesture. Dropping these kinds of artificial constraints, the recognition problem is becoming notoriously difficult. Assuming that a hand is manipulating a spatially near object, it becomes hard to decide if the object is just passed by the hand or manipulated. Besides this segmentation ambiguity, there is a large spatio-temporal variability of how hand trajectories reach different object types and the appearance of a hand trajectory in a 2D image will also heavily vary according to the position of the object and the view-angle. Finally, the mutual occlusion between the hand and the object causes even more difficulties for object detection and tracking.

In the present approach we will focus on two problems in the recognition of manipulative actions: (i) the segmentation ambiguity and (ii) spatio-temporal variability of the hand trajectory. We propose a unified graphical model with a two-layered recognition structure. On the lower layer, the object-specific manipulative primitives are represented as Hidden Markov Models (HMM) which are coupled with task-specific Markovian models on the upper level. A top-down processing mechanism predicts which kinds of objects are relevant according to the currently recognized tasks. Thereby, a dynamic attention mechanism is realized that reduces the number of considered objects and simplifies the segmentation task of the hand trajectory. Furthermore, the manipulative primitives are spotted by a particle filter (PF) realized HMM matching process. Due to an explicit modeling of an action abortion and resampling step, this method is more promising than traditional HMM forward-backward (Rabiner, 1990) processing and also could achieve more flexible transitions between model states than condensation-based trajectory recognition (Black & Jepson, 1998). Afterwards, the results are fed back into the task level in order to predict the following primitives closing the bottom-up and top-down cycle.

[2] Nehaniv refers to them as *manipulative gestures* (Nehaniv 2005).

Fig. 1. The Bielefeld Robot Companion (BIRON)

In the following part of this chapter, we will firstly review some related work in the field of human action and activity recognition. Then, we will present our system architecture which takes the temporal as well as the spatial context into account. The recognition of human actions is realized in a tightly coupled loop of bottom-up and top-down processing. We start by describing the low-level image processing of the bottom-up part. Then, we discuss how the object-specific manipulative primitives are spotted under spatio-temporal variability. The modeling of the manipulative task lies on top. The other half of the loop combines the top-down task knowledge with the bottom-up processing scheme. The experiment section presents the results on a corpus of 8 persons performing 3 different tasks consisting of different sequences of primitive actions. Finally, the conclusion will give some discussion on the approach and the possible future work.

2. Recognition of Human Movements, Activities, and Actions

A robot that is autonomously moving and acting in a human environment needs to understand and predict human behavior to a certain degree. While small automatic vacuum cleaners will mainly deal with collision avoidance for safety issues, larger movable robots, like the Bielefeld Robot Companion (Haasch et al., 2004) in Fig. 1 which is based on a Pioneer peopleBot platform, need to respect human activities and situations beyond physical predictability leading to the recognition of human intentions. This starts by considering social spaces, detecting when a person does not want to be disturbed, and ends in solving cooperative tasks with a human partner. The same accounts for human-robot communication starting with the problem to detect if and when a person communicates with the robot (Lang et al. 2003), via the interpretation of a communicative gesture (Pavlovic et al. 1997) to the interpretation of the action context of an unspecific verbal statement (Wachsmuth & Sagerer, 2002; Ballard & Yu, 2003). The reason for the increasing complexity in the interpretation of human motion patterns is the underlying factor of human intentions. The meaning of very similar human motions heavily depends on different levels of human intention. In this regard, Fleischman and Roy (2005) argue that learning the meaning of verbs is much harder that learning nouns. They distinguish between two different kinds of

ambiguities. (1) The vertical ambiguity refers to a possible causal chain of intentions, e.g. in order to *get a cup*, I need to *find a cup, open the cupboard,* and *grab the handle*. Thus, the same action *'the hand moves to the handle of the cupboard'* could be named on different levels of intention. (2) The horizontal ambiguity resambles that the high level interpretation could be ambiguous. For example, the same action as before could be interpreted as *clean the cupboard* instead of *get a cup.*

The different levels of intention have a different scope of interpretation in time and space. The physical prediction can be managed on a subsymbolic level considering the current trajectory of the human movement. Modeling social spaces needs at least some kind of representation of the human's mental state, while the recognition of actions like the opening of a cupboard needs to consider the relation of a human pose with regard to environmental objects and the changes of the object states over time.

The concept of different interpretation scopes directly fits Bobick's categorization of motion recognition: movement, activity, and action (Bobick, 1998). While movements can be characterized by reoccurring trajectories with a dedicated symbolic meaning, the interpretation of activities needs the extension of the scope in time in order to infer a higher level of intention. It represents larger-scale events, which typically include interactions with the environment and causal relationships. Actions involve a state change of the environment extending the scope into space.

So far we did not focus on the kind of body movement performed by a human. A large amount of work is dedicated to whole body movements. An overview of several approaches is given by Gavrila (1999). Spatial as well as temporal contexts are considered by Intille & Bobick (2001) in terms of multiperson actions and Fleischman, Decamp, & Roy (2006) in terms of places in an living environment. However, these approaches are mainly based on top-down views from surveillance cameras. In the robotics field most work is dedicated to *gestures,* i.e. intentional hand/arm movements that are mainly used for human-computer or human-robot interaction. A taxonomy of these is given by Pavlovic, Sharma, & Huang (1997). They distiguish between manipulative and communicative gestures, on the one hand, and unintentional movements, on the other hand. Manipulative gestures are used to act on objects in the environment and, thereby, constitute actions, while communicative gestures are mainly characterized by a temporally structured activity. In the following, we will focus on manipulative gestures.

The recognition of manipulative gestures is one of the most complex tasks as the system needs to extract relational features between the human motion and the environmental objects in cases of a high degree of occlusion. Therefore, most related work on manipulative action recognition simplifies the setting to a certain degree. A common scenario that is well motivated from domestic environments assumes that all relevant actions are performed on a table top (e.g. preparing a meal, decorating a table, performing typical office work, watering flowers). Thus, we assume that a mobile robot moves to a place around the table where it is able to observe the sequence of actions in focus.

In order to recognize these, more sophisticated schemes are needed that explicitly model contextual factors defining actions. Jo used a Finite State Machine (FSM) for modeling possible state transitions in the manipulative gesture recognition (Jo et al., 1998). Bobick developed a PNF (past-now-future) constraint network to model the temporal structure of actions and subactions (Pinhanez & Bobick, 1998). These typically are pure semantic approaches, which have not used explicit motion models. In Chan's work, a simple feature

vector is used for modeling the interaction primitive, e.g. *approach*. The transition of the semantic primitives are modeled by HMMs (Chan et al., 2004). Because of the early symbolic abstraction of trajectory information, this method can only be applied in a restricted scenario. An approach that actually combines both types of information, sensory trajectory data and symbolic object data, in a structured framework is Moore's concept of objectspaces (Moore et al., 1999). Here a camera mounted on the ceiling observes a human interacting with different objects. Certain image processing steps are carried out to obtain image-based, object-based, and action-based evidences for objects and actions, which are integrated using Bayesian networks. Action primitives are recognized from hand trajectories using HMMs that are trained offline on different activities related to the known objects. Our approach uses a similar object represention scheme but goes beyond this work because the spotting of meaningful parts in longer hand trajectories is seriously considered and a combined top-down and bottom-up mechanism solves the object attention problem. Furthermore, the proposed model enables the system to infer high-level intentions in the manipulative gesture detected.

While these approaches center a context area around detected objects, hand-centered methods define context areas relative to a hand trajectory. Fritsch et al. (2004) put forward such an approach. In this case, the trajectory information is augmented in each time step by contextual objects that are searched on-line using the context area bound to the moving hand. A hierarchical structure is used to model the manipulative sequence by Li et al. (2005). In both works, the segmentation and spatio-temporal variability problems are coped with a particle filter. But the hand trajectory template, which is used as the primitive, lacks the capability of generalization. For representing all possible hand trajectories in manipulation, a huge number of templates are needed.

Another specific application is presented by Yu & Ballard (2002). They argue that the eyes guide the hand in almost every action or object manipulation. In their work, the eye motion is measured by a head-mounted eye tracker and used for the segmentation of hand trajectories and the detection of objects. HMMs are used for action recognition which is purely based on trajectory information. Then object and action information is integrated on a symbolic level using action scripts.

3. System Architecture

In contrast to purely trajectory-based techniques, the presented approach is called object-oriented w.r.t. two different aspects: it is object-centered in terms of trajectory features that are defined relative to an object, and it uses object-specific models for action primitives. In our definition, the manipulative action has two semantic layers. The bottom layer consists of the object-specific manipulative primitives. Each object has its own set of manipulative primitives because we argue that different object types serve different manipulative functions and even manipulations with the same functional meaning are performed differently on different objects. The top layer is used for representing the manipulative task, which are modeled by typical transitions between certain manipulative primitives. The system architecture is shown in Figure 2. The architecture realizes a combined bottom-up top-down processing loop that utilizes the task-level prediction of possible primitives in order to restrict the object types possibly detected as well as the action primitives possibly recognized. In the bottom-up path, according to the top-down prediction a processing thread is created for each detected object that consists of a trajectory segmentation, a feature

computation, and an HMM-based recognition step. Thus, all three steps are performed differently for each object in parallel and the hand trajectory information is passed to each object-centered processing thread. The parallel processing for the objects avoids the ambiguity of the trajectory context if there are many objects in the scene. In the following sections, we will show how the object-specific manipulative primitives are detected in each thread, are combined for task recognition and effect the top-down process.

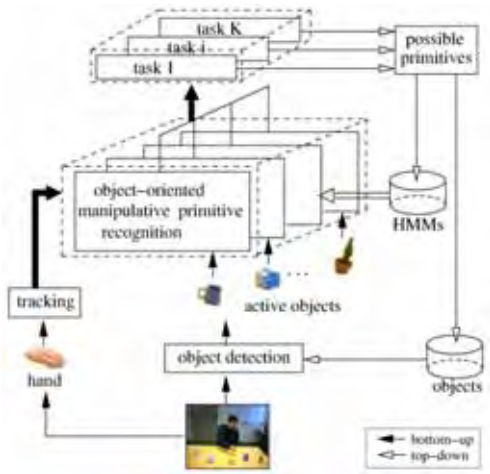


Fig. 2. The system architecture and the processing flow

4. Feature Extraction

The manipulative gesture is different to the face-to-face interactional gesture because the former reflects the interaction between the human's hand and the objects while the latter is typically characterized by a meaningful trajectory of the pure hand movement, e.g. the American Sign Language (Starner & Pentland 1995). Therefore, besides tracking the performing hand over time, the objects in the scene are also detected. For modeling typical object manipulations like "take" or "pour", the selected features describe the relative movements between the hand and the objects in 2D images. The reason why we are not using 3D representation is two fold. On the one hand, the 3D tracking of a person would need an elaborated body model and its tracking in mono-camera images is still a field of active research (Schmidt et al., 2006). Better tracking results can be achieved by using stereo cameras, which poses further constraints on the hardware setting. On the other hand, we argue that the perspective of a robot with regard to manipulative actions performed on a table top (as described in Section 2) can be assumed to be roughly stable, if the robot is able to chose an appropriate position relative to the human actor. In the following, this section will explain the computation for locating the hand and objects in the images and the construction of the interaction feature vector.

4.1 Hand Detection and Tracking

The hand is detected in a color image sequence by an adaptive skin-color segmentation algorithm (Fritsch, 2003) and tracked over time using Kalman filtering. Figure 3 shows the

screen shot of the processing from left to right: the raw image, the thresholded image indicating the skin-color pixels, and the region tracking. Currently only single hand manipulations are assumed. So the bigger skin-color region is labeled as face. The smaller is the hand. The hand observation O_t^{hand} is represented by the hand position $(h_x, h_y)_t$ at time t.

Fig. 3. The screen shots of hand tracking

4.2 Pre-knowledge and Detection of Object

Because the features of the manipulative gesture are based on the relative movements, a reliable detection of objects is crucial for the overall system performance. In order to avoid occlusion problems with interacting hands, we assume that a standard object recognizer, like those using Scale Invariant Feature Transform (SIFT) (Lowe, 2003), is applied on the static scene. Then, object-dependent primitive actions are always defined with regard to the object that is approached by the hand trajectory. If a moved object is applied to another object, the second object defines the object context. As we can have several static objects in the scene, the overall object observation vector contains multiple objects:

$$\mathrm{O}^{obj} = \{\mathrm{O}_1^{obj}, \cdots, \mathrm{O}_i^{obj}, \cdots, \mathrm{O}_L^{obj}\} \tag{1}$$

with

$$\mathrm{O}_i^{obj} = (o_x, o_y, ID, o_h, o_w) \tag{2}$$

The observation vector of a detected object O_i^{obj} contains its position (o_x, o_y), a unique identifier ID for each different object type in the scene and its height o_h and width o_w.

4.3 Segmentation of Trajectory

It is common sense that the relative movement between hand and object contains less interaction features when they are far away from each other. A vicinity of an object is defined that is centered in the middle of the object detected. It is limited by the ratio β of its radius and the object size, which is shown by a blue circle in Figure 4. Based on this vicinity, a pre-segmentation step of the hand trajectory is performed that ignores irrelevant motions for primitive recognition. Considering the possible occlusions in manipulation and the uncertainty in moving an object, a segment is started when the hand enters the vicinity or when an object is detected and the hand is already in the vicinity (object put down into the

scene). It ends when the hand goes out of the object's vicinity or when the object is lost after the hand moves away (object has been taken). As a consequence, the trajectory is segmented differently based on the different objects in the scene. To handle this multi-observation problem, one processing thread is started for each detected object. In the following, the processing in a single thread will be introduced. There, the final segmentation is directly coupled with the recognition step.

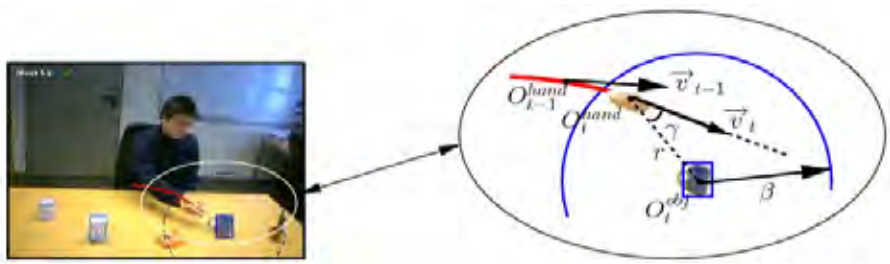


Fig. 4. The interaction feature vector

4.4 Interaction Feature Vector

During a manipulative action, the hand movements in the object vicinity can indicate an intended physical contact with object i, e.g. the hand will move towards the cup and slow down when the person wants to take it. Thus in the processing thread i, the interaction of the hand and the object is represented by a five-dimensional feature vector V_f that is calculated from O^{hand} and O_i^{obj}. It contains the features: magnitude of hand speed v, change of the hand speed Δv, change of speed direction $\Delta\alpha$, distance r between the object and the operative hand, as well as the angle γ of the line connecting object and hand relative to the direction of the hand motion.

$$V_f = (v, \Delta v, \Delta\alpha, r, \gamma) \tag{3}$$

The parameter v, Δv, and r are all scaled by object size. So the features are invariant with regard to translations, scale, and rotations.

5. Manipulative Primitive Detection

Although an object vicinity is defined for cutting away the hand trajectories which are less relevant to object manipulation, it is a coarse segmentation. The relative movements of the hand in an object vicinity can also contain both a typical interaction and some meaningless part. Consequently, the typical hand-object interactions, which we named *object-specific manipulative primitive*, have to be detected in a longer trajectory. The major methods include Dynamic Time Warping (DTW) (Alon et al., 2005), Artificial Neural Networks (ANN) (Kjeldsen & Kender 1995), and Hidden Markov Models (HMM) (Morguet & Lang, 1998; Lee & Kim, 1999). The DTW can to a certain extent cope with spatio-temporal variability. But as a template-based dynamic matching technique, it needs a large number of templates for a range of variations. ANN can achieve good detection results on static patterns,

including fixed length trajectories. It is not suited for the manipulative primitives which can have huge temporal variance. The HMM is another well-known technique for modeling sequential signals. By defining the transition between states and the state dependend observations in a probabilistic way, variations can be coped with to a certain degree. It is effectively used in speech recognition, handwriting recognition and human activities recognition. However, the standard forward algorithm to calculate the probabilities of the HMM candidates given the observation has the assumption that the whole sequence is emitted by one HMM. In order to spot the partition which conforms to an HMM from a long observation, some approaches, e.g. HMM-based threshold model (Lee & Kim, 1999) and Normalized Viterbi algorithm (Morguet & Lang, 1998) were put forward. Because the output score of the continuous observations of a given HMM will permanently increase or decrease, a window is used to tune the weights of the observation. Nonetheless, the fixed length of the window conflicts with the temporal variability of the signal. Recently the Sequential Monte Carlo (SMC) method also named Particle Filter (PF) is getting more and more focus in the pattern recognition society, which allows an on-line approximation of probability distributions using samples (particles). It has been used for template-based trajectory matching (Blake & Jepson, 1998). In order to keep the spatio-temporal variability of HMMs and use the advantage of PF on tracking the models with weighted particles, a PF realized HMM matching method is implemented to detect object-specific manipulative primitives. This process is building the bridge between the low-level image processing and the task knowledge.

5.1 HMM for Manipulative Primitive

The object-oriented manipulative primitives are modeled by ergodic HMMs. Different to the normal parameter set $\lambda = (A, B, \Pi)$ of an HMM, a terminal probability E is added. It reflects the terminal probability of an HMM given a hidden state s_i. So the whole set consists of:

- $\Pi = \{\pi_i | \pi_i = P(q_1 = s_i)\}$, initial probability of state s_i.
- $A = \{a_{ij} | a_{ij} = P(q_{t+1} = s_j | q_t = s_i)\}$, transition probability from state s_i to s_j.
- $B = \{b_i(k) | b_i(k) = P(o_t = v_k | q_t = s_i)\}$, probability of observing v_k given hidden state s_i.
- $E = \{e_i | e_i = P(q_{end} = s_i)\}$, terminal probability of state s_i.

Considering the small amount of training data, we use discrete HMMs. The whole feature space is discretized into 2x 2 x 4 x 3 =48 cells based on the following quantized dimensions:

Parameters	Quantization
v	$< v_{threshold}, \geq v_{threshold}$
Δv	$< 0, \geq 0$
$\Delta\alpha$	$< 90, \geq 90$
r	$[0 \ldots \beta/4 \ldots \beta/2 \ldots 3\beta/4 \ldots \beta]$
γ	$< 90, \geq 90$ if $v \geq v_{threshold}$

Table 1 Vector quantization of the interaction feature space

They define the observation states for the following HMMs. The angle γ between the object -hand connection line and the direction of the hand motion is quantizied conditioned on v because it has much noise when the hand speed is very low. The HMM parameter set is learned from manually segmented trajectories with the Baum-Welch algorithm, e_i is calculated similar to π_i, except using the last states.

5.2 PF-based HMM Matching

In order to detect the primitives from the pre-segmented trajectories, a PF called Sampling Importance Resampling (SIR) is used, better known as Condensation introduced by Isard and Blake (Isard & Black 1996). Figure 2 shows a two time-slice Dynamic Bayesian Network (DBN) which indicates the dependency structure of the probabilistic model. For each one in the L objects, the matching of the M HMMs and the observation are achieved by temporal propagation of a set of weighted particles:

$$\{(S_t^{(1)}, w_t^{(1)}), \cdots (S_t^{(N)}, w_t^{(N)})\} \tag{4}$$

with

$$S_t^{(i)} = \{p_t^{0(i)}, q_t^{(i)}, e_t^{(i)}\} \tag{5}$$

The number of particles is N. The sample $S_t^{(i)}$ contains the primitive index $p_t^{0(i)}$, the hidden state $q_t^{(i)}$, and the terminal state of this primitive $e_t^{(i)}$ at time t (see Figure 5). The resampling step reallocates a certain fraction of the particles with regard to the initial distribution Π. Consequently, the weight $w_t^{(i)}$ of a sample can be calculated from

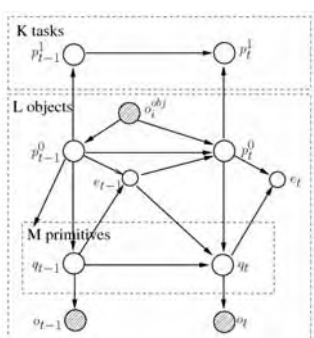


Fig. 5. A Dynamic Bayesian Network represents the dependency structure of two time slices in the recognition model. Each object-centered processing thread corresponds to one of the L plates in the dependency model. K is the number of different tasks modeled in the system and M is the number of possible primitives which each corresponds to one state of the variables p_t^0 and p_t^1, respectively. The upper index of these variables denotes the primitive vs. task level.

$$w_t^{(i)} = \frac{p(o_t | S_t^{(i)})}{\sum_{j=1}^{N} p(o_t | S_t^{(j)})} \tag{6}$$

The $p(o_t | S_t^{(i)})$ in it is the observation probability of o_t given $q_t^{(i)}$ and HMM $p_t^{0(i)}$. The propagation of the weighted samples over time consists of three steps:

Select: Selection of $N-M$ samples $S_{t-1}^{(i)}$ according to their respective weight $w_{t-1}^{(i)}$ and random initialization of M new samples. That means some particles which have high weights will be selected multiple times and some particles which have low weights will not be selected at all.

Predict: The current state of each sample $S_t^{(i)}$ is predicted from the samples of the select step according to the graphical model given in Figure 5. The terminal state $e_{t-1}^{(i)}$ is a bi-valued variable, 0 means the primitive is continuing and 1 means the primitive ends here. So if $e_{t-1}^{(i)}$ is 0, the next hidden state $q_t^{(i)}$ is sampled according to the transition probability of the HMM of primitive $q_{t-1}^{(i)}$ and the primitive index $p_t^{0(i)}$ keeps the same as $p_{t-1}^{0(i)}$. If the terminal state $e_{t-1}^{(i)}$ is 1, the primitive index $p_t^{0(i)}$ will be sampled according to the current possible primitives of this object. Then the hidden state $q_t^{(i)}$ is sampled according to the initial probability of the HMM of the new primitive $p_t^{0(i)}$. At the end of this step, the terminal state of this particle $e_t^{(i)}$ is sampled based on the terminal probability of the current primitive state $q_t^{(i)}$.

Update: Determination of the weights $w_t^{(i)}$ of the predicted samples $S_t^{(i)}$ using Eq. 6.

The recognition of a manipulative primitive is achieved by calculating the **end-probability** P_{end} that a certain HMM model p_i is completed at time t:

$$P_{end,t}(p_i) = \sum_n w_t^{(n)} \text{, if } p_i \in S_t^{(n)} \tag{7}$$

A primitive model is considered recognized if the probability $P_{end,t}(p_k)$ of the primitive model p_k exceeds a threshold p_{th}^0 which has been determined empirically.

The resampling step in the particle propagation is able to adapt the starting point of the model matching process if the beginning of the primitive does not match the beginning of the segment. The end-probability gives an estimation of the primitive's ending point. This combination to a certain extent solves the problem of the forward-backward algorithm which needs a clear segmentation of the pattern.

6. Task Level Processing

6.1 Model of Tasks

The manipulative tasks are modeled as the first-level Markovian process which is the same as Moore's definition (Moore et al., 1999). Although this assumption violates certain domain dependencies, it is an efficient and practical way to deal with task knowledge. In the model Λ_i for a manipulative task i, a set of possible manipulative primitives P_i^1 are defined, e.g., in the "prepare tea" task, the primitives "take cup", "take tea can" could appear but not "take milk". Because of the high effort needed for recording a huge mount of task sequences, the number of training examples for each complete task is too low for robustly estimating transition probabilities. Therefore, we model a task by a set of possible primitive pair transitions similar to a word pair grammar in automatic speech recognition. The set of transition rules A_i^1, the possible start symbols Π_i^1, and the set of possible end symbols E_i^1 is learned from the output of the primitive recognition layer on a training set by thresholding the frequency of pairs observed in sequences of action primitives (see Section 7.2 for more details). Suppose the result from the manipulative primitive recognition is the sequence $p_1^1,\ldots,p_t^1$. To calculate the acceptance of a task $\Lambda_i = (P_i^1, \Pi_i^1, A_i^1, E_i^1)$, only the primitives which are in the primitive list of the task Λ_i will be chosen because of the possible insertion in the primitive recognition.

$$(p_1^1,\ldots p_t^1 \mid p_j^1 \in P_i^1, j=1\ldots t) \in \{\mathrm{P} \mid p_1^1 \rightarrow^*_{A_i^1} p_t^1, p_1^1 \in \Pi_i^1, p_t^1 \in E_i^1\} \tag{8}$$

where P denotes the possible sequences from primitive p_1^1 to p_t^1 while considering transitions in A_i^1. Eq. 8 can easily be evaluated according to the parameter set Λ_i.

6.2 Top-down Process

Because of the object-specific primitive definition and its parallel processing for each affected object, the system confronts an attention problem when there are many objects appearing in the scene, simultaneously. In order to solve this problem, a top-down process is introduced, in which the possible subsequent primitives are predicted on the ground of the active task models and the previous results from the manipulative primitive recognition. This prediction is similar to the computation of a look ahead symbol in parsing strategies. For the prediction step, different parsing alternatives are considered during the HMM matching process. For all primitives that gain an end probability $P_{end,t}(p_i) > 0$ a lookahead symbol is generated. If a primitive has been recognized this primitive is eliminated as a lookahead symbol. Because the predicted action primitives are specific for certain object types, the set of the next possibly manipulated object types can be calculated and be passed to the object detection component. This realizes a task driven attentional cue for early processing steps of the system (Figure 2). Additionally, the expectations from the predicted action primitives are used to restrict the HMMs applied in the PF based matching process.

7. Experiments and Results

In order to evaluate the quality of the manipulative gesture recognition, a scenario in an office environment has been designed as shown in Figure 6. A person is sitting behind a table and manipulates the objects that are located on it. She or he is assumed to perform one of three different manipulation tasks:
(1) *water plant*: take cup, water plant, put cup;
(2) *prepare tea*: consists of take/put cup, take tea can, pour tea into cup, put tea can;
(3) *prepare coffee*: consists of take/put cup, take milk/sugar, pour milk/take sugar into cup, put milk.
In the experiment, each task is performed 4-5 times by 8 different persons resulting in 36 sequences for each task and a total of 108 sequences. The images are recorded with a resolution of 320x240 pixels and with a frame-rate of 15 images per second. The object recognition results have been labeled because the evaluation experiment should concentrate on the performance of the action and task recognition. The object in the hand is ignored so that *pour milk into cup* and *pour tea into cup* are the same primitive actions. The scenario is restricted in so far that we assume a static camera, a known configuration of objects, and a camera view that is roughly orthogonal to the relevant movements.

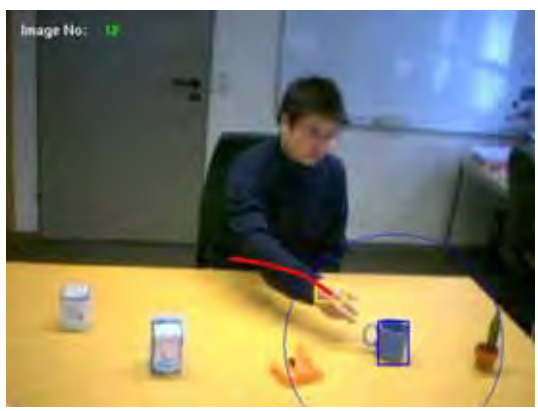


Fig. 6. The office scenario used in the experiment.

7.1 Manipulative Primitive Recognition

The first evaluation is used to test the performance of the object-oriented manipulative primitive recognition. There are five different objects used in the experiment: tea can, milk, sugar, cup and plant. Figure 7 shows the primitives defined for each object type. The evaluation is done for all segments computed by the pre-segmentation step (see Section 4.3). These segments either contain a real manipulative primitive action which we call positive segments (PS) or contain just a hand passing by an object which we call negative segments (NS). For the positive segments, we calculate the false negative (FN) rate. For negative segments, the false positive (FP) rate is calculated. In order to achieve a good system performance both rates should be low because both kinds of errors would seriously affect human-robot interaction. We randomly divided the 108 whole task sequences into a training set of 60, and a test set of 48 sequences. Because of the low number of training examples, we

run the Baum-Welch algorithm used for the HMM learning procedure 10 times with random initialization and give a standard deviation for the FN and FP rates. The results are computed using the parameter setting: $N = 500$, $M = 50$, $p_{th}^{0} = 0.2$, and $\beta = 3$. From the results shown in Figure 7, it could be found that the "put" primitives are recognized with lower FN rate than the "take" and "pour" primitives because the variations of the latter two are much higher from person to person. Figure 8 shows the end probabilities of different manipulative primitives in a prepare tea task. The horizontal line above zero is the recognition threshold and the temporal periods which are coloured indicate that the hand is in the object vicinity at that moment.

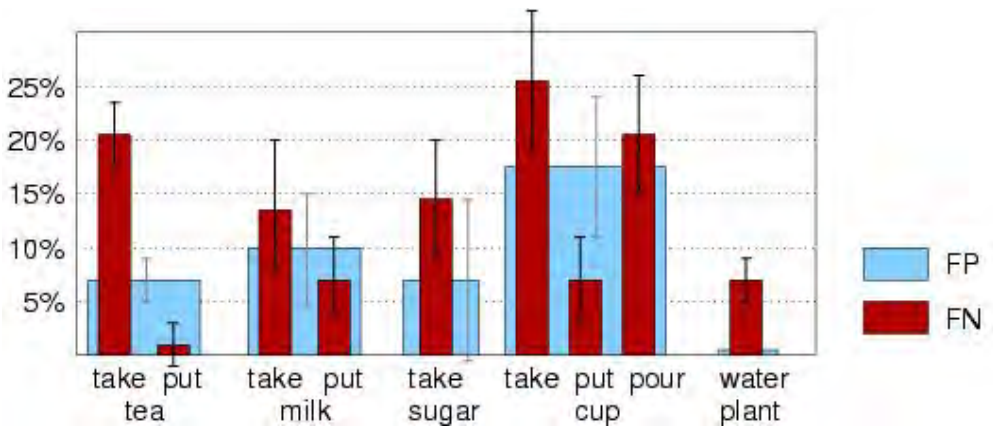


Fig. 7. The recognition results of the object-specific manipulative primitives in both positive and negative segments.

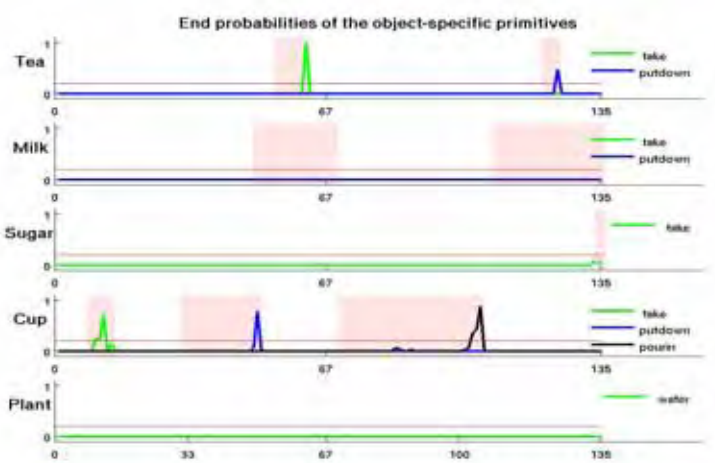


Fig. 8. The end probabilities of the object-specific manipulative primitives in a *prepare tea* task

7.2 Manipulative Task Recognition

The second evaluation assesses the overall system performance. A manipulative task consists of the manipulative primitive sequence. However the ordering of the sequence is neither pre-determined nor completely fixed. For example some people may take sugar before taking milk, some will do it the other way around. But there probably will be an ordering between taking the cup and the watering action which needs to be learned from the data. For learning the possible transition pairs of each task model, the data set is divided into the set of 20 observation sequences, that was already used for learning the primitive action models, and a set of 16 sequences that are used for a one-leave-out experiment. Thus, each task model is learned from 35 task sequences in each experiment. The possible word pair transitions are extracted from the training data by a frequency threshold. The task

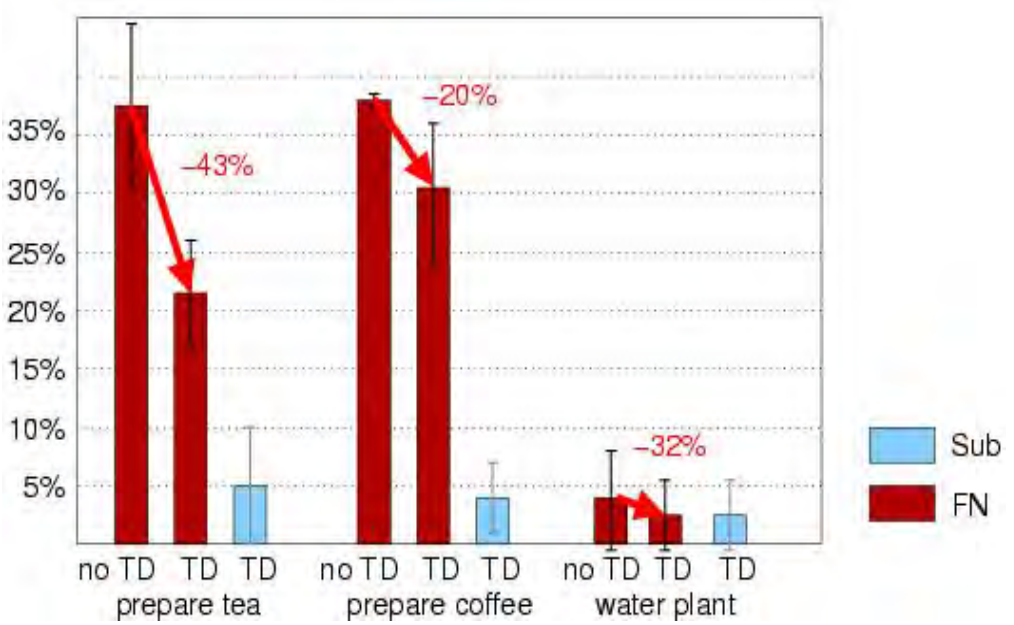


Fig. 9. The recognition results of the manipulative tasks with and with out top-down processing.

recognition results of the whole system are compared with (TD) and without (no TD) the top-down attention processing (see Figure 9). The FN rate clearly shows a significant drop in case of top down processing for *prepare tea* and *prepare coffee*. Because sometimes an expected primitive was misrecognized in a way that was not covered by the task grammar, the rejection of these tasks caused relatively high FN rates but nearly no substitution errors (Sub.). The processing time for a 180-frame ``prepare coffee'' sequence with the former method is 54s running on MATLAB, which is much lower than the 86s needed by the pure bottom-up processing.

8. Conclusion

The recognition of manipulative actions and tasks is an essential component for the natural, pro-active, and non-intrusive interaction between humans and robots. However, most techniques for the recognition of symbolic, interactional or referential gestures cannot be transferred because they ignore the object context and assume an object independent characteristic of the hand trajectory. Other approaches that focus on action recognition either use a pure semantic approach without considering motion models or simplify the trajectory segmentation problem in a pure bottom-up process.

The presented approach overcomes several of these deficiencies. The contextual objects are used for a pre-segmentation of the hand trajectory; the manipulative action primitives are spotted by a particle filter approach that matches object specific HMMs in a more flexible way than the traditional forward-backward algorithm; tasks are defined by a set of possible transition rules similar to a word pair grammar that is automatically extracted from a small test set. By calculating a set of lookahead symbols on the task level, a task-driven attention filter is realized that tightly couples bottom-up and top-down processing. We were able to show first experiments that underline the potential of the presented approach. The action primitives were recognized quite robustly. The top-down attention filter significantly improves the computation time as well as the recognition performance.

Further work needs to concentrate on several issues. In terms of feature description neither pure symbolic nor trajectory-based characterizations will be general enough to describe the huge variety of manipulative actions. Trajectory-based features allow to distinguish actions that do not result in observable state changes of the objects, but suffer from large trajectory variations. The proposed object specific motion-models account to these variations to a certain degree. How to deal with multiple representations on both symbolic and sub-symbolic levels is still an open research question. The coupling of motion models and object types also leads to another important aspect of actions: the concept of object affordances. The observed shape and function of an object activates an expected set of hand trajectories and vice versa. We expect that this kind of coupling will be a key issue both in categorization of objects and learning new action verbs. Another aspect is the development of more sophisticated task models that need to include human intentions on multiple scopes of time and space. Finally, more sophisticated experiments are needed to evaluate current action recognition approaches. Appropriate benchmark datasets for manipulative action recognition are currently not available and most approaches focus on their specific application domain.

9. References

Alon, J.; Athitsos, V. & Sclaroff, S. (2005) Accurate and efficient gesture spotting via pruning and subgesture reasoning. In *Proc. ICCV Human-Computer Interaction Workshop*, pages 189–198.

Black, M. J. & Jepson, A. D. (1998). A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *European Conf. On Computer Vision, ECCV-98*, pages 909–924, Freiburg, Germany.

Ballard, D. H. & Yu, C. (2003). A multimodal learning interface for word acquisition. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing.* volume 5, pages 784–790.

Bobick, A. F. (1998). Movement, activity, and action: The role of knowledge in the perception of motion. In *Royal Society Workshop on Knowledge-based Vision in Man and Machine*.

Chan, M.T.; Hoogs, A.; Schmiederer, J. & Petersen, M. (2004). Detecting rare events in video using semantic primitives with hmm. In *ICPR04*, pages IV: 150–154.

Fleischman, M.; Decamp, P. & Roy, D. (2006). Mining temporal patterns of movement for video content classification. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 183–192, New York, NY, USA. ACM Press.

Fleischman, M. & Roy, D. (2005). Why verbs are harder to learn than nouns: Initial insights from a computational model of intention recognition in situated word learning. In *Proc. of the 27th Annual Meeting of the Cognitive Science Society*.

Fritsch, J. (2003). *Vision-based Recognition of Gestures with Context*. Dissertation, Bielefeld University, Technical Faculty.

Fritsch, J.; Hofemann, N. & Sagerer, G. (2004). Combining Sensory and Symbolic Data for Manipulative Gesture Recognition. In *Proc. IEEE ICPR*, pages 930–933, Cambridge, UK.

Fukuda, T.; Nakauchi, Y.; Noguchi, K. & Matsubara, T. (2005). Time series action support by mobile robot in intelligent environment. In *Proc. IEEE Int'l Conf. Robotics and Automation*, pages 2908–2913, Barcelona, Spain.

Gavrila, D. M. (1999). The visual analysis of human movement: a survey. In *Comput. Vis. Image Underst.*, 73(1):82–98.

Haasch, A.; Hohenner, S.; Huwel, S.; Kleinehagenbrock, M.; Lang, S.; Toptsis, I.; Fink, G. A.; Fritsch, J.; Wrede, B.; & Sagerer, G. (2004). Biron — the bielefeld robot companion. In *Proc. Int. Workshop on Advances in Service Robotics*, pages 27–32, Stuttgart, Germany.

Intille, S. S. & Bobick, A. F. (2001). Recognizing planned multiperson action. In *Comput. Vis. Image Underst.*, 81(3):414–445, March 2001.

Isard, M.; & Blake, A. (1998). Condensation – conditional density propagation for visual tracking. In *Int. J. Computer Vision*, pages 5–28.

Jo, K. H.; Kuno, Y. & Shirai, Y. (1998). Manipulative hand gesture recognition using task knowledge for human computer interaction. In *Proc. Int'l Conf. on Automatic Face and Gesture Recognition*, pages 468–473.

Kjeldsen, R. & Kender, J.R. (1995) Visual hand gesture recognition for window system control. In *Proc. Int'l Workshop Automatic Face and Gesture Recognition*, pages 184–188.

Lang, S.; Kleinehagenbrock, M.; Hohenner, S.; Fritsch, J.; Fink, G. A. & Sagerer, G. (2003). Providing the basis for human-robot-interaction: a multi-modal attention system for a mobile robot. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 28–35, New York, NY, USA, ACM Press.

Lee, H. K. & Kim, J. H. (1999). An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973.

Li, Z.; Hofemann, N.; Fritsch, J. & Sagerer, G. (2005). Hierarchical modeling and recognition of manipulative gesture. In *Proc. ICCV, Workshop on Modeling People and Human Interaction*, Beijing, China.

Li, Z.; Fritsch, J.; Wachsmuth, S. & Sagerer, G. (2006). An object-oriented approach using a topdown and bottom-up process for manipulative action recognition. In *Annual Symposium of the German Association for Pattern Recognition (DAGM),* pages 212–221, Berlin, Germany, Springer-Verlag.

Lowe, D. (2003). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 20:91–110.

Moore, D.J.; Essa, I.A. & Hayes III, M.H. (1999). Exploiting human actions and object context for recognition tasks. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 20–27.

Morguet, P. & Lang, M. (1998). Spotting dynamic hand gestures in video image sequences using hidden markov models. In *International Conference on Image Processing*, pages 193–197, Chicago, USA.

Nehaniv, C. P. (2005). Classifying types of gesture and inferring intent. In *Proceedings of the Symposium on Robot Companions: Hard problems and Open Challenges in Robot-Human Interaction AISB'05*, pages 74–81, Hatfield, UK.

Pavlovic, V.; Sharma, R. & Huang, T. S. (1997). Visual interpretation of hand gestures for humancomputer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695.

Pinhanez, C. S. & Bobick, A. F. (1998). Human action detection using pnf propagation of temporal constraints. In *Proc. IEEE CVPR*, pages 898–907, Washington, DC, USA.

Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Schmidt, J.; Kwolek, B. & Fritsch J. (2006). Kernel particle filter for real-time 3D body tracking in monocular color images. In *Proc. of Automatic Face and Gesture Recognition*. Pages 567-572, Southampton, UK.

Starner, T. & Pentland, A. (1995). Real-time american sign language recognition from video using hidden markov models. In *IEEE International Symposium on Computer Vision*, pages 265–270.

Yu, C. & Ballard, D. H. (2002). Learning to Recognize Human Action Sequences. In *2nd International Conference on Development and Learning (ICDL'02)*, pages 28–34.

Wachsmuth, S. & Sagerer, G. (2002) Bayesian networks for speech and image integration. *In Eighteenth national conference on Artificial intelligence*, pages 300–306, Menlo Park, CA, USA.

9

Image Matching based on Curvilinear Regions

J. Pérez-Lorenzo, R. Vázquez-Martín, R. Marfil, A. Bandera & F. Sandoval
Grupo de Ing. de Sist. Integrados, Dept. Tecnología Electrónica, Universidad de Málaga
Spain

1. Introduction

Image matching, or comparing images in order to obtain a measure of their similarity, is a fundamental aspect of many problems in computer vision, including object and scene recognition, content-based image retrieval, stereo correspondence, motion tracking, texture classification and video data mining. It is a complex problem, that remains challenging due to partial occlusions, image deformations, and viewpoint or lighting changes that may occur across different images (Grauman & Darrell, 2005).

Image matching can be defined as "the process of bringing two images geometrically into agreement so that corresponding pixels in the two images correspond to the same physical region of the scene being imaged" (Dai & Lu, 1999). Therefore, according to this definition, image matching problem is accomplished by transforming (e.g., translating, rotating, scaling) one of the images in such a way that the similarity with the other image is maximised in some sense. The 3D nature of real-world scenarios makes this solution complex to achieve, specially because images can be taken from arbitrary viewpoints and in different illumination conditions. Instead, the similarity may be applied to global features derived from the original images. However, this is not the more efficient solution. Besides, these global statistics cannot usually deal with real-world scenarios because they do not often give adequate descriptions of the local structures or discriminating features which are present on the image (Grauman & Darrell, 2005).

Other solution to the image matching problem is to describe the image using a set of *distinguished regions* (Matas et al., 2002). These regions must own some invariant and stable property in order to be detected with high repeatability in images taken from arbitrary viewpoint. Then, the matching between two images is posed as a search in the correspondence space established between the associated sets of distinguished regions. If each region is described by a vector of image pixels, then cross-correlation can be used to obtain a similarity value between two regions (Mikolajczyk & Schmid, 2005). However, due to the high dimensionality of such vector, the generation of the correlation space typically presents a high computational cost. In order to reduce the computational complexity, the number of tentative correspondences can be limited by computing local invariant descriptors for distinguished regions (Matas et al., 2002; Grauman & Darrell, 2005). These descriptors can be also employed to estimate the similarity value between two regions.

In this paper, we have adopted an approach which describes the image using a set of distinguished regions and exploits local invariant descriptors to estimate the similarity value between two distinguished regions belonging to different images. Thus, there are four

main procedures involved in the image matching process: i) detection of distinguished regions, ii) local invariant description of these regions, iii) definition of the correspondence space, and iv) searching of a globally consistent subset of correspondences. This subset of correspondences will permit to associate a similarity score to the images being matched. The main contribution of this work is the introduction of a new set of distinguished regions, the so called *curvilinear regions*.

The choice of the location and shape of the distinguished regions can be considered as a crucial issue in these image matching approaches (Matas et al., 2002). In a typical case, when images are taken from different viewpoints, local image deformations cannot be realistically approximated by translations and rotations, and it is required a full affine model. Then, correspondence cannot be established by comparing regions of a fixed shape like rectangles or circles since their shape is not preserved under affine transformation. Region shape must depend on the image data (Dai & Lu, 1999; Matas et al., 2002). In our case, the proposed method exploits a particular image structure. It is based on the presence, in a typical image, of numerous objects which can be built using cylinders or generalized cylinders (Biederman, 1987). The main disadvantage of the method is to use shapes which must be explicitly present in the image, so it depends on the presence of these specific structures in the scene. On the contrary, curvilinear regions automatically deform with changing viewpoint as to keep on covering identical physical parts of a scene.

This chapter is organised as follows: Section 2 describes related work. The curvilinear region detector is presented in Section 3. Section 4 describes the contour-based descriptor computed for each extracted region. This descriptor is compared to other similar approaches in Section 5.1. The correspondence algorithm is presented in Section 5.2. This Section also describes some experimental results and finally, Section 6 discusses extracted conclusions and future work.

2. Related work

The development of algorithms which use a set of local distinguished items for image matching can be traced back to the works of Moravec (1981) and Harris and Stephens (1988). Although the initial applications of both approaches are for stereo and short-range motion tracking, it can be considered that a similar strategy has been later extended to deal with more difficult problems. Thus, Zhang et al. (1995) propose to match Harris points over a large image range by using a correlation window around each point. The Harris point detector selects any image location that has large gradients in all directions at a predetermined scale. Outliers are then removed by solving for a fundamental matrix describing the geometric constraints between the two views of a rigid scene and removing matches that did not agree with the majority solution.

Local invariant feature matching is extended to general image recognition problems in which a feature is matched against a large set of images by Schmid and Mohr (1997). This approach also employs Harris points as distinguished items, but rather than matching with a correlation window, they use a rotationally invariant descriptor of the local image region. The 2D translation and 2D rotation invariant features are extracted from the intensity pattern in fixed circular regions around Harris points. Invariance under scaling is handled by including circular regions of several sizes. This allows features to be matched under arbitrary orientation change between the two images. Besides, they demonstrate that multiple feature matches could accomplish general recognition under occlusion and clutter

by identifying consistent clusters of matched features. This method has been modified to deal with very large scale changes (Dufournaud et al., 2000) or with colour images (Montesinos et al., 2000).

The Harris point detector is very sensitive to scale changes, so it does not provide a good basis for matching images of different sizes. In any case, representations that are stable under scale change have been proposed. Crowley and Parker (1984) developed a detector that identifies peaks and ridges in scale-space and links these into a tree structure. The tree structure can then be matched between images with arbitrary scale change. The Harris point local feature approach has been modified by Lowe (1999) to achieve scale invariance. Circular regions that maximise the output of a difference-of-Gaussian (doG) filters in scale-space are employed. More recent work on graph-based matching by Shokoufandeh et al. (1999) provides more distinctive feature descriptors using wavelet coefficients. Harris-Laplace regions (Mikolajczyk & Schmid, 2001) are also invariant to rotation and scale changes. These points are detected by the scale-adapted Harris function and selected in scale-space by the Laplacian-of-Gaussian operator. Hessian-Laplace regions (Lowe, 2004) are localised in space at the local maxima of the Hessian determinant and in scale at the local maxima of the Laplacian-of-Gaussian. This detector obtains higher localisation accuracy than the doG approach and the scale detection accuracy is also higher than in the case of the Harris-Laplace detector (Mikolajczyk & Schmid, 2005). The problem of identifying an appropriate and consistent scale for feature detection has been studied in depth by Lindeberg (1993, 1994).

As it is commented above, when images are taken from different viewpoints, image regions are subject to affine transformations. The affine transformation includes rotation, scaling, skewing and translation (Bala & Cetin, 2004). It preserves parallel lines and equispaced points along a line. Therefore, it has been used to approximate the perspective transformation in some cases. Local features have been extended to be invariant to full affine transformations. Harris-affine regions (Mikolajczyk & Schmid, 2004) and Hessian-affine regions (Mikolajczyk et al., 2005) are invariant to affine image transformations. However, they start with initial feature scales and locations selected in a non-affine-invariant manner. Then, the affine neighbourhood is determined by the affine adaptation process based on the second moment matrix. Baumberg (2000) has proposed an invariant descriptor which cannot deal with scale changes. Thus, these regions are invariant under rotation, stretch and skew, but scale changes are dealt with by applying a scale-space approach. The error on the scale also influences the other components of the transformation. Tuytelaars and Van Gool (2004) propose two types of affine-invariant regions, one based on a combination of Harris points and edges and other one based on image intensities. Matas et al. (2002) describe the Maximally Stable Extremal Regions (MSER). They are extracted with a watershed like segmentation algorithm. An important issue that affine invariant approaches must take into account is the sensitivity to noise. Thus, affine features are sensitive to noise, so in practice they have typically lower repeatability than the scale-invariant features (Mikolajczyk, 2002). To deal with this problem, the local descriptor must allow relative feature positions to shift significantly with only small changes in the descriptor. This not only allows the descriptors to be reliably matched across a considerable range of affine distortion, but it also makes the features more robust against changes in 3D viewpoint for non-planar surfaces (Lowe, 2004).

Many other features have been proposed. Some of them make use of region boundaries, which should make them less likely to be disrupted by cluttered backgrounds near object

boundaries. Thus, Matas et al. (2002) have shown that their MSERs can produce large numbers of matching features with good stability. Mikolajczyk et al. (2003) uses local edges while ignoring unrelated nearby edges, providing the ability to find stable features even near the boundaries of narrow shapes superimposed on background clutter. Nelson and Selinger (1998) employ local features based on groupings of image boundaries. Finally, Pope and Lowe (2000) use features based on the hierarchical grouping of image boundaries. A curvilinear-based region detector has been proposed by Deng et al. (2006). It starts by detecting curvilinear structures followed by watershed segmentation to define regions. On the other hand, phase-based local features have been described by Carneiro and Jepson (2002). These features represent the phase rather than the magnitude of local spatial frequencies, which is likely to provide improved invariance to illumination. Schiele and Crowley (2000) have proposed the use of multidimensional histograms. These histograms represent the distribution of measurements within image regions and they may be particularly useful for matching textured regions with deformable shapes. Other useful properties to incorporate include colour, motion, figure-ground discrimination, region shape descriptors, and stereo depth cues.

3. Curvilinear regions

3.1 Definition

Basically, in a digital image, a curvilinear region is a set of pixels delimited by left and right boundaries, $r_l(l)$ and $r_r(l)$. This region can be defined by the parameter vector, $\{a_i, w_i\}_{i=0...L}$, where L is the length of the region, a_i a vector defining the axis between the boundaries and w_i the width of the curvilinear region (see Fig. 1). In a curvilinear region, the ratio between its average width and its total length should be less than a predefined threshold. Besides, left and right borders should be locally parallel, it should exist a geometric similarity around the region axis and the colour along this axis should be homogeneous. These items will be extended in next epigraphs.

I. Symmetry around the axis

If we define $\Delta w(l)$ as the difference of width at both sides of the medial axis:

$$\Delta w(l) = | w_l(l) - w_r(l) | \tag{1}$$

Then, we can evaluate the error on the symmetry around the axis as:

$$E_{\Delta w}(L) = \frac{1}{L} \int_0^L (\Delta w(l) - \overline{\Delta w})^2 \, dl \tag{2}$$

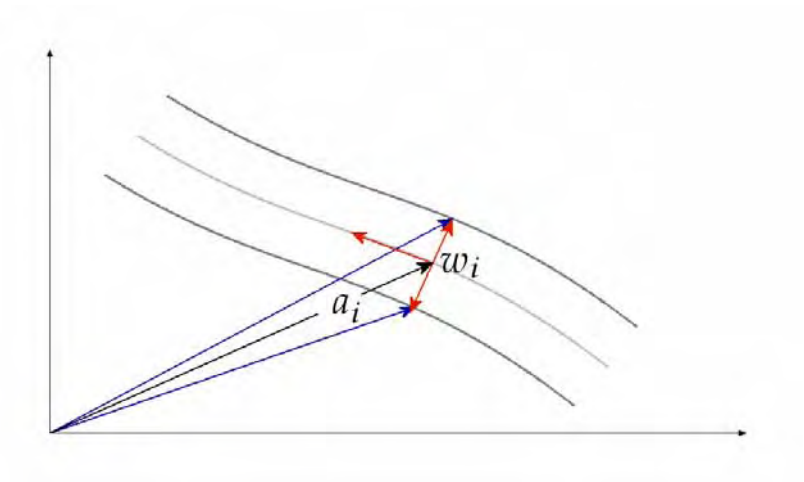


Fig. 1. Curvilinear region definition

In a curvilinear region, this error must be limited by a threshold. In our case, this threshold depends on two parameters, $U_{\Delta w}$ and $\sigma_{\Delta w}$. A curvilinear region complies with:

$$E_{\Delta w}(L) \leq U_{\Delta w}(1 - e^{\frac{-L^2}{2\sigma_{\Delta w}^2}}) \tag{3}$$

II. Ratio between average width and length

If we define *w(l)* as

$$w(l) = w_l(l) + w_r(l) \tag{4}$$

Then, a curvilinear region complies with:

$$L_{\max} \geq U_w \cdot \overline{w} \tag{5}$$

where U_w is a parameter of the method and $L_{\max}$ is the maximum length of the curvilinear region. This length is obtained from all connected pixels inside the region.

III. Left and right borders locally parallel

The mean value of the difference of the tangents at both sides of the region, $\overline{\Delta\alpha}$, must be also bounded. If

$$\Delta\alpha(l) = |\alpha_l(l) - \alpha_r(l)|, \tag{6}$$

then

$$\overline{\Delta\alpha} \leq U_{\Delta\alpha} \tag{7}$$

where $U_{\Delta\alpha}$ is a parameter of the method.

Section 3.5 will present an extended description of these three curvilinear region restrictions.

3.2 Overview of the proposed method

The algorithm for detecting the curvilinear regions works in a simple way. Firstly, the input image is segmented into a set of homogeneous colour regions, so the obtained regions comply with the requirement that colour must be homogeneous through the region. In order to achieve it in a fast way, a pyramid algorithm is employed: the Bounded Irregular Pyramid (BIP) (Marfil et al., 2004). The BIP divides the original image into a set of connected regions which present an homogeneous colour. Then, every image region is checked in order to look for curvilinear regions by analysing its medial axis and borders. Several curvilinear regions can be detected in the same object. Once the curvilinear regions have been extracted from the input image, an extra normalisation step is applied to compensate for part of the deformations (Tuytelaars & Van Gool, 2004). If the curvilinear region is enclosed inside an elliptical region whose centre is obtained as the centre of mass of the region, the normalisation step transforms this elliptical region to a circular reference region of fixed size. Then, normalised curvilinear regions are employed as the input of a shape descriptor. The used shape descriptor is described in Section 4. Basically, it is a contour-based approach to object representation which characterises the region boundary using a curvature function. The obtained contour descriptor is invariant to rotation and translation, and partially invariant to noise, scaling and skewing.

Finally, the approach uses these high-level features for scene recognition. The recognition proceeds with matching individual features to a database of features from known scenes using a nearest-neighbour algorithm based on a curvature matching criterion. The relative pose of recognised features is employed to identify the image layout. Experimental results show that this approach to scene recognition can match images taken from different viewpoints if they present a similar layout, i.e. spatial distribution of curvilinear objects. The image matching process is described in Section 5.2.

3.3 Image segmentation based on the Bounded Irregular Pyramid

In our approach, image segmentation is employed to obtain a global set of image regions. Subsequent stages will perform the region characterisation and they will obtain the final set of curvilinear regions. Particularly, we have used a pyramid segmentation algorithm because these approaches exhibit interesting properties with respect to segmentation algorithms based on a single representation. Thus, local operations can adapt the pyramidal hierarchy to the topology of the image, allowing the detection of global features of interest and representing them at low resolution levels. This general principle was briefly described by Jolion and Montanvert (1992): "*a global interpretation is obtained by a local evidence accumulation*."

In order to accumulate the local evidence, a pyramid represents the contents of an image at multiple levels of abstraction. Each level of this hierarchy is at least defined by a set of vertices V_l connected by a set of edges E_l. These edges define the horizontal relationships of the pyramid and represent the neighbourhood of each vertex at the same level (*intra-level edges*). Another set of edges define the vertical relationships by connecting vertices between adjacent pyramid levels (*inter-level edges*). These inter-level edges establish a dependency relationship between each vertex of level l+1 and a set of vertices at level l (*reduction window*). The vertices belonging to one reduction window are the sons of the vertex which defines it. The value of each parent is computed from the set of values of its sons using a

reduction function. The ratio between the number of vertices at level l and the number of vertices at level l+1 is the *reduction factor*.

Using this general framework, the local evidence accumulation is achieved by the successive building of level $G_{l+1}=(V_{l+1},E_{l+1})$ from level $G_l=(V_l, E_l)$. This procedure consists of three steps:

1. Selection of the vertices of G_{l+1} among V_l: This selection step is a *decimation procedure* and selected vertices V_{l+1} are called the surviving vertices.
2. Inter-level edges definition: Each vertex of G_l is linked to its parent vertex in G_{l+1}. This step defines a partition of V_l.
3. Intra-level edges definition: The set of edges E_{l+1} is obtained by defining the adjacency relationships between the vertices V_{l+1}.

The parent–son relationship defined by the reduction window may be extended by transitivity down to the base level. The set of sons of one vertex in the base level is named its *receptive field*. The receptive field defines the embedding of this vertex in the original image. In a general view of the pyramid hierarchy, the vertices of the bottom pyramidal level (level 0, also called base level) can be anything from an original image pixel via some general numeric property to symbolic information, e.g. a vertex can represent an image pixel grey level or an image edge. Corresponding to the generalization of the vertex contents, the intra-level and inter-level relations of the vertices are also generalized.

After building the pyramidal structure, the segmentation of the input image can be achieved either by selecting a set of vertices from the whole hierarchy as region roots, or by choosing as roots all the vertices which constitute a level of this hierarchy. In any case, this selection process depends on the final application and it must be performed by a higher level task. The efficiency of a pyramid to solve segmentation tasks is strongly influenced by two related features that define the intra-level and inter-level relationships. These features are the data structure used within the pyramid and the decimation scheme used to build one graph from the graph below (Brun & Kropatsch, 2003). The choice of a data structure determines the information that may be encoded at each level of the pyramid and it defines the way in which edges E_{l+1} are obtained. Thus, it roughly corresponds to setting the horizontal properties of the pyramid. On the other hand, the reduction scheme used to build the pyramid determines the dynamics of the pyramid (height, preservation of details, etc.). It defines the surviving vertices of a level and the inter-level edges between levels which correspond to the vertical properties of the pyramid. Taking into account these features, pyramids have been roughly classified as regular and irregular pyramids. A *regular pyramid* has a rigid structure where the intra-level relationships are fixed and the reduction factor is constant. In these pyramids, the inter-level edges are the only relationships that can be changed to adapt the pyramid to the image layout. The inflexibility of these structures has the advantage that the size and the layout of the structure are always fixed and well-known. However, regular pyramids can suffer several problems (Bister et al., 1990): non-connectivity of the obtained receptive fields, shift variance, or incapability to segment elongated objects. In order to avoid these problems, *irregular pyramids* were introduced. In the irregular pyramid framework, the spatial relationships and the reduction factor are not constant. Original irregular pyramids presented a serious drawback with respect to computational efficiency because they gave up the well-defined neighbourhood structure of regular pyramids. Thus, the pyramid size cannot be bounded and hence neither can the time to execute local operations at each level (Willersinn & Kropatsch, 1994). This problem has

been resolved by recently proposed strategies (Brun & Kropatsch, 2003; Haxhimusa et al., 2003; Marfil et al., 2004).

The bounded irregular pyramid (BIP) (Marfil et al., 2004) is a hierarchical structure that merges characteristics from regular and irregular pyramids. Its data structure combines the simplest regular and irregular structures: the 2×2/4 regular one and the simple graph irregular representation. The algorithm firstly tries to work in a regular way by generating, from level l, a 2×2/4 new level l+1. However, only the 2×2 homogeneous arrays of V_l generate a new vertex of V_{l+1}. Therefore, this step creates an incomplete regular level l + 1 which only presents vertices associated to homogeneous regions at the level below. Vertices of level l which generate a new vertex in V_{l+1} are linked to this vertex (son–parent edges). Then, all vertices without parent (orphan vertices) of level l search for a neighbour vertex with a parent in level l + 1 whose colour will be similar to the orphan vertex's colour (*parent search step*). If there are several candidate parents, the orphan vertex is linked to the most similar parent. Finally, the irregular part of the BIP is built. In this step, orphan vertices, of level l, search for all neighbour orphan vertices at the same level. Among the set of candidates, they are linked with the most similar. When two orphan vertices are twined, a new parent is generated at level l + 1 (*intra-level twining step*). This parent is a node of the irregular part of the BIP. The algorithm performs these two steps simultaneously. Thus, if an orphan vertex does not find a parent in the parent search stage, it will search for an orphan neighbour to link to it (*intra-level twining*). In the parent search stage an orphan vertex can be linked with the irregular parent of a neighbour. Once this is completed, intra-level edges are generated at level l + 1. The decimation process stops when it is no longer possible to generate new vertices in the regular part of the BIP. When all the levels are generated, homogeneous vertices without parent are regarded as roots and their corresponding receptive fields constitute the segmented image.

Fig. 2 shows some segmentation results obtained using the proposed algorithm. It can be noted as the different homogeneous regions present in the image have been correctly segmented.

3.4 Medial axis extraction

The geometric properties used to check if a region is curvilinear or not are based on the extraction of the skeleton of the region. The skeleton is defined as a subset of pixels that preserve the topological information of the region and it must approximate the medial axis. There are a lot of methods to estimate the skeleton of an object and they are either based on distance transforms defined by different metrics or algorithms based on simple shape deformations (Klette, 2003). The choice of the method often depends on the task, as there is no "best method". One category is based on distance transforms, where a *distance skeleton* is a subset of grid points such that every point of this subset represents the centre of a maximal disc contained in the given component. A second category is based on iterative thinning methods, where the term *linear skeleton* can be used for the result of a continuous deformation of the frontier of a connected subset without changing the connectivity of the original set, until only a set of lines and points remains. In this work a distance transformed approach is used for each colour segmented region, therefore obtaining a skeleton for each region. This skeleton will be used to estimate further geometric properties.

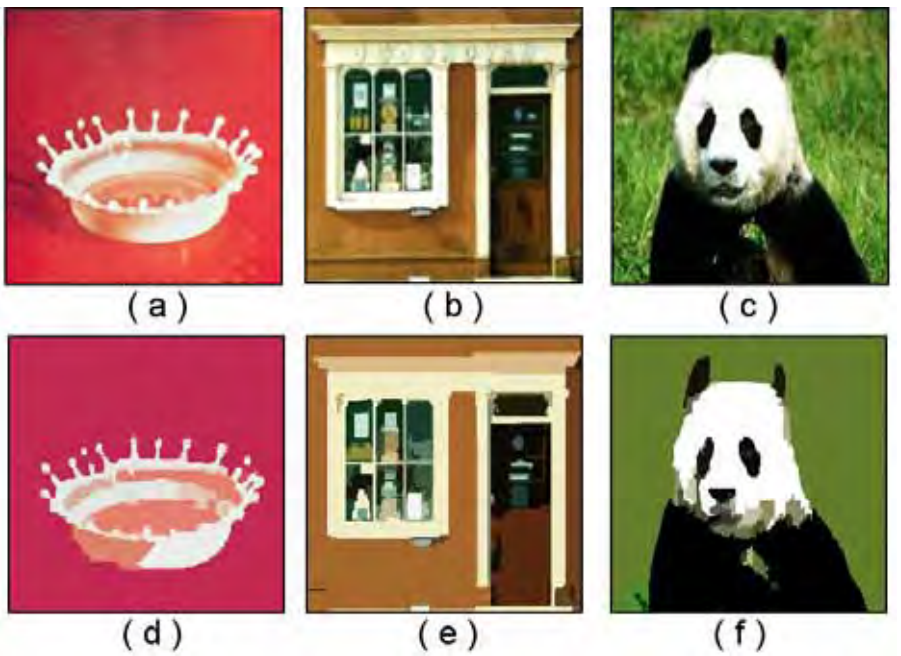


Fig. 2. Segmentation results obtained using the BIP structure (Marfil et al., 2004): a-c) original images; and d-f) segmentation results.

The distance from one point to another is the smallest positive integer n such that there exists a sequence of distinct points $p_0, p_1, p_2 \ldots p_n$ with p_i being an a-neighbour of p_{i-1}, $1 \le i \le n$. For $a = 8$, the distance $d(p,q)$ is called the d_8-distance. If (i_p, j_p) and (i_q, j_q) are the coordinates of p and q respectively, then

$$d_8(p,q) = \max\left\{\left|i_p - i_q\right|, \left|j_p - j_q\right|\right\} \tag{8}$$

For estimating the distance transform of a region we use the algorithm described in (Klette, 2003) which can approximate the distance transform inside the region in only two steps, so it has got a low computational cost. We define the original region as an image: $I(i,j) = 0$, if the pixel (i, j) belongs to the border of the region, and $I(i,j) = 255$ otherwise. In the first step the function f_1 is defined as

$$f_1(i,j,I(i,j)) \begin{cases} 0 \;\; if \;\; I(i,j)=0 \\ \min\left\{I^*(i-1,j)+1, I^*(i,j-1)+1, I^*(i-1,j-1)+1, I^*(i-1,j+1)+1\right\} \\ \qquad if \;\; I(i,j)=255 \;\; and \;\; i \neq 1 \;\; or \;\; j \neq 1 \\ i+j \;\; otherwise \end{cases} \tag{9}$$

The function f_1 is applied to the image I from top to bottom and from left to right, producing $I^*(i, j) = f_1(i, j, I(i, j))$. In the second step the function f_2 is defined as

$$f_2(i,j,I^*(i,j)) = \min\left\{I^*(i,j), T(i+1,j)+1, T(i,j+1)+1, T(i+1,j-1)+1, T(i+1,j+1)+1\right\} \tag{10}$$

and the resulting image T is calculated as $T(i, j) = f_2(i, j, I^*(i,j))$, applying f_2 from bottom to top and from right to left, and being T the distance transform image of I. If we choose those pixels (i_s, j_s) in the image T such as none of the points in the vicinity $A_8((i_s, j_s))$ has a value in T equal to $T(i_s, j_s)+1$ then those pixels (i_s, j_s) belong to the distance skeleton and they are supposed to be local maxima in the distance transform.
The resulting distance skeletons are generally not connected, so we post-process them with morphological operations (interpolation, dilatation, erosion and elimination of not useful pixels) to obtain a connected and smooth skeleton. By this way we obtain an approximation to the medial axis of the object.

3.5 Skeleton classification

Once the skeletons are calculated for each segmented region our method decides which parts of the skeleton belong to a curvilinear region and which not. In order to achieve this goal, several geometric characteristics are estimated: symmetry around the skeleton, ratio between average width and length, and borders parallelism (see Section 3.1).

3.5.1 Symmetry around the skeleton

The method checks those pixels which comply with the requirement of (3). To describe the algorithm we can define a skeleton as the set of connected pixels $p_s=(i_s, j_s)$, $0 \le s \le N-1$, and N the number of pixels being evaluated of the skeleton. In a first step, the normal vector is calculated for each pixel p_s in the skeleton, and the cross-points between the normal and the left and right borders of the region are estimated. If we define p_s^l and p_s^r as these cross-points, then we obtain the triplets (p_s, p_s^l, p_s^r), $0 \le s \le N-1$. We can implement (3) as

$$\frac{1}{N}\sum_{s=0}^{N-1}\left(\Delta w_s - \overline{\Delta w}\right)^2 \le U_{\Delta w}\left(1 - e^{-\frac{N^2}{2\sigma_{\Delta w}^2}}\right) \tag{11}$$

with

$$\Delta w_s = \left| w_s^l - w_s^r \right| \tag{12}$$

$$\overline{\Delta w} = \frac{1}{N}\sum_{s=0}^{N-1}\Delta w_s \tag{13}$$

being w_s^l the Euclidean distance between pixels p_s and p_s^l and w_s^r the Euclidean distance between pixels p_s and p_s^r.
The left side in (11) is a term that grows with the asymmetries of the region and the values $U_{\Delta w}$ and $\sigma_{\Delta w}$ in the right side are parameters of the method. For our experiments, we have used $U_{\Delta w} = 10$ and $\sigma_{\Delta w} = \sqrt{50}$. The number of pixels N also appears on the right side of (11), in a way that longer regions are allowed to have a higher value of asymmetry.

3.5.2 Ratio $L/\overline{w}$

In a similar way that Section 3.5.1, we define w_s as the width of the region estimated as the Euclidean distance between pixels p_s^l and p_s^r given a position s in the skeleton. Then, (5) is implemented as

$$L_{\max} \geq U_w \cdot \frac{1}{N} \sum_{s=0}^{N-1} w_s \tag{14}$$

$L_{\max}$ is the maximum length that the curvilinear skeleton could have and is calculated with all the connected pixels of the skeleton of the object. U_w is also a parameter of the method. In our experiments, it has been set to 1.5.

3.5.3 Borders parallelism

To check the borders parallelism requirement we estimate the tangential vectors on the borders at pixels p_s^l and p_s^r. Then, we calculate the angle between those vectors and the normal vector given a position s, obtaining angles α_s^l and α_s^r. Equation (7) is implemented as

$$\frac{1}{N} \sum_{s=0}^{N-1} \left| \alpha_s^l - \alpha_s^r \right| \leq U_{\Delta\alpha} \tag{15}$$

$U_{\Delta\alpha}$ is a parameter of the method. For our experiments, it has been set to 30 degrees.

3.5.4 Classification algorithm

The algorithm to classify the skeletons into the curvilinear group or the not curvilinear one works in an easy way. Once the skeleton has been extracted from the distance transform image associated to an object, the algorithm tries to join as many pixels as possible to form a curvilinear skeleton. So the algorithm begins in an endpoint of the skeleton and it looks for adding the connected pixels checking if (11), (14) and (15) are true with each new added pixel. If these equations are true for a pixel, then the new pixel is added and the algorithm will check the next connected pixel in the extracted skeleton. If the new pixel does not comply with all the requirements, then the curvilinear skeleton is finished and a new curvilinear region will begin with the next positive evaluation.

Given an object and its skeleton, when all the pixels have been evaluated, the curvilinear skeletons whose endpoints are near are linked to form a longer curvilinear skeleton. At the end of the process, the parts of the objects whose skeleton has been evaluated as a curvilinear skeleton are considered as curvilinear regions. The algorithm allows to demand a minimum length L_{min} to the regions. In our experiments, the minimum length has been set to 10 pixels.

Figs. 3 and 4 present an experiment with a real scene obtained using our typical set of parameters. In Fig. 3, the results of the detection of objects and classification of the extracted skeletons are presented. Fig. 4 presents the original scene with the curvilinear skeletons superimposed.

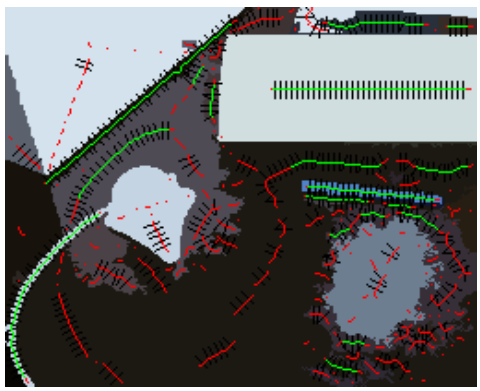


Fig. 3. Detected segmented regions in a segmentation image. The extracted skeletons have been drawn (in green colour the skeletons classified as curvilinear and in red colour as not curvilinear). Also some estimated normal vectors (black colour) to the skeletons have been drawn.

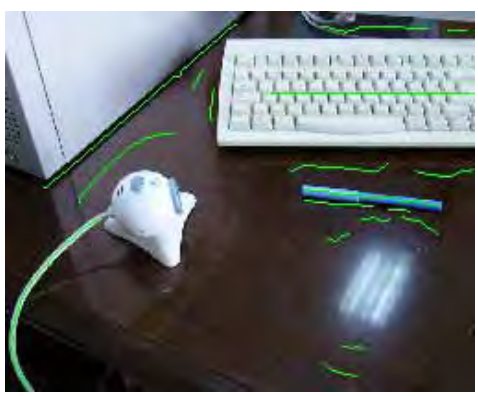


Fig. 4. Original image with the detected curvilinear skeletons (see Fig. 3). Several interesting objects as the ball pen, keyboard and webcam cable have been detected. Parameters used are: segmentation threshold = 95.0, $U_{\Delta w}$ = 10, $\sigma_{\Delta w} = \sqrt{50}$, U_w =1.5, $U_{\Delta\alpha}$ =30º, L_{min} = 10 pixels.

3.6. Normalisation stage

As it is pointed out by Tuytelaars and Van Gool (2004), it is better to compensate for part of the geometric deformations through a normalisation stage, before obtaining the descriptor associated to the region. In our case, the geometric normalisation stage will be achieved by enclosing the curvilinear region inside an elliptically-shaped region and by transforming this region to a circular reference region of fixed size (see Fig. 5). This process leaves one degree of freedom to be determined which corresponds to a free rotation of the circular region around its centre. In our case, it is not a problem because the shape will be represented using a contour descriptor which is invariant to rotation distortions.

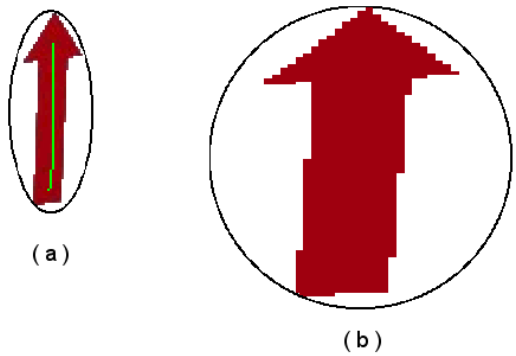


Fig. 5. a) Original curvilinear region; and b) normalised region.

4. Shape description

Once the curvilinear regions have been extracted from the input image, they are characterised using a shape descriptor. Shape representation constitutes one of the most powerful tools to represent a planar object. Therefore, many approaches have been proposed to describe shapes from a small set of features. These descriptors can be divided into those which work on a shape as a whole (*global descriptors*) and those which work on the contours of the shape (*boundary-based descriptors*). Boundary-based descriptors are less computationally intense than global ones. However, since they are based on the shape contour, they cannot take into account the internal structure of the object. Therefore, boundary-based methods are not suited to deal with certain kinds of applications. On the other hand, most of the boundary-based descriptors do not need to normalise the 2D representation of the object to achieve common geometrical invariance. Thus, a boundary-based method, the popular *curvature scale space* (Mokhtarian & Mackworth, 1986), has been used in the MPEG-7 standard.

In this work, we employ a boundary-based descriptor. Particularly, this descriptor is based on the estimation of the curvature associated to the shape contour. By definition, the curvature function encodes the shape contour in terms of their local curvature or orientation. If $c(t)=(x(t), y(t))$ is a parametric plane curve, then its curvature function $\kappa(t)$ can be calculated as (Mokhtarian & Mackworth, 1986)

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}} \tag{16}$$

This equation implies that estimating the curvature involves the first and second order directional derivatives of the plane curve co-ordinates. This is a problem in the case of computational analysis where the plane curve is represented in a digital form. In order to solve this problem, two different approaches are often encountered: those that approximate the plane curve co-ordinates (*interpolation-based curvature estimators*), and those that estimate the curve orientation at each contour point with respect to a reference direction (*angle-based curvature estimators*). In addition, both type of methods can be subdivided in single scale methods and multiscale ones. Single scale methods are based upon an analysis of the

contour using a fixed set of parameters. Multiscale methods represent the evolution (or deformation) of the original contour when a certain parameter value is varied.
The described shape descriptor is grouped into the angle-based curvature estimators. These approaches propose an alternative curvature measure based on angles between vectors which are defined as a function of the curve co-ordinates. Thus, the contour curvature $\kappa(t)$ can be defined as the variation of the curve slope $\psi(t)$ with respect to t, that is, the inverse of the curvature radius $\rho(t)$:

$$\kappa(t) = \frac{\partial \psi(t)}{\partial t} = \frac{1}{\rho(t)} \tag{17}$$

In order to extract $\kappa(t)$ from a digital contour, several methods have been proposed. The majority of these approaches consist of comparing segments of k-points at both sides of a given point to estimate its curvature. Therefore, the value of k determines the cut frequency of the curve filtering. So, these algorithms are single scale methods in which only features unaffected by the filtering process may be detected. On the contrary, Beus and Tiu (1987) propose a multiscale angle-based approach which modifies the Freeman's approach (Freeman, 1978) by averaging the results obtained for several values of k. However, this approach is slow and, in any case, it must choose the cut frequencies for each iteration (Bandera et al., 2000).
Another solution is to adapt the cut frequency of the filter at each curve point as a function of the local properties of the shape around it. A k-slope algorithm which estimates the curvature using a k value which is adaptively changed according to the local information of the boundary is proposed by Bandera et al. (2000). In this work, we will employ this curvature estimator. Thus, Fig. 6 shows several examples of curvature functions associated to different shape contours.

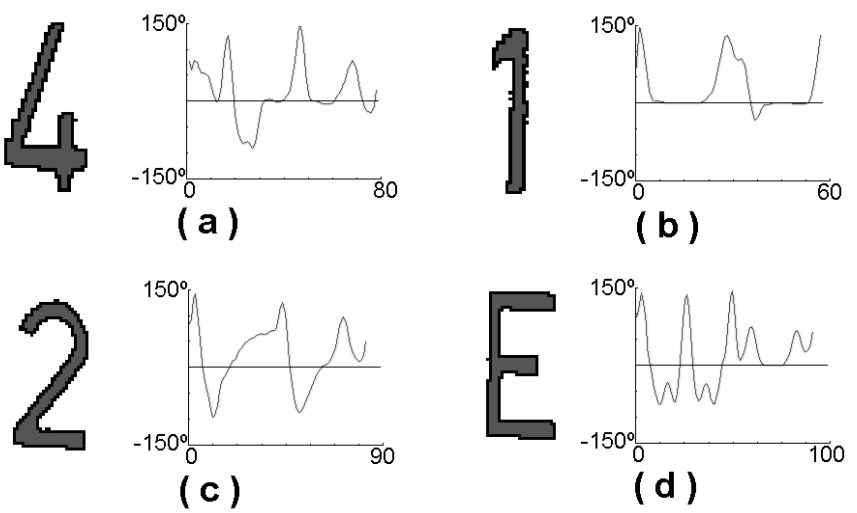


Fig. 6. a-d) Curvilinear region shapes and associated curvature functions

5. Experimental results

5.1. Shape description: a comparative study

The proposed shape descriptor has been compared to other methods to test its performance. Particularly, we chose for the purpose of comparison the methods proposed by Bernier and Landry (2003) and Zhang and Lu (2005). The first method employs a contour-based descriptor, whereas the second one is rather region-based. In order to compare the performance of the different methods, a publicly available data set (Sebastian et al., 2001) was employed[1]. This data set consists of nine classes with eleven shapes in each cluster (see Fig. 7).

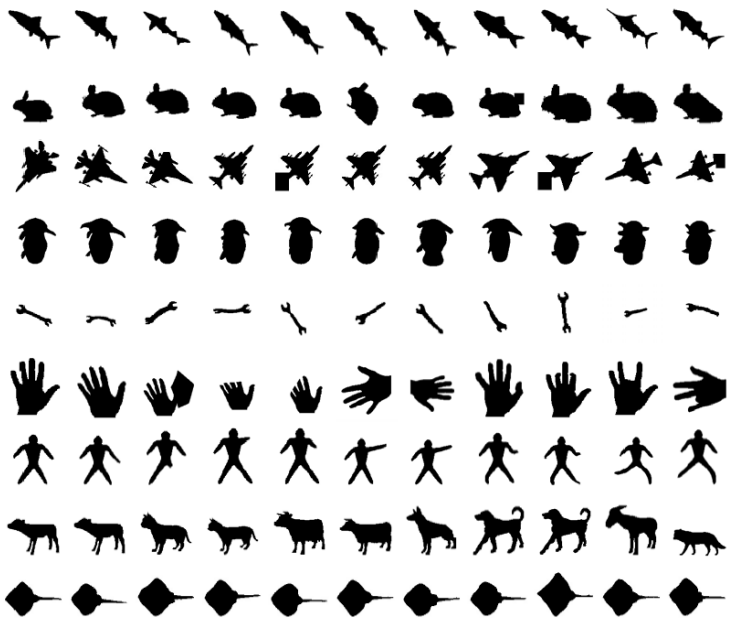


Fig. 7. A data set of 99 shapes (Sebastian et al., 2001)

The experiments were performed on a Pentium IV 2.6 GHz PC. Each shape was matched against all the other shapes of the data set and the number of times the test image was correctly classified was counted in the *n*th nearest neighbours (*n* ranging from 1 to 8) (Tabbone et al., 2006). Fig. 8 shows the *n*th nearest match rates for each approach. Although the results of the first nearest matches were quite similar among all methods, the results for the matches from 5 to 8 were better with our approach. Finally, it must be mentioned that these results are quite similar to the ones reported by Tabbone et al. (2006) which use a more computationally expensive shape descriptor defined on the Radon transform.

[1] http://www.lems.brown.edu/vision/researchAreas/SIID/

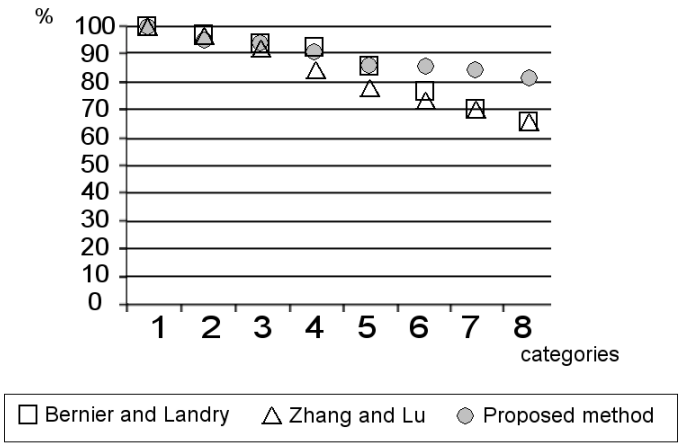


Fig. 8. Comparison of the employed shape descriptor with other approaches (see text)

5.2. Scene recognition experiments

Once the curvilinear regions have been detected, they are characterised using a 260-dimensional space whose first two dimensions $(x, y)_i$ are the co-ordinates of the centre of mass of the region (the image co-ordinates are ranged from 0 to 256), the second two dimensions $(h, s)_i$ are the mean hue and saturation values of the region (HSV colour space), and the other 256 values $\{fc_i\}_{i=1...256}$ are the curvature function of the object shape. Each image is then described by the properties of the associated set of curvilinear regions.

In this image matching scheme, two images will be similar if their associated sets of curvilinear regions are similar. The distance between two curvilinear regions i and j can be defined as

$$D(i,j) = \alpha_1 \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \alpha_2 \sqrt{s_i^2 + s_j^2 - 2 s_i s_j \cos\theta} + \alpha_3 \max\{(fc_i * fc_j)_{k=1...256}\} \tag{18}$$

where θ is equal to $|h_i - h_j|$ if this value is less than π, or equal to $(2\pi - |h_i - h_j|)$ in any other case. The parameters a_i define the importance of the position, colour and shape into the distance measure and they have been experimentally adjusted. The * operator denotes the convolution and it is applied ranging from 1 to 256, providing rotation invariance. Then, given a query image Q and a dataset of images B_i, whose associated sets of curvilinear regions have been detected and characterised off-line, the image matching process firstly extracts the set of N_Q curvilinear regions $\{cQ\}_{i=1...NQ}$ present in the query image. They are sorted as a function of their lengths. Then, the comparison between Q and each image B_i is achieved by comparing each curvilinear region in Q, cQ_i, with all the N_{Bi} curvilinear regions present in B_i, $\{cBi\}_{i=1...NBi}$, using (18). The most similar region is selected and, if the similarity value, $D(cQ_i, cBi_j)$, is less than a given threshold U, both curvilinear regions are paired. This implies that the selected curvilinear region of B_i cannot be paired with other curvilinear region of Q. Finally, a similarity value is assigned to the comparison between images Q and Bi. This value is defined as

$$\lambda = \left(N_{Bi} - N'_Q + 1\right)\sum_{i=1}^{N'_Q} D(cQ_i, cBi_j) \tag{19}$$

where N'_Q is the number of paired curvilinear regions.

The images B_i are then sorted according to the obtained similarity values. To test the method, a database of 40 images obtained in an office-like environment has been created. This database can be divided into 10 different scenarios (4 different images for each scenario). Fig. 9 presents two example retrievals for this database. Query is the leftmost image in each row, and subsequent images are nearest neighbours. Detected curvilinear regions employed to match both images have been marked.

To evaluate the matching performance, we have employed the normalised average rank $\overline{R}$ (Grauman & Darrell, 2005)

$$\overline{R} = \frac{1}{NN_R}\left(\sum_{i=1}^{N_R} R_i - \frac{N_R(N_R - 1)}{2}\right) \tag{20}$$

where R_i is the rank at which the ith relevant image is retrieved, N_R is the number of relevant images for a given query, and N is the number of examples in the database. A normalised average rank equal to 0 implies a perfect performance, that is all relevant images in the database have been retrieved as nearest neighbours of the query image. For the reported experiment, the normalised average rank of relevant images present an average value of 0.025 and a standard deviation of 0.001.

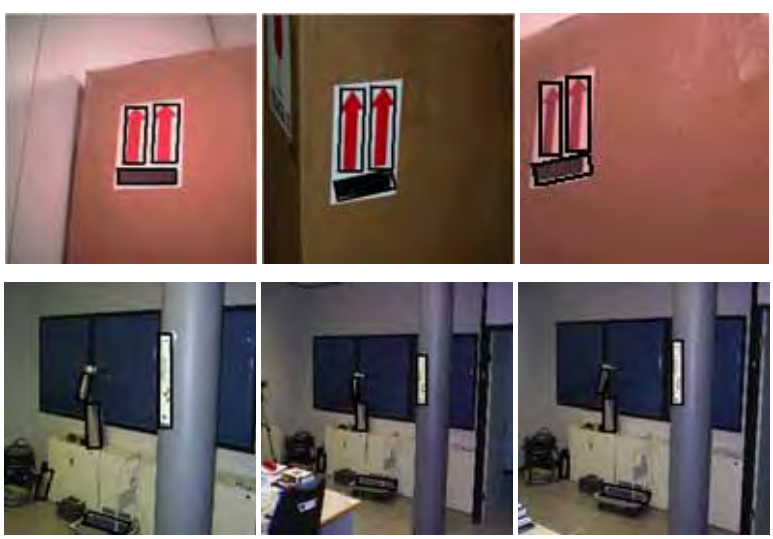


Fig. 9. Example retrievals for a database of office-like environment images (see text for details)

6. Conclusions and future work

This chapter presents a method for image matching which is based on the detection and characterisation of curvilinear regions. In a curvilinear region, the ratio between its average width and its total length should be less than a predefined threshold. Besides, left and right borders should be locally parallel, it should exist a geometric similarity around the region axis and the colour along this axis should be homogeneous. That is, they constitute particular image structures and, therefore, the method is restricted to scenes where these particular items are presented. On the contrary, curvilinear regions automatically deform with changing viewpoint as to keep on covering identical physical parts of a scene. For this reason, they can be used as distinguished regions. The shape contour of these curvilinear regions is characterised using the adaptive curvature function. Experimental results show that this shape descriptor is invariant to rotation and translation, and partially invariant to noise and skewing. Scaling invariance is achieved by employing an extra normalisation stage. Thus, this descriptor, plus the region colour and position, can be used to match curvilinear regions detected on the input image with those previously stored in a database. This is the basis of the correspondence algorithm described in this paper: the similarity index between two images is determined by the presence of the same set of curvilinear regions localised in similar positions.

There are many directions for further research. One of this is the integration of several types of distinguished regions. As we commented above, the obligatory presence of these regions in the images is the main disadvantage of the proposed system. Besides, further work must be accomplished in the correspondence algorithm to employ the most similar region correspondences as ground control points. These points could be used to generate a fundamental matrix describing the geometric constraints between the two images. Thus, matches that did not agree with the majority solution could be removed.

7. Acknowledgments

This work has been partially granted by the Spanish Ministerio de Ciencia y Educación (MEC), under project nº TIN2005-01359.

8. References

Bala, E. & Cetin, A.E. (2004). Computationally efficient wavelet affine invariant functions for shape recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, 1095-1098

Bandera, A.; Urdiales, C.; Arrebola, F. & Sandoval, F. (2000). Corner detection by means of adaptively estimated curvature function. *Electronics Letters*, Vol. 36, No. 2, 124-126

Baumberg, A. (2000). Reliable feature matching across widely separated views. *Proc. of the Conference on Computer Vision and Pattern Recognition*, 774-781

Bernier, T. & Laundry, J.A. (2003). A new method for representing and matching shapes of natural objects. *Pattern Recognition*, Vol. 36, 1711-1723

Beus, L. & Tiu, S. (1987). An improved corner detection algorithm based on chain-coded plane curves. *Pattern Recognition*, Vol. 20, No. 3, 291-296

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, Vol. 94, No. 2, 115-147

Bister, M.; Cornelis, J. & Rosenfeld, A. (1990). A critical view of pyramid segmentation algorithms. *Pattern Recognition Lett.*, Vol. 11, 605–617

Brun, L. & Kropatsch, W.G. (2003). Construction of combinatorial pyramids, in: *Graph Based Representations in Pattern Recognition,* E. Hancock & M. Vento (Eds.), Vol. 2726, 1-12

Carneiro, G. & Jepson, A.D. (2002). Phase-based local features. *Proc. of the European Conference on Computer Vision (ECCV)*, 282-296

Crowley, J. L. & Parker, A.C. (1984). A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 2, 156-170

Dai, X.L. & Lu, J. (1999). An object-based approach to automated image matching. *Proc. of the IEEE Int. Conf. on Geoscience and Remote Sensing Symposium,* Vol. 2, 1189-1191

Deng, H.; Zhang, W.; Dietterich, T. & Mortensen, E. (2006). *A comparative evaluation of a new curvilinear region detector for object recognition*. Technical Report, Electrical Engineering and Computer Science, Oregon State University

Dufournaud, Y.; Schmid, C. & Horaud, R. (2000). Matching image with different resolutions. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 612-618

Freeman, H. (1978). Shape description via the use of critical points. *Pattern Recognition*, Vol. 10, 159-166

Grauman, K. & Darrell, T. (2005). Efficient image matching with distributions of local invariant features. *Proc. of IEEE Conf. Computer Vision and Pattern Recogn.,* 627-634

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. *Proc. of the Fourth Alvey Vision Conference*, 147-151

Haxhimusa, Y.; Glantz, R. & Kropatsch, W.G. (2003) Constructing stochastic pyramids by MIDES—maximal independent directed edge set, in: *Fourth IAPR-RC15 Workshop on GbR in Pattern Recognition,* E. Hancock & M. Vento (Eds.), Vol. 2726, 35-46

Jolion, J.M. & Montanvert, A. (1992). The adaptive pyramid, a framework for 2D image analysis. *CVGIP: Image Understanding*, Vol. 55, 339–348

Klette, G. (2003). A comparative discussion of distance transformations and simple deformations in digital image processing. *Machine Graphics & Vision*, Vol. 12, No. 2, 235-256

Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, Vol. 11, No. 3, 283-318

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, Vol. 21, No. 2, 224-270

Lowe, D.G. (1999). Object recognition from local scale-invariant features. *Proc. of the International Conference on Computer Vision*, 1150-1157

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, Vol. 60, No. 2, 91-110

Marfil, R.; Rodríguez, J.A.; Bandera, A. & Sandoval, F. (2004). Bounded irregular pyramid: a new structure for color image segmentation. *Pattern Recognition*, Vol. 37, No. 3, 623-626

Matas, J.; Chum, O.; Urban, M. & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. *Proc. of the British Machine Vision Conf.*, 384-393

Mikolajczyk, K. & Schmid, C. (2001). Indexing based on scale invariant interest points. *Proc. of the 8th Int. Conf. on Computer Vision*, 525-531

Mikolajczyk, K. (2002). *Detection of local features invariant to affine transformations*, Ph.D. thesis, Institut National Polytechnique de Grenoble, France.

Mikolajczyk, K.; Zisserman, A. & Schmid, C. (2003). Shape recognition with edge-based features. *Proc. of the British Machine Vision Conference*, 799-788

Mikolajczyk, K. & Schmid, C. (2004). Scale and affine invariant interest point detectors. *Int. Journal on Computer Vision*, Vol. 1, No. 60, 63-86

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, 1615-1630

Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T. & Van Gool, L. (2005). A comparison of affine region detectors. *Int. Journal on Computer Vision*, Vol. 65, No. 1/2, 43-72

Mokhtarian, F. & Mackworth, A. (1986). Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Analysis and Machine Intell.*, Vol. 8, No. 1, 34-43

Montesinos, P.; Gouet, V. & Pele, D. (2000). Matching color uncalibrated images using differential invariants. *Image and Vision Computing*, Vol. 18, No. 9, 659-671

Moravec, H. (1981). Rover visual obstacle avoidance. *Proc. of the Int. Joint Conference on Artificial Intelligence*, 785-790

Nelson, R.C. & Selinger, A. (1998). Large-scale tests of a keyed, appearance-based 3-D object recognition system. *Vision Research*, Vol. 38, No. 15, 2469-2488

Pope, A.R. & Lowe, D.G. (2000). Probabilistic models of appearance for 3-D object recognition. *International Journal of Computer Vision*, Vol. 40, No. 2, 149-167

Schiele, B. & Crowley, J.L. (2000). Recognition without correspondence using multi-dimensional receptive field histograms. *International Journal of Computer Vision*, Vol. 36, No. 1, 31-50

Schmid, C. & Mohr, R. (1997). Local gray value invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vo. 19, No. 5, 530-534

Sebastian, T.; Klein, P. & Kimia, B. (2001). Recognition of shapes by editing shock graphs. *Proc. of the ICCV'2001*, 755-762

Shokoufandeh, A.; Marsic, I. & Dickinson, S.J. (1999). View-based object recognition using saliency maps. *Image and Vision Computing*, Vol. 17, 445-460

Tabbone, S.; Wendling, L. & Salmon, J.P. (2006). A new shape descriptor defined on the Radon transform. *Computer Vis. Image Understanding*, Vol. 102, 42-51

Tuytelaars, T. & Van Gool, L. (2004). Matching widely separated views based on affine invariant regions. *Int. Journal of Computer Vision*, Vol. 59, No. 1, 61-85

Willersinn, D. & Kropatsch, W.G. (1994). Dual graph contraction for irregular pyramids. *Proc. of the 12th IAPR International Conference on Pattern Recognition*, Vol. 3, 251–256

Zhang, Z.; Deriche, R.; Faugeras, O. & Luong, Q.T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, Vol. 78, 87-119

Zhang, D. & Lu, G. (2005). Study and evaluation of different Fourier methods for image retrieval. *Image and Vision Computing*, Vol. 23, 33-49

10

An Overview of Advances of Pattern Recognition Systems in Computer Vision

Kidiyo Kpalma and Joseph Ronsin
IETR (Institut d'Electronique et de Télécommunications de Rennes)
UMR – CNRS 6164
Groupe Image et Télédétection
Institut National des Sciences Appliquées (INSA) de Rennes

1. Introduction

First of all, let's give a tentative answer to the following question: **what is pattern recognition (PR)?** Among all the possible existing answers, that which we consider being the best adapted to the situation and to the concern of this chapter is: "pattern recognition is the scientific discipline of machine learning (or artificial intelligence) that aims at classifying data (patterns) into a number of categories or classes". **But what is a pattern?**
In 1985, Satoshi Watanabe (Watanabe, 1985) defined a pattern as "the opposite of chaos; it is an entity, vaguely defined, that could be given a name." In other words, a pattern can be any entity of interest which one needs to recognise and/or identify: it is so worthy that one would like to know its name (its identity). Examples of patterns are: a pixel in an image, a 2D or 3D shape, a typewritten or handwritten character, the gait of an individual, a gesture, a fingerprint, a footprint, a human face, the voice of an individual, a speech signal, ECG time series, a building, a shape of an animal.
A pattern recognition system (PRS) is an automatic system that aims at classifying the input pattern into a specific class. It proceeds into two successive tasks: (1) the **analysis** (or description) that extracts the characteristics from the pattern being studied and (2) the **classification** (or recognition) that enables us to recognise an object (or a pattern) by using some characteristics derived from the first task.
The classification scheme is usually based on the availability of the training set that is a set of patterns already having been classified. This learning strategy is termed as supervised learning in opposition to the unsupervised learning. A learning strategy is said to be unsupervised if for the system is not given an a priori information about classes; it establishes the classes itself based on the regularities of the features. Features are those measurements which are extracted from a pattern to represent it in the features space. In other words, pattern analysis enables us to use some features to describe and represent it instead of using the pattern itself. Also called characteristics, attributes or signatures the recognition efficiency and reliability are dependent on their choice.
Pattern recognition constitutes an important tool in various application domains, but unfortunately, that is not always an easy task to carry out. Commonly, one can encounter four major methodologies in PRSs; which are: statistical approach, syntactic approach,

template matching, neural networks. In this chapter, our remarks and details will be directed, mainly, towards systems based on the statistical approach since it is the more commonly used in practice.

1.1 Statistical approach

Typically, statistical PRSs are based on statistics and probabilities. In these systems, features are converted to numbers which are placed into a vector to represent the pattern. This approach is most intensively used in practice because it is the simplest to handle.
In this approach, patterns to be classified are represented by a set of features defining a specific multidimensional vector: by doing so, each pattern is represented by a point in the multidimensional features space. To compare patterns, this approach uses measures by observing distances between points in this statistical space. For more details and deeper considerations on this approach, one can refer to (Jain, 2000) that presents a review of statistical pattern recognition approaches.

1.2 Syntactic approach

Also called structural PRSs, these systems are based on the relation between features. In this approach, patterns are represented by structures which can take into account more complex relations between features than numerical feature vectors used in statistical PRSs (Venguerov & Cunningham, 1998). Patterns are described in hierarchical structure composed of sub-structures composed themselves of smaller sub-structures.
As explained in (Sonka et al., 1993), the shape is represented with a set of predefined primitives called the codebook and the primitives are called codewords. For example, given the codewords on the left of figure 1, the shape on the right of the figure can be represented as the following string S, when starting from the pointed codeword on the figure:

$$S = d\ b\ a\ b\ c\ b\ a\ b\ d\ b\ a\ b\ c\ b\ a\ b \tag{1}$$

The system parses the set of extracted features using a kind of predefined grammar. If the whole features extracted from a pattern can be parsed to the grammar then the system has recognised the pattern. Unfortunately, grammar-based syntactic pattern recognition is generally very difficult to handle.

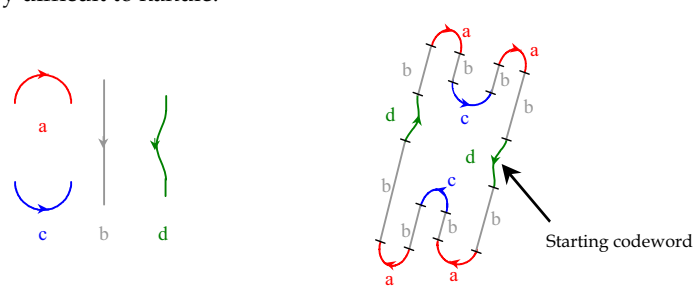


Fig. 1. Example of syntactic description features

1.3 Template matching

Template matching approach is widely used in image processing to localize and identify shapes in an image. In this approach, one looks for parts in an image which match a

template (or model). In visual pattern recognition, one compares the template function to the input image by maximising the spatial cross-correlation or by minimising a distance: that provides the matching rate.

The strategy of this approach is: for each possible position (in the image), each possible rotation, or each other geometric transformation of the template, compare each pixel's neighbourhood to this template. After computing the matching rate for each possibility, select the largest one, that exceeds a predefined threshold. It is a very expensive operation while dealing with big templates and/or large sets of images (Brunelli & Poggio, 1997 ; Roberts & Everson, 2001 ; Cole et al., 2004). Figures 2 illustrate a pattern recognition based on the template matching approach. Figure 2.a is the input image I, Fig.1.b represents two templates (K representing letter 'K' and P letter 'P'). Figures 2.c and 2.d represent, respectively, the normalized cross-correlation of I with K and the normalized cross-correlation of I with P. On these two images, the cross-correlation peaks surrounded by a circle indicate the location of the most matching letter in the input image. On figure 2.e, we have superposed the templates on the input image, accordingly to the coordinates of corresponding correlation peaks. For this study, we didn't take the rotation and the scaling into account: from the result, it clearly appears that this approach retrieves only the shape that matches perfectly the model (size and rotation). This explains why only one 'K' (the rotated one) and only one 'P' (the down-scaled one) are recognised.

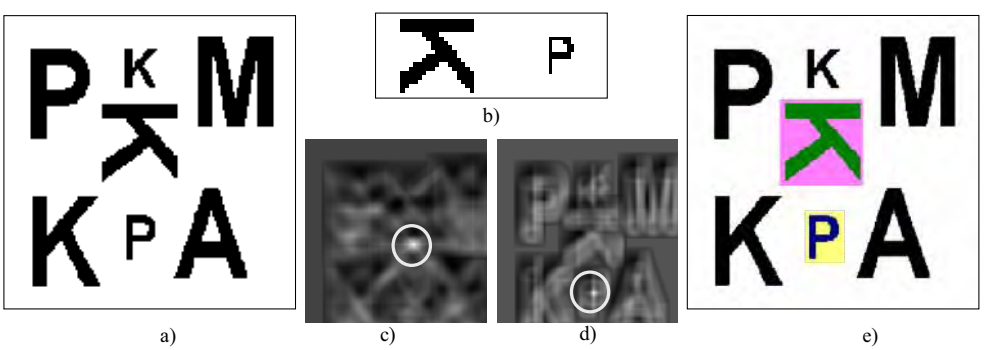


Fig. 2. Illustration of the template matching method

1.4 Neural networks

Typically, an artificial neural network (ANN) is a self-adaptive trainable process that is able to learn to resolve complex problems based on available knowledge. A set of available data is supplied to the system so that it finds the most adapted function among an allowed class of functions that matches the input.

An ANN-based system simulates how the biological brain works: it is composed of interconnected processing elements (PE) that simulate neurones. Using this interconnection (or synapse), each neurone (or PE) can pass information to another. As can be seen on figure 3, these interconnections are not necessarily binary (on or off) but they may have varying weights defined by the weight matrix W: the weight applied to a connection results from the learning process and indicates the importance of the contribution of the preceding neurone

in the information being passed to the following neurone. Figure 3 shows a simple neural network representing the Perceptron as defined by Frank Rosenblatt in 1957. On this example, the output Out_j (j=1 or 2) is defined by a weighted combination of the inputs. In the reference (Abdi, 1994), the author presents a nice introduction to ANNs.
Besides these approaches, one can encounter other methodologies like those based on fuzzy-set theoretic, genetic algorithms. In some applications, hybrid methodologies combine different aspects of these approaches to design more complex PRSs. In (Liu et al., 2006), the authors present an overview of pattern recognition approaches and the classification of their associated applications.

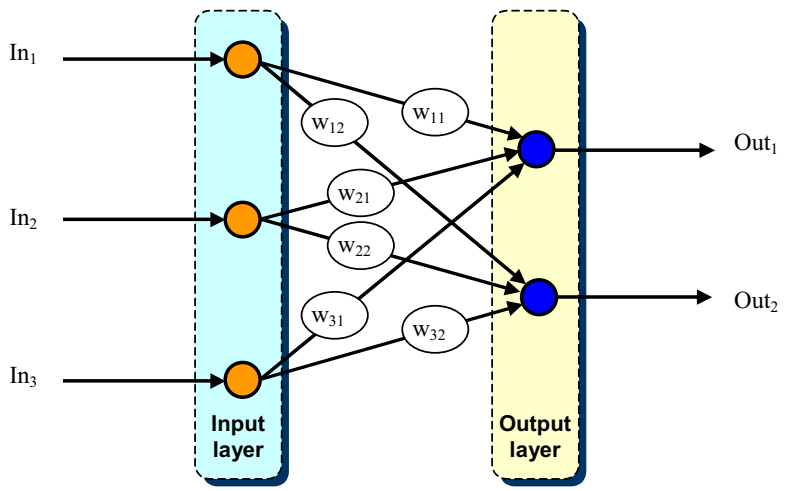


Fig. 3. Example of neural network

In the remainder of this chapter, we will develop three sections. First, we present a generic scheme of a pattern recognition system. Then we give an overview of the advances of different PRSs and some examples of their applications. Last, as an illustration, we present a specific application example based on our MSGPR (Multi-Scale curve smoothing for Generalised Pattern Recognition) description method. As presented further, MSGPR is a multi-scale method we have developed for describing planar objects by analysing their boundary.

2. A generic scheme of a pattern recognition system

From now, our concerns will be primarily focused on PRSs in computer vision. Commonly, in this field, the input is one or more images and the output is one or more images with eventually, some semantic and/or textual entities.
In figure 4, we represent a generic scheme of a (statistical) PRS. This figure summarises the principal aspects of a PRS in computer vision. On this figure the two successive tasks can be observed: on one hand, the analysis/description task (see ❶ on figure 4) and on the other hand the classification/recognition task (see ❷ on figure 4).
After features are extracted, the features selection that may follow aims at reducing the number of features to be provided to the classification process. Features that are likely to

improve discrimination are retained and the others are discarded. During this processing, higher level features can be derived by combining and/or transforming low level features, e.g. by applying the so called independent component analysis (ICA) (Roberts & Everson, 2001): this operation thus leads to the reduction of the dimension of the feature space.
These features must be as discriminative as possible to reduce false alarms due to misclassification during the second task. Efficient features must also present some essential properties such as:

- translation invariance: whichever be the location of the pattern, it must give exactly the same features,
- rotation invariance: extracted features must not vary with the rotation of the pattern,
- scale invariance: scale changing must not affect the extracted features,
- noise resistance: features must be as robust as possible against noise i.e. they must be the same whichever be the strength of the noise that affects the pattern,
- statistically independent: two features must be statistically independent,
- compact. The number of retained features is not too large. It must also be fast in extraction time and in matching,
- reliable: as long as one deals with the same pattern, the extracted features must remain the same.

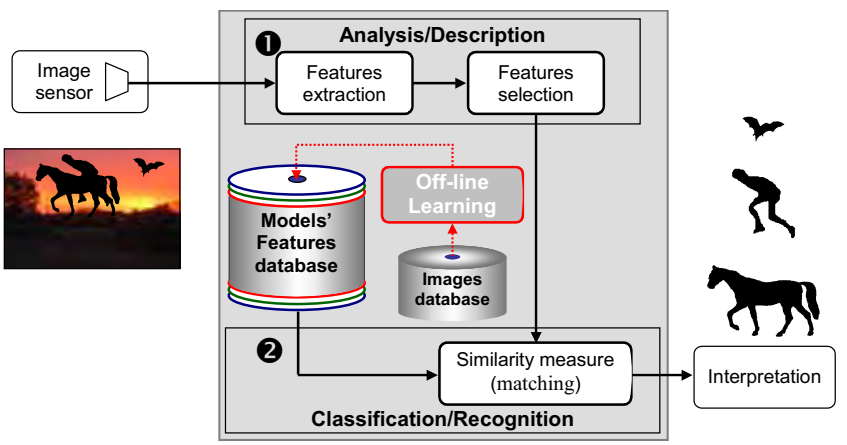


Fig. 4. A generic PRS scheme

During the classification task, the system uses the features extracted in the analysis stage from each of the patterns to compare. As illustrated on figure 4, features are extracted from the patterns of the database during an off-line learning processing. This enables to create features database before each query occurs: by proceeding this way, one doesn't need to compute features of models at each query. To compare two patterns, the system uses a metric that measures a kind of distance (the similarity or the dissimilarity) to assess how similar are two patterns: it is an expression of the distance between the points representing the two patterns in the features space. This procedure gives the similarity index or similarity score between two patterns. In some cases (probably the most natural way), the similarity index is given in terms of a rate varying from 0% for totally different patterns to 100% for perfectly similar patterns (Kpalma & Ronsin, 2006). Some commonly used metrics are Minkowski distance, cosine distance, Hausdorff distance, Mahalanobis Distance (Veltkamp

& Hagedoorn, 2001 ; Zhang, 2002) or city block distance and Euclidian distance that are particular Minkowski distances. The following paragraph illustrates formalism of some of them.

Let $V_A(a_1, a_2,\ldots, a_N)$ and $V_B(b_1, b_2,\ldots, b_N)$ be the features vectors representing patterns A and B in an N-dimensional features space ; examples of distances are defined by the following expressions.

City block distance (d_1)

$$d_1(V_A, V_B) = \sum_{i=1}^{N} |a_i - b_i| \tag{2}$$

Euclidian distance (d_2)

$$d_2(V_A, V_B) = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2} \tag{3}$$

Cosine distance (d_3)

$$d_2(V_A, V_B) = 1 - \cos(\theta) = 1 - \frac{V_A V_B^T}{\|V_A\| \times \|V_B\|} = 1 - \frac{\sum_{i=1}^{N} a_i \times b_i}{\sqrt{\sum_{i=1}^{N} (a_i)^2} \times \sqrt{\sum_{i=1}^{N} (b_i)^2}} \tag{4}$$

where θ is the angle between the two vectors V_A and V_B.

Figure 5 shows an example of three vectors V, U and W represented in 2D space. As it can be seen on this example the value of the similarity/dissimilarity depends on the used distance (metric). In the tables on this figure, d_3 gives the same distance between U and W, on one hand, and between V and W, on the other hand, ($d_3(U,W)= d_3(V,W)=0.15$) but it gives 0 distance between U and V. This leads to confusions, because a distance of 0 that also means vectors equality, may lead to the decision that the patterns to be compared are the same. A particular attention must be paid while choosing a distance. In (Kpalma & Ronsin, 2006) we have proposed a cosine-based distance that enables to remove the ambiguity of the distance between collinear vectors. Since the obtained distance varies from a metric to another, one must be very careful and be sure to use the same metric during all the procedure.

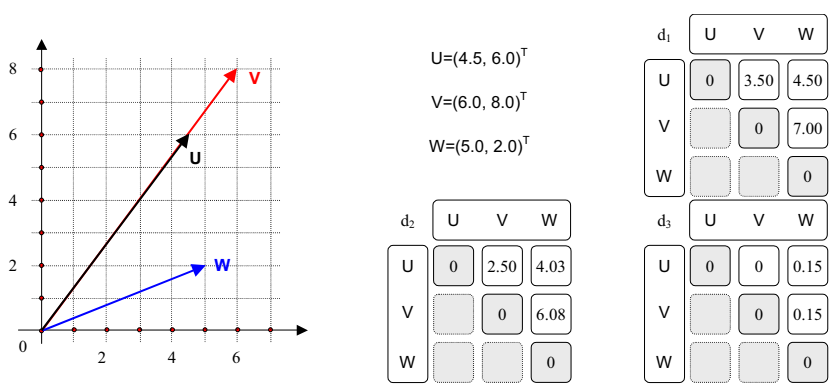


Fig. 5. Examples of similarity measures between two vectors depending on the chosen metric

3. Pattern recognition applications and an overview of advances

Pattern recognition is studied in many fields, including psychology, ethnology, forensics, marketing, artificial intelligence, remote sensing, agriculture, computer science, data mining, document classification, multimedia, biometrics, surveillance, medical imaging, bioinformatics and internet search. Pattern recognition helps to resolve various problems such as: optical character recognition (OCR), zip-code recognition, bank check recognition, industrial parts inspection, speech recognition, document recognition, face recognition, gait recognition or gesture recognition, fingerprint recognition, image indexing or retrieval, image segmentation (by pixels classification)...

In (Pal & Pal, 2002) number of experts address the problem of pattern recognition and present basic concepts involved. One can find the evolution of pattern recognition ; this enables the reader to establish a categorisation of the existing PRSs according to the used methodology and the application.

In (Kuncheva, 2004), the author addresses the non-trivial concept of forgetting in the challenging field of machine learning in non-stationary changing environments. This point of view is essential in on-line diagnosis when using medical imaging: indeed while dealing with PR in real world, the pattern being studied is subject to variation with respect to time. A possible solution is to continuously update the classifier. By doing so, the classifier must be able to "forget" the outdated knowledge. The idea behind this concept is to design an adaptive training system that is able to self-adapt itself accordingly to the changing of the pattern being studied.

Pattern recognition is also applied in more complex fields like data mining (DM) also called knowledge-discovery in databases (KDD). This emerging topic includes the process of automatically searching large volumes of data for patterns such as association rules. As defined in (Frawley et al., 1992), the DM "is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Given a set of facts (data) F, a language L, and some measure of certainty C, we define a pattern as a statement S in L that describes relationships among a subset F_S of F with a certainty c, such that S is simpler (in some sense) than the enumeration of all facts in F_S. A pattern that is interesting (according to a user-imposed interest measure) and certain enough (again according to the user's criteria) is called knowledge. The output of a program that monitors the set of facts in a database and produces patterns in this sense is discovered knowledge".

3.1 Pattern recognition in robotics

The applications of PRSs in robotics are permanent. More recently, Mario E. Munich and his co-authors (Munich et al., 2006) have presented a summary on this subject. In this paper, they show that recent advances in computer vision have given rise to a robust and invariant visual pattern recognition technology based on extracting a set of characteristic features from an image. With visual pattern recognition systems, a robot may acquire the ability to explore its environment without user intervention ; it may be able to build a reliable map of the environment and localize itself in the map: this will help the robot achieve full autonomy. Examples of robots using visual pattern recognition approaches are the Sony's AIBO ERS-7, Yaskawa's SmartPal, and Phillips' iCat.

In robotics, visual servoing or visual tracking is of high interest. For example visual tracking allows, robots to extract themselves the content of the observed scene as a human observer can do it by changing his different perspectives and scales of observation. François

Chaumette (Chaumette, 1994), has addressed the problem and proposed some solutions in a closed loop system based on vision-based task. In (Chaumette, 2004), he proposes various visual features based on the image moments to characterise planar objects in visual-servoing schemes.

3.2 Pattern recognition in biometrics

The biometric authentication takes increasing place in various applications ranging from personal applications like access control to governmental applications like biometric passport and fight against terrorism. In this applications domain, one measures and analyses human physical (or physiological or biometric) and behavioural characteristics for authentication (or recognition) purposes. Examples of biometric characteristics include fingerprints, eye retinas and irises, facial patterns and hand geometry measurement, DNA (Deoxyribonucleic acid). Examples of biometric behavioural characteristics include signature, gait and typing patterns. This helps to identify individual people in forensics applications.

Reference (Jain et al., 2004a) is an interesting starting point to pattern recognition approaches and systems in biometrics. This paper gives a brief overview of the field of biometrics and summarizes some of its advantages, disadvantages, strengths, limitations, and related privacy concerns. In (Jain et al., 2004b), the authors also address the problem of the accuracy of the authentication and that of the individual's right to the security, to the privacy and to the anonymity.

The reader is encouraged to have a look on the article presented in (Jain & Pankanti, 2006). The authors of this article address a problem of identity steeling through a true story and then they present some current or forthcoming systems based on biometric PRSs that will help prevent identity steeling.

3.3 Content-based image retrieval

Content-based image retrieval systems aim at automatically describing images by using their own content: the colour, the texture and the shape or their combination. As explained in (Sikora, 2001; Bober, 2001), image retrieval has became an active research and development domain since the early 1970s. During the last decade the research on image retrieval became of high importance. The most frequent and common means for image retrieval is to index them with text keywords. If this technique seems to be simple, it becomes rapidly laborious and fastidious while facing large volumes of images. On the other hand, images are rich in content so, to overcome difficulties due to the huge data volume, the content-based image retrieval emerged as a promising mean for retrieving images and browsing large images databases

With the simultaneous rapid growth of computer systems and the growing huge availability of digital data, such pattern recognition systems become increasingly necessary to help browse databases and find the desired information within a reasonable time limit. Accordingly to this observation, systems like CBIR (Content-Based Image Retrieval), QBIC (Query By Image Content), QBE (Query By Example) need more attention and take more and more place in the concerns of the researchers (Mokhtarian et al., 1996 ; Trimeche et al., 2000 ; Veltkamp & Tanase, 2001 ; Veltkamp & Hagedoorn, 2001). With query by example, the user supplies a query image and the PRS finds images of the database that are most

similar to it based on various low-level features like colour, texture or shape. With query by sketch, the user draws roughly the image he is looking for and the PRS locates images of the database that match the best the sketch. In the reference (Veltkamp & Tanase, 2001), are reported various CBIR systems. After a brief description of CBIR system, the authors present different kinds of existing systems along with the features involved.
In the context of image indexing, CBIR systems use content information as summarised in figure 6. An image can then be described by using features derived from colour, texture, shape or a combination of those features.

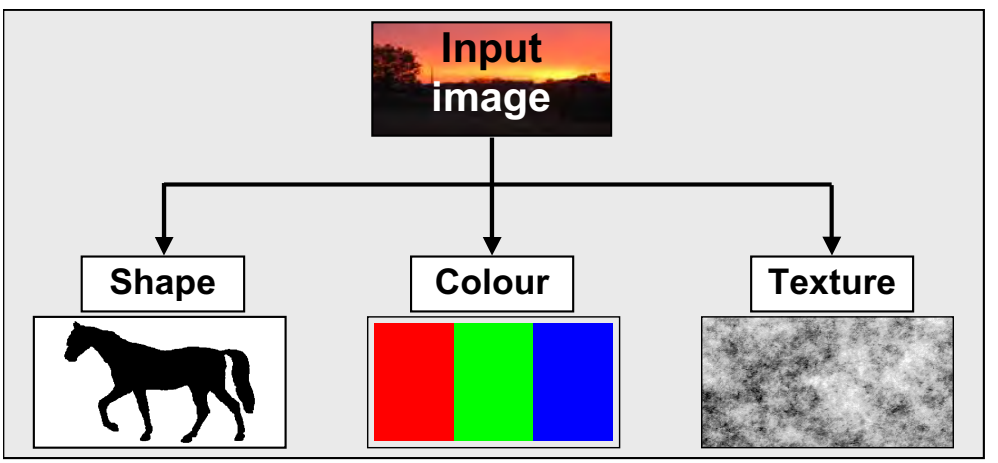


Fig. 6. Content-based image description features

3.3.1 Colour-based features

Colour features are based on colour distribution inside the image. There are many approaches to define colour-based features: dominant colour, colour histogram or colour space. Various colour representation space exist: red-green-blue (RGB) space, hue-saturation-value (HSV) space or those based on the international commission on illumination (or CIE: *commission internationale de l'éclairage*) CIELUV space, CIELAB space, CIEXYZ space. From these representations, features are defined based on the colour histograms. There are different types of colour histograms depending on how the colour space is partitioned. The fixed binning for all images based on scalar linear quantisation, the adaptive binning based on an adaptive quantisation and the clustered binning based on the concept of vector quantisation. Some particular distances between histograms or main modes of histograms are used to measure the similarity/dissimilarity between colour histograms: Euclidian distance, histogram quadratic distance, histogram intersection distance (Smith & Chang, 1996), Jeffrey divergence, Kullback-Leibler divergence earth mover's distance. In the current description of the colour within MPEg-7, the following colour spaces are supported: RGB, YCrCb, HSV, hue-min-max-difference (HMMD), Linear transformation matrix with reference to RGB and monochrome (Martinez, 2004).

3.3.2 Texture-based features

For each pixel of the image, one can determine the histogram of grey levels in predefined neighbouring region centred on that pixel. Distribution of pairs of grey levels for a given spatial relation on pixels can be observed in co-occurrence matrix M(i,j) (Haralick, 1973). Examples of various grey level co-occurrence matrices (GLCM) features defined by Haralick are based on these co-occurrence matrices. In Table 1, by considering a textured image with grey levels ranging from 0 to L-1, we present some of these texture features.

Angular Second Moment	$ASM = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} M(i,j)^2$
Contrast	$C = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 M(i,j)$
Inverse Difference Moment	$IDM = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{M(i,j)}{1+(i-j)^2}$
Homogeneity	$H = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{M(i,j)}{1+\lvert i-j \rvert}$
Entropy	$E = -\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} M(i,j) \ln(M(i,j))$

Table 1. Example of textures features

3.3.3 Shape-based features

There are many approaches (Coster & Chermant, 1985 ; Kpalma, 1994 ; Sossa, 2000), to estimates some properties of the shapes. We present, below, some samples of these properties. Figure 7 shows various shapes and the corresponding measures of their properties.

The **elongation** (EL) indicates how long is the pattern relatively to its width. It is defined by the following expression:

$$EL = 100 \frac{\lambda_m}{\lambda_M} \tag{5}$$

λ_m and λ_M being, respectively, the smallest and the largest eigenvalues of the inertia matrix of the shape. Also called elongation factor or elongation coefficient, this parameter varies from 0% for long but thick shapes to 100% for isotropic shape (see Fig. 7.e and Fig. 7.f).
The **compactness** (CO) measures how branchy or how tortuous is the shape. For a given 2D shape, let A be the enclosed area and P the perimeter ; the compactness is defined by:

$$CO = 100 \frac{4\pi A}{P^2} \tag{6}$$

The compactness varies from 0% for very branchy or very tortuous shapes to 100% for compact shapes like a circle (see Fig. 7.a and Fig. 7.c).
The **mass deficit coefficient** (MD) measures the area variation between the shape and the minimum enclosing circle centred on the centre of gravity of the shape. For a shape with area A, let S_C be the area of the circumscribed circle, then the mass deficit area is defined as follows:

$$MD = 100\frac{S_C - A}{S_C} \tag{7}$$

The **mass excess coefficient** (ME) measures the area variation between the shape and the maximum enclosed circle centred on the centre of gravity of the shape. For a shape with area A, let S_I be the area of the inscribed circle, then the mass deficit area is defined as follows:

$$ME = 100\frac{A - S_I}{A} \tag{8}$$

The two previous parameters, give another estimation of the compactness: they vary from 0% for compact shapes (e.g. a circle) to 100% for spread out tortuous patterns (see Fig. 7.a and Fig. 7.d)
The **isotropic factor** (IF) tells how isotropic is the pattern: it indicates how regular is the shape around its centre of gravity. For a given 2D shape, let R_m be its minimal radius and R_M its maximal radius then the IF parameter is defined by:

$$IF = 100\frac{R_m}{R_M} \tag{9}$$

The isotropic factor varies from 0% for anisotropic shapes to 100% for isotropic shapes like a circle (see Fig. 7.a and Fig. 7.d).

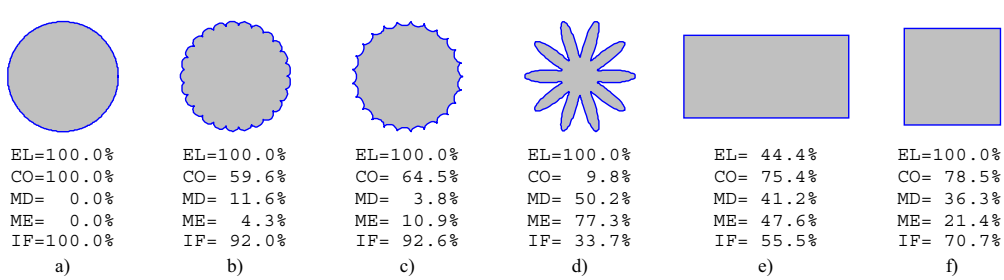


Fig. 7. Various shapes and examples of shape-based features

In the context of shape description, D. Zhang summarized very well the situation (Zhang & Lu, 2004). Figure 8 shows the flowchart of shape description approaches in a pattern recognition system. Typically, there are two kinds of approaches in shape description: the contour-based approach and the region-based one.

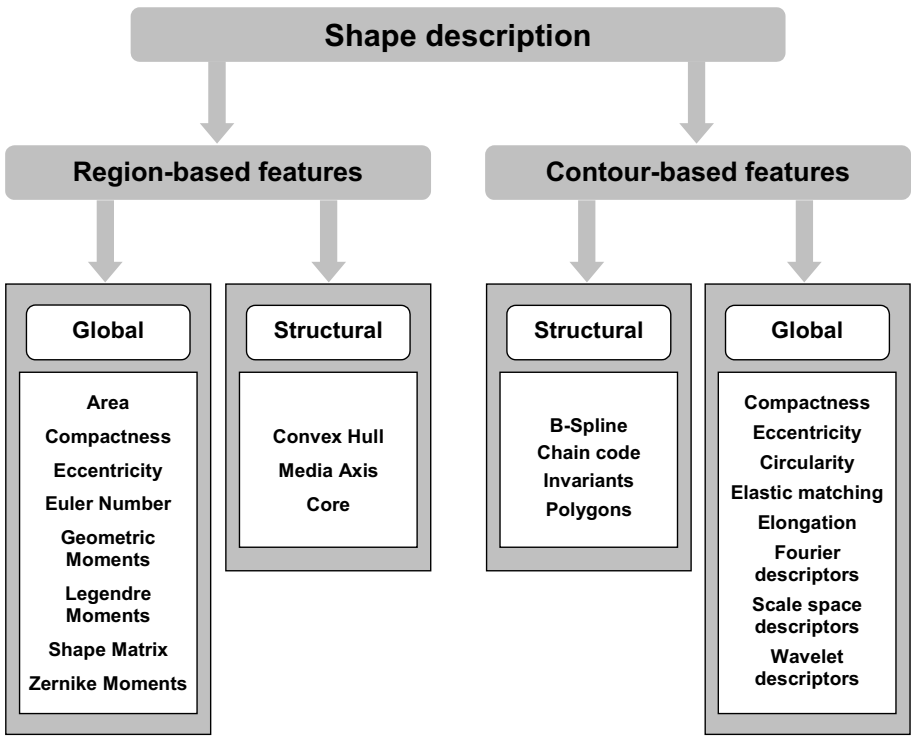


Fig. 8. A classification of shape description approaches

Contour-based approach

Contour-based approaches extract shape features from the only contour in two possible ways: structural or global. In the structural approach, the contour is divided into sub-sections to generate strings or trees according to a particular syntax. The similarity between two shapes is then measured by matching their strings or their trees.

While dealing with the contour in the global way, an appropriate technique is used to extract primitive features from the integral contour: eccentricity, perimeter, circularity… From these basic features, one defines a multidimensional vector representing the shape in the features space. From this representation, the similarity measure or the matching of two shapes is done by directly measuring a specific distance between their feature vectors.

For contour-based shape description, MPEG-7 working group (Bober, 2001 ; Martinez, 2004) has selected the so-called Curvature Scale-Space (CSS) representation which is proved to capture perceptually meaningful features of the shape (Mokhtarian et al., 1996 ; Matusiak & Daoudi, 1998 ; Lindenberg, 1998 ; Mokhtarian & Bober, 2003).

A CSS image, represented on figure 9, is a multi-scale organization of the invariant local features of a 2-D contour: it consists of the curvature zero-crossing points recovered from the contour at multiple scales of resolution. The features extracted from the CSS image consist of the coordinates of the peaks of the CSS image. Scale decreasing is obtained through progressive low-pass filtering by convolutions of a parametric representation of the

contour data with Gaussian filters of increasing width. This representation carries a number of important properties, such as:

- it captures very well characteristic features of the shape, enabling similarity-based retrieval,
- it reflects properties of the perception of human visual system and offers good generalization,
- it is robust to non-rigid motion,
- it is robust to partial occlusion of the shape,
- it is robust to perspective transformations, which result from the changes of the camera parameters and are common in images and video,
- it is compact.

Some of the above properties of this descriptor are illustrated in figure 11, each frame containing very similar images according to CSS, based on the actual retrieval results from the MPEG-7 shape database. In figure 9, we represent two shapes and their corresponding CSS images. On the CSS images (bottom row) we have superposed the peaks points that are used to generate features (Mokhtarian & Bober, 2003).

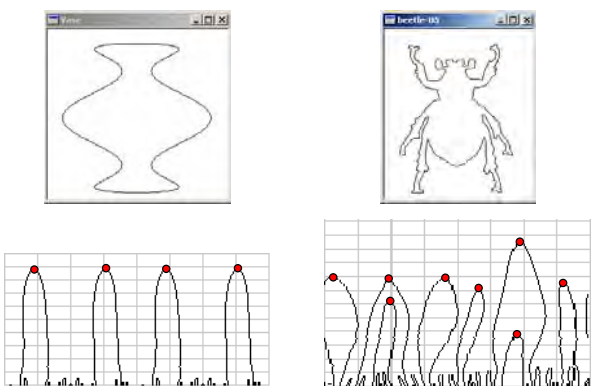


Fig. 9. Example of contours (top row) and the corresponding CSS image with the peaks points (bottom row)

Region-based approach

In region-based approaches, all pixels surrounded by the shape boundary are taken into account to generate the shape descriptor. Like in the case of contour-based approaches, we encounter the same two different ways in region-based shape description: global and structural one. In the structural approach, the shape is decomposed into sub-regions to generate a tree to represent the shape. In the global way, one computes some characteristic features to generate a vector to represent the shape. Common global features derived from a region-based approach are: geometrical moment invariants, shape matrix, area, compactness, eccentricity, Euler number, geometric moments, Legendre moments, Zernike moments… For region-based shape description, the MPEG-7 working group (Bober, 2001 ; Martinez, 2004) has selected the angular radial transform (ART). It is a moment-based approach for a 2D region-based shape description. In (Ricard et al., 2005) the authors proposed a generalization of the ART approach to describe 2D and 3D shapes for content-based image retrieval purpose.

The contour-based approaches are more appealing than region-based approaches because they involve less computation complexity, than the region-based ones, with enough discriminating efficiency. It is also demonstrated that characteristic information about a shape lie essentially on its contour features. The main drawback of contour-based descriptors is that they are more subject to noise and variations than region-based ones.
Figure 10 shows examples of shapes and illustrates situations for which the contour-based or the region-based descriptors are most suitable.
A shape may consist of just one single region (see Fig.10.a-c) or a set of several regions as well as regions with some holes inside them as illustrated in figures 10.d-f. Since the region-based descriptors make use of all pixels constituting the shape, they can describe any kind of shapes. They are more suitable than the contour-based descriptors to handle complex shape consisting of holes in the object or several disjoint regions (see Fig.10.d-f) in a single descriptor. Indeed, for contour-based descriptors, these shapes consist not of a single contour but of multiple contours leading, thus, to multiple descriptors.

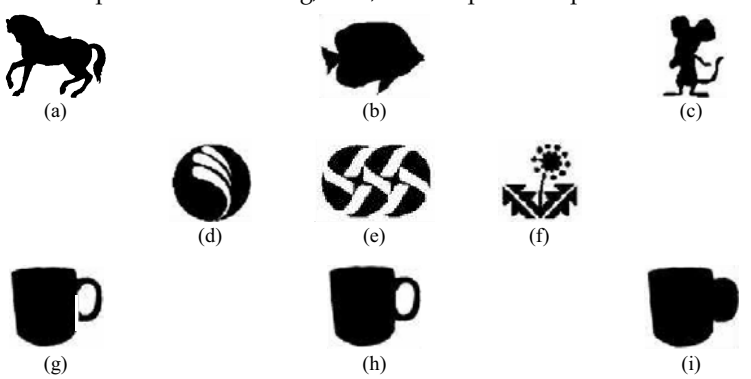


Fig. 10. Examples of various shapes

Figures 10.g-i show very similar shapes from images of a same cup. They only differ by the handle: shape 10.g has a crack at the lower handle while the handle in 10.i is filled. When comparing these shapes:

- the region-based shape descriptor will consider 10.g and 10.h similar but different from 10.i,
- the contour-based shape descriptor will consider 10.h and 10.i similar but different from 10.g.

As illustrated by MPEG-7 (Martinez, 2004), a challenge for a pattern descriptor is to enable the recognition of a pattern even if it has undergone various deformations namely partial occultation (Fig.11.a) and non-rigid deformation (Fig. 11.b).
Figure 11.a, according to (Martinez, 2004), illustrates the robustness to partial occultation: indeed, in this figure, one can note that the tails or the legs of the horses are sometimes occulted but they are recognised to be from the same class. As presented in (Mokhtarian, 1997 ; Petrakis, 2002) , this is possible because of the ability of the descriptor to handle local properties. On figure 11.c are represented various shapes that are classified in the same class based on the visual perceptual similarity

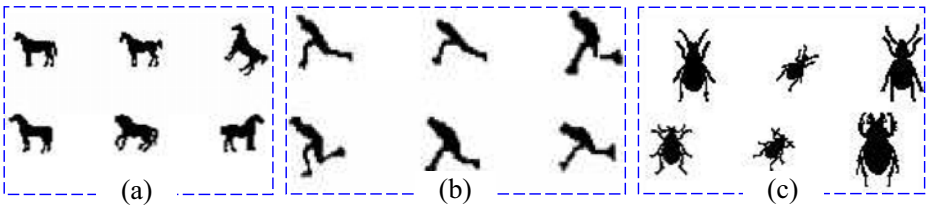


Fig. 11. a) robustness to partial occlusion, b) robustness to non-rigid deformation, c) perceptual similarity among different shapes

The choice of a description method will depend on the application so, sometimes, one needs to make a compromise. Nevertheless, MPEG-7 has set some essential principles to evaluate the suitability of shape descriptor: retrieval accuracy, compactness, generics, low computation complexity, robustness and the ability to represent a shape in hierarchical way: from coarse to fine representation.

3.4 An overview of the advances in pattern recognition

Remco C. Veltkamp and Mirela Tanase presented in (Veltkamp & Tanase, 2000) a large panel of CBIR systems. Various approaches of the state of the art in content-based image retrieval and video retrieval are explored along with the features used in each approach, they also describe the matching functions used. This overview enables to confirm, as it was said before, that commonly designed CBIR systems are generally based on visual features such as colour, texture and shape.

In (Iqbal & Aggarwal, 2002) is presented CIRES (Content-based Image REtrieval System), an online system for retrieval in image libraries. It is done to extend the retrieval paradigm, which was mostly limited to colour and texture analyses, by using image structure. Image structure was extracted via hierarchical perceptual grouping principles.

In (Mittal, 2006) the author presents an overview of the content-based retrieval along with different strategies in terms of syntactic and semantic indexing for retrieval. After an analysis of the matching techniques used and the learning methods the author addresses some directions for future research in the content-based retrieval domain.

Recently, N. Snavely and co-authors (Snavely et al., 2006) have presented a system that consists of 3D image-based modelling and representation of an unorganised images taken by different cameras in different conditions. The challenging aim of the system is to use the content-based information to browse an image database and reply to questions like:

- **"where was I?** Tell me where I was when I took this picture"
- **"what am I looking at?** Tell me about objects visible in this image by transferring annotations from similar images"

To do this, they used the SIFT (Scale Invariant Feature Transform) keypoints detector that was shown to be transformation invariant (Lowe, 2004).

Among the various forthcoming systems, we can encounter MPEG-7. Formally named "Multimedia Content Description Interface", MPEG-7 aims at managing data in the way that content information can be retrieved easily. It is under development by the Moving Picture

Coding Experts Group (MPEG) that is a working group of ISO/IEC(*) standards organization. It is in charge of the development of international standards for video and/or audio compression, decompression, processing and representation. This group has also developed well-known standards that are MPEG-1, MPEG-2 and MPEG-4. MPEG-1, MPEG-2 and MPEG-4 also make content available but MPEG-7 enables to find the desired content. MPEG-7 visual description tools consist of basic structures and descriptors that cover basic visual features: colour, texture, shape, motion, localization. Each category consists of elementary and sophisticated descriptors (Sikora, 2001; Bober, 2001). One must note that MPEG-7 addresses many different applications in various environments, thus it needs to provide a standard flexible and extensible framework for describing audio-visual data.

4. Application example based on the MSGPR method

In (Kpalma & Ronsin, 2006) we have presented an original pattern description approach based on the multi-scale analysis of the contour of planar objects. This proposed approach summarises the different presented considerations in this chapter. It is well known that some objects, especially natural ones, exist with a more or less large range of scales; and that the aspect of the object can change from one scale to another. Without a priori information about the distance of observation inside a given scene, an interesting challenge can be to find an object without any precision about its scale of observation. Faced with this situation, it is very difficult to significantly describe a pattern using only one meaningful scale. To overcome this problem, increasingly more pattern description techniques are based on multi-scale or multiresolution representation methods (Lindeberg, 1998). Within this context, methods based on the pattern itself (Torres-Méndez et al., 2000 ; Kadyrov & Petrou, 2001 ; Belongie et al., 2002 ; Grigorescu & Petkov, 2003) exist as well as methods based on pattern contour behaviour (Matusiak & Daoudi, 1998 ; Roh & Kweon, 1998 ; Wang et al., 1999 ; Latecki et al., 2000).
This study deals exclusively with methods based on the pattern contour. Called MSGPR (A Multi-Scale curve smoothing for Generalised Pattern Recognition) this scale-space (Mokhtarian et al., 1996 ; Matusiak & Daoudi, 1998 ; Wang et al., 1999 ; Mokhtarian & Bober, 2003) method is based on multi-scale smoothing of a planar pattern contour. This method is totally translation and rotation insensitive and as showed in the initial studies it is also robust against scale change for a large range of scaling and resistant to additive noise.

4.1 Description of the MSGPR method

The framework of the MSGPR can be broken down into four main stages as follows (see Fig.12):

1. the input contour is separated into two parameterised functions,
2. both functions are low-pass filtered (smoothed),
3. scale adjustment is then applied to both filtered functions so that the corresponding smoothed contour has the same scale as the input one,

(*)ISO/IEC stands for International Standards Organization/International Electro-technical Committee.

4. finally, the intersection points map (IPM) is generated by detecting the intersection points of the input contour and the smoothed scale-adjusted one.

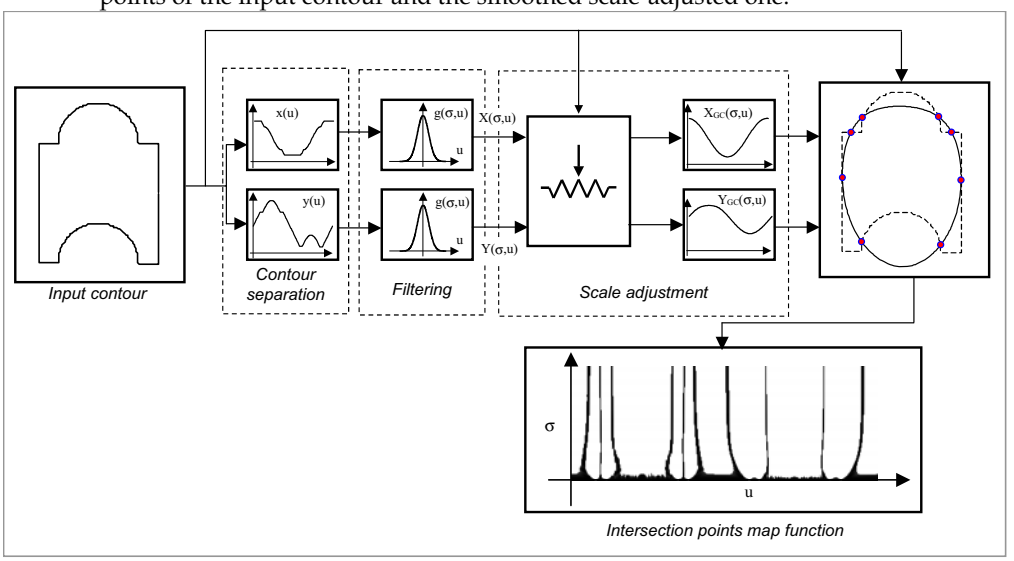


Fig. 12. MSGPR description scheme

4.1.1 Coordinate separation

The input contour is represented by a series of points defined by their (x,y) coordinates. First, the input contour is separated into two functions x(u) and y(u) which are functions of the normalised curvilinear u parameter that varies from 0 to 2π relative to the curve length. Each point of the curve is then represented by its parameterised coordinates $(x(u), y(u))$.

4.1.2 Curve smoothing

Functions x(u) and y(u) are then gradually smoothed by decreasing the filter bandwidth. Similarly to the curvature scale space (CSS) method (Mokhtarian et al., 1996 ; Matusiak & Daoudi, 1998 ; Wang et al., 1999 ; Mokhtarian & Bober, 2003) or other scale-space methods, smoothing is based on the Gaussian filters g(σ,u) with standard deviation σ:

$$g(\sigma, u) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{u^2}{2\sigma^2}} \tag{10}$$

The filtered functions are then given by: $X(\sigma,u) = g(\sigma,u) * x(u)$ and $Y(\sigma,u) = g(\sigma,u) * y(u)$ so that each $(x(u), y(u))$ point on the input contour leads to the $(X(\sigma,u), Y(\sigma,u))$ point on the output smoothed contour.

Since the bandwidth is conversely proportional to σ, it is clear that the bandwidth decreases as σ increases. Thus the filter cuts increasingly lower so that the output functions move towards their mean values when σ tends towards infinity.

4.1.3 Scale adjustment

After low-pass filtering, the scale adjustment system stretches the output contour so that it reaches the same scale as the input one and so that both contours intersect at certain points. Figure 13 shows an example of a contour and two smoothed ones (σ=30 and σ=180) after they have been scale-adjusted. The input contour C_0 and smoothed scale-adjusted contours $C_{GC}(\sigma)$ are now on the same scale so that they can intersect.

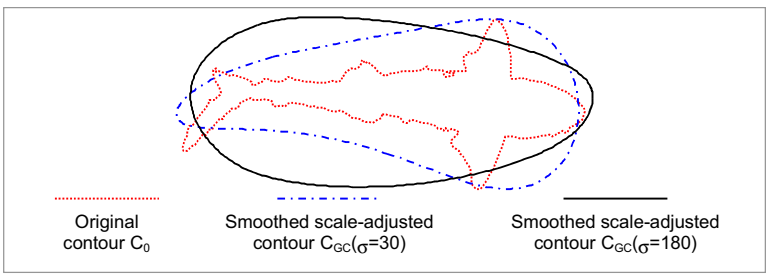


Fig. 13. Example of a contour and two smoothed scale-adjusted ones (σ=30 and σ=180)

4.1.4 Definition of the IPM function

By increasing σ, the output contour moves towards a convex curve that has some intersection points with the input contour. By marking these intersection points for each σ, we obtain the intersection points map (IPM) function defined below which characterises the pattern.

After the scale adjustment system, the IPM function is generated as follows. For each σ value, we define a function which is an image in the scale-space (u,σ) plane so that (see Fig.14):

- $\mathrm{IPM}(u,\sigma)=0$ (black) if the $(x(u),y(u))$ point is an intersection point between the original curve and the filtered scale-adjusted one,
- $\mathrm{IPM}(u,\sigma)=1$ (white) if point $(x(u),y(u))$ is not an intersection point.

Figure 14 shows examples of contours (left column) and the corresponding IPM functions (right column). On this figure, intersection points are indicated by (1) through (6) or (8), for the contour in Fig.14.a or for that in Fig.14.c, respectively. On the right column, one can see the marks corresponding to those intersection points in the IPM representation. As can be seen on this figure, the IPM function is characteristic of the contour it is derived from.

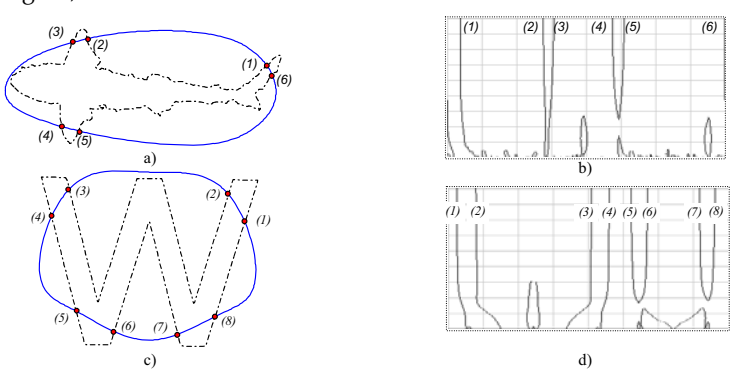


Fig. 14. Example of the IPM function

4.1.5 Features definition and selection

After generating the IPM function, the following stage, but not the least one is features definition and selection. In (Kpalma & Ronsin, 2006) we used the circular distance between IPM points at various scale values. To extract these characteristic features, we first set the scale parameter σ to σ_0 value (e.g. $\sigma_0 = 180$). Then, for each pattern:

- we consider the IPM points at the set σ_0 and select two consecutive p_a and p_b points which are, circularly, the furthest apart in the IPM function as illustrated in figure 15,
- we determine the circular distance between both points to produce the first d_1 component of the V_0 features vector,
- the next components of V_0 are distances coming after d_1:

$$V = (V_0, V_1, \ldots, V_{M-1}) \tag{11}$$

To benefit from multi-scale information of the IPM function, we can define a set of M values of σ ($\sigma_0, \sigma_1, \ldots, \sigma_{M-1}$) and determine the V_i feature vectors (i=0, 1, 2,…, M-1) corresponding to the σ_i scales. The global V features vector is then produced by a concatenation of the individual V_i scale vectors as follows:

$$V = (V_0, V_1, \ldots, V_{M-1}) \tag{12}$$

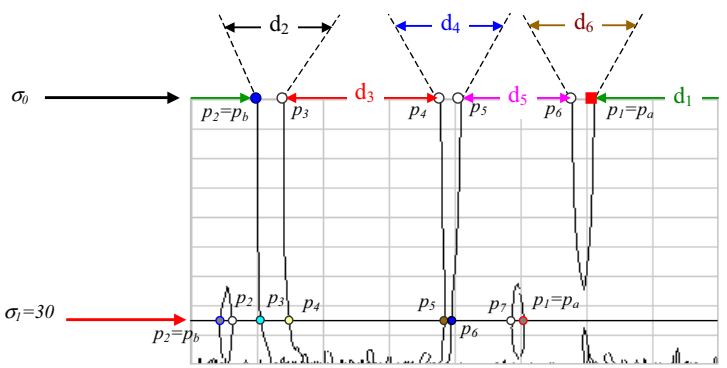


Fig. 15. Example of the IPM function

4.1.6 Similarity measure

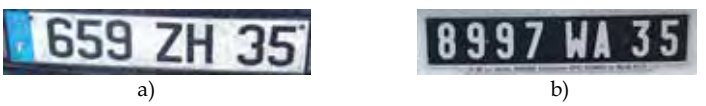
To measure the matching rate between two V_A and V_B attribute vectors associated to two patterns, we define a similarity function as follows:

$$\text{SimScore}(V_A, V_B) = 50(1+\cos(\gamma))\frac{\text{Min}(\|V_A\|,\|V_B\|)}{\text{Max}(\|V_A\|,\|V_B\|)} \tag{13}$$

where γ is the angle between both vectors and where $\|.\|$ indicates the module of a vector. This function ranges from 0% for very different vectors to 100% for perfectly matching vectors.

4.2 Application to car plate character recognition

In this section, we present a system we have developed to illustrate pattern recognition systems. This application can be classified into the group of the contour-based statistical approaches. Our application illustrates an automatic reading of the number plates by using their digital images. Applying the IPM-based features we carry out the automatic recognition of the characters of the number plate. Figure 16 shows two images of plates written with different fonts: the difference appears more clearly for digit '**3**' out of the two plates.



a) b)

Fig. 16. Examples of number plates images

4.2.1 Character recognition procedure

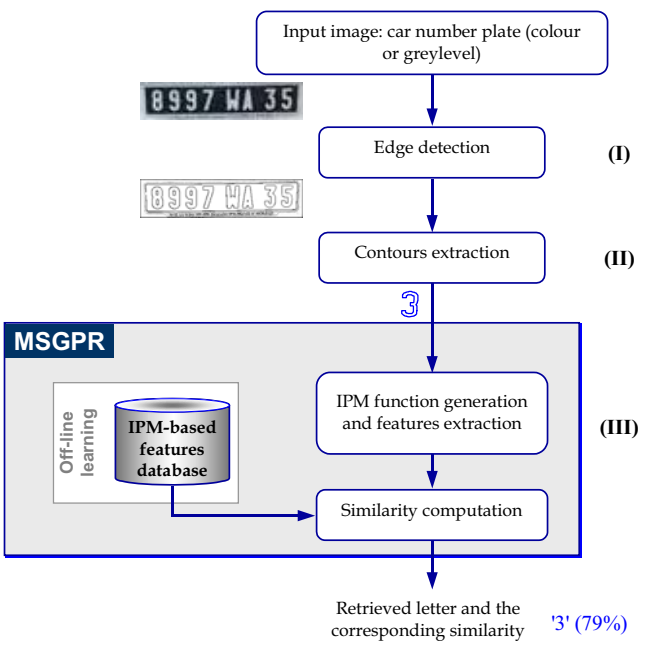


Fig. 16. An overview of an automatic number plate reading system.

The recognition procedure is carried out into three stages as depicted in figure 17:

(I) **edge (or contours) detection** that will enable to obtain contours delimiting each character in the image (Fig. 18.a et Fig. 18.b). One must note that this stage is very important in our process, because, the effectiveness of character recognition will depend on it.

(II) **contour extraction:** in this stage, one considers only the external (or the outer) boundary (Fig. 18.c), because only these contours are taken into account. As for the stage (I), one must pay particular care to the extraction of the characters so that they are continuous and closed, without self-intersection.

(III) **character recognition:** at this last stage, we apply our IPM-based description approach to extract the features and to integrate them into the identification process to measure the similarity score between each extracted character and the models of the data base. In this application, the similarity measure is based on the *SimScore* function defined by equation (13).

4.4.2 Experimental results

Figures 18.a and 18.b represent the output images of the edge detection when applied to images corresponding to figure 16. The figure 18.c presents the set of the extracted characters from figures 18.a and 18.b. On figure 18.d we present a sample set of characters of the database: this base consists of the character set "bold.chr" of Borland®.

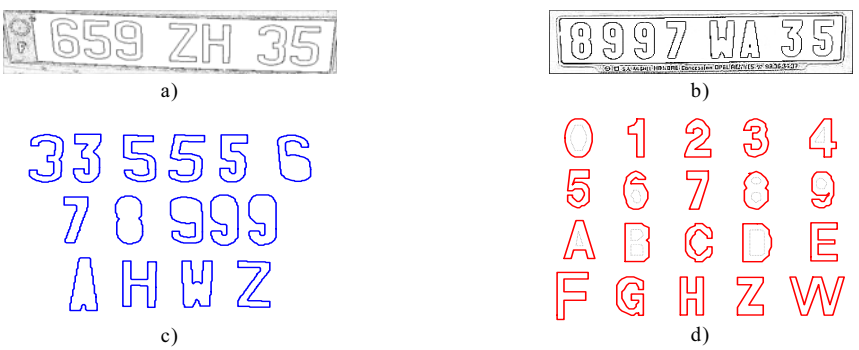


Fig. 18. a) and b) detected edges - c) extracted contours from a) and b) - d) examples of the content of the database.

It must be noted that in this study, the database is composed of only one font while the query characters come from two different fonts. In order to improve the identification results, a possible solution would be to integrate in the database, all the possible fonts used to create car plates. Figures 19 show some results obtained from the input images presented on figure 16. On these figures, we represent some results of character recognition: on each figure, the contour on the upper left corner represents the query contour. Following contours in left-to-right and top-to-down scanning, represent eight retrieved contours which give the highest similarity scores.

As can be seen on these figures, the identification of different characters is effective enough: for each query, the identified character (the most similar: the character next to the query in figures 19.a-d) is exactly the required character. Thus, for the query '**3**', we identify the letter '**3**' with a similarity score of 79%. Table 2 summarises the three highest similarity scores for the contours presented on figure 19. For the contour '**9**' as a query, we retrieved the digit '**9**' with a similarity score up to 96% followed by the digit '**6**' with a similarity score of 79%. One can notice that the contour '**6**' of the used font is not other than the contour '**9**' which underwent a rotation of 180°: this explains that the digit '**6**' occupies the second position

during the retrieval process. In the same way, the topological similarity between the digit '**5**' and the letter '**S**' or between the digit '**8**' and the letter '**B**' results in the appearance of '**S**' and '**B**', respectively, into the second position in the retrieval ranging. In spite of this topological similarity, specific properties of each character lead to sufficiently important variations of similarity scores to avoid mistakes.

a) b) c) d)

Fig. 19. Examples of the recognition output

Query	Retrieved character (Similarity score)		
'3'	'3' (79%)	'C' (62%)	'E' (56%)
'5'	'5' (72%)	'S' (58%)	'6' (55%)
'8'	'8' (91%)	'B' (63%)	'1' (61%)
'9'	'9' (96%)	'6' (79%)	'K' (76%)

Table 2. Retrieved characters and the corresponding similarity scores.

5. Conclusion

As mentioned before, pattern recognition does not appear as a new problem. A lot of studies have been performed on this scientific field and a lot of works are currently developed. Pattern recognition is a wide topic in machine learning. It aims to classify a pattern into one of a number of classes. It appears in various fields like psychology, agriculture, computer vision, robotics , biometrics… With technological improvements and growing performances of computer science, its application field has no real limitation. In this context, a challenge consists of finding some suitable description features since commonly, the pattern to be classified must be represented by a set of features characterising it. These features must have discriminative properties: efficient features must be affined transformations insensitive. They must be robust against noise and against elastic deformations due, e.g., to movement in pictures.

Through the application example based on our MSGPR method, we have illustrated various aspects of a PRS. With this example, we have illustrated the description task that enabled us to extract multi-scale features from the generated IPM function. By using theses features in the classification task, we identified the letters from a car number plate so that we automatically retrieved the license number of a vehicle.

The research topic of pattern recognition is under continuous development and in perpetual progress. With the large volumes of digital images, the challenge for pattern recognition in computer vision is now the development of a CBIR-like system: system that is able to retrieve useful information by using the only content of the input image. With the growing huge availability of digital images, pattern recognition takes more and more place in our daily life to help us find the desired information in a reasonable time limit, while browsing large databases.

Pattern recognition is integrated into the forthcoming standard MPEG-7 via indexing approaches. Such standardization does not bring restriction to a domain: it gives synergy of best actors mixing challenge and cooperation. And moreover international standardization occurs as a requirement from different applications so it meets all conditions for large diffusion. Standards use the possibilities of last technological developments, and drive strong investments and focus research on the concerned domain. As it has been observed, for example, for coding when it was integrated inside different MPEG standards, the integration of pattern recognition inside MPEG-7 will boost its last developments.

6. References

Abdi, H. (1994). A neural network primer. Journal of Biological Systems, Vol. 2, No. 3, pp. 247-281

Belongie, S., Malik, J., and Puzicha, J., Shape matching and object recognition using shape contexts. IEEE PAMI-24, No 24, pp 509-522, 2002

Bober, M. (2001). MPEG-7 Visual Shape Descriptors, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp 716-718.

Bruckstein, A. M., Rivlin, E., and Weiss, I. (1996). Recognizing objects using scale space local invariants, Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96), August 25-29, pp. 760-764, Vienna, Austria.

Bruckstein, A., Katzir, N., Lindenbaum, M., and Porat, M. (1992). Similarity invariant signatures for partially occluded planar shapes, IJCV, Vol. 7, No. 3, pp. 271-285.

Brunelli, R. and Poggio, T. (1997). Template Matching: Matched Spatial Filters And Beyond, Pattern Recognition, Vol. 30, No 5, pp. 751-768

Chaumette, F. (2004), Image Moments: A General and Useful Set of Features for Visual Servoing, IEEE Transactions on Robotics, Vol. 20, No. 4, pp. 713-723

Chaumette, F. (1994). Visual servoing using image features defined upon geometrical primitives, International 33rd IEEE Conference on Decision and Control, Vol. 4, pp. 3782-3787, Orlando, Florida

Cole, L.; Austin, D. and Cole, L. (2004). Visual Object Recognition using Template Matching, Australasian Conference on Robotics and Automation 2004

Coster, M. and Chermant, J.-L. (1985). Précis d'Analyse d'Images, Editions du CNRS, 15, quai A. France, Paris, 1985

Frawley, W. J.; Piatetsky-Shapiro, G. & Matheus, C. J. (1992). Knowledge Discovery in Databases: An Overview, AI Magazine 13(3), pp. 57-70

Grigorescu, C., and Petkov, N. (2003). Distance Sets for Shape Filters and Shape Recognition. IEEE Trans. Image Processing 12(9).

Haralick, R.M. (1979), Statistical and structural approaches to texture, Proceedings of the IEEE, No. 5, Vol. 67, pp. 786-804

Haralick, R.M., Shanmugam, K. and Dinstein, I. H. (1973). Textural features for image classification, IEEE Transaction on Systems, Man and Cybernitics, Vol. SMC-3, n°6, pp. 610-621

Iqbal, Q. and Aggarwal, J. K. (2002). CIRES: A System for Content-based Retrieval in Digital Image Libraries, Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, pp. 205-210, December 2-5, 2002

Jain, A. K. and Pankanti, S. (2006). A Touch of Money, *IEEE Spectrum*, vol. 43, no. 7, pp. 22-27, July 2006.

Jain, A. K.; Ross, A. and Prabhakar, S. (2004a).An Introduction to Biometric Recognition, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 1, January 2004

Jain, A. K., Pankanti, S., Prabhakar, S., Hong, L., Ross, A., and Wayman, J. L. (2004b). Biometrics: A Grand Challenge, Proceedings of the 17th International Conference on Pattern Recognition, Vol. 11, August 2004, pp. 935–942.

Jain, A. K.; Duin R. P.W. and Mao, J. (2000). Statistical Pattern Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4-37

Kpalma, K., and Ronsin, J. (2006). Multiscale contour description for pattern recognition, Elsevier Science Inc, Pattern Recognition Letters, Vol.27, No.13, pp 1545-1559, 1 October 2006

Kpalma, K., and Ronsin, J. (2003). A Multi-Scale curve smoothing for Generalised Pattern Recognition (MSGPR), Seventh International Symposium on Signal Processing and its Applications (ISSPA), pp 427-430, Paris, France.

Kpalma, K. (1994). Caractérisation de textures par l'anisotropie de la dimension fractale, Proceedings of the 2nd African Conference on Research in Computer Science (CARI), October 1994, Ouagadougou, Burkina Faso.

Kadyrov A., Petrou, M. (2001). Object descriptors invariant to affine distortions. Proceedings of the British Machine Vision Conference, BMVC'2001, Manchester, UK.

Kuncheva, L. I. (2004). Classifier Ensembles for Changing Environments, Proc. 5th International Workshop on Multiple Classifier Systems, Cagliari, Italy, Springer-Verlag, LNCS, Vol. 3077, 1-15

Latecki, L. J., Lakamper, R., and Eckhardt, U. (2000). Shape Descriptors for Non-rigid Shapes with a Single Closed Contour, IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), pp. 424-429, 2000

Lindeberg, T. (1998). Principles for Automatic Scale Selection, Technical report ISRN KTH NA/P--98/14--SE. Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden.

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, Dordrecht, Netherlands.

Liu, J., Sun, J. and Wang, S. (2006). Pattern Recognition: An overview, International Journal of Computer Science and Network Security (IJCSNS), Vol. 6, No.6, June 2006

Lowe, D. G. (2004). Distinctive image features from scale invariant keypoints, IJCV, 60 (2):91–110, 2004.

Martinez, J. M., (editor), (2004), MPEG-7 Overview (version 10), ISO/IEC JTC1/SC29/WG11 N6828, Palma de Mallorca, October 2004

Martinez, J.M. (2002). Standards - MPEG-7 overview of MPEG-7 description tools, part 2., IEEE Multimedia 9 (3), July-Sept. 2002, pp. 83 –93

Matusiak S., Daoudi M. (1998). Planar Closed Contour Representation by Invariant Under a General Affine Transformation, IEEE International Conference on System, Man and Cybernetics (IEEE-SMC'98), pp. 3251-3256, October 11-14, Hyatt Regency La Jolla, San Diego, California, USA.

Mittal A. (2006). An Overview of Multimedia Content-Based Retrieval Strategies, Informatica, International Journal of Computing and Informatics, Vol. 30, No. 3, pp. 347–356

Mokhtarian, F., and Bober, M. (2003). *Curvature Scale Space Representation: Theory, Applications and MPEG-7 Standardizatio*n. Kluwer Academic.

Mokhtarian, F. (1997). Silhouette-Based Occluded Object Recognition through Curvature Scale Space, Machine Vision and Applications, Vol. 10, No. 3, pp. 87-97.

Mokhtarian, F., Abasi, S., and Kittler, J. (1996). Efficient and Robust Retrieval by Shape Content through Curvature Scale Space, in Proceedings International Workshop on Image Databases and MultiMedia Search, pp 35-42, Amsterdam, The Netherlands.

Mokhtarian, F., and Mackworth, A. K. (1992). A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-14, N° 8.

Munich, M. E.; Pirjanian, P.; Di Bernardo, E.; Goncalves, L.; Karlsson, N. and Lowe, D. (2006). Application of Visual Pattern Recognition to Robotics and Automation, IEEE Robotics & Automation Magazine, pp.72-77, September 2006

Pal, S.K. & Pal, A., (Editors). (2002). *Pattern recognition: from classical to modern approaches*, World Scientific, ISBN No. 981-02-4684-6, Singapore

Petrakis, E. G.M.; Diplaros, A. and Milios, A. (2002). Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1501-1516

Ricard, J., Coeurjolly, D. and Baskurt A. (2005). Generalizations of angular radial transform for 2D and 3D shape retrieval, Elsevier Science Inc, Pattern Recognition Letters Volume 26, Issue 14 , 15 October 2005, Pages 2174-2186

Roberts, S. and Everson, R. (2001). *Independent Component Analysis- principles and practice*, Cambridge University Press, ISBN 0521792983

Roh, K.-S., Kweon, I.-S. (1998). 2-D object recognition using invariant contour descriptor and projective refinement, Pattern Recognition, Vol. 31, N° 4, pp. 441-455.

Smith, J. and Chang, S. F. (1996).Tools and Techniques for Color Image Retrieval. in IS&T/SPIE proceedings of Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV vol. 2670, pp. 1630-1639, San Jose, CA, February 1996.

Snavely, N., Seitz, S. M. and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D, ACM Transactions on Graphics (SIGGRAPH Proceedings), 25 (3), pp. 835-846.

Sonka, M.; Hlavac, V. and Boyle, R. (1993). *Image Processing, Analysis and Machine Vision*, Chapman & Hall, London, UK, 1993, pp. 193–242

Sossa, H., 2000. Object Recognition, Summer School on Image and Robotics, INRIA Rhône-Alpes, France.

Sun, K. B. and Super, B. J. (2005). Classification of Contour Shapes Using Class Segment Sets Full text, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2

Torres-Méndez, L. A., Ruiz-Suárez, J. C., Sucar, L. E. and Gómez, G. (2000). Translation, Rotation, and Scale-Invariant Object Recognition, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, Vol. 30, No. 1, pp 125-130.

Trimeche M., Alaya Cheikh F., and Gabbouj, M. (2000). Similarity Retrieval of Occluded Shapes Using Wavelet-Based Shape Feature, Proc. SPIE International Symposium on Internet Multimedia Management Systems (VV10), Boston, Massachusetts, USA.

Vapillon, A.; Collin, B. and Montanvert, A. (1998). Analyzing and Filtering Contour Deformation, International Conference on Image Processing (ICIP), Chicago, Illinois, USA.

Wang Y.-P., Lee, S.L., and Toraichi, K. (1999). Multiscale curvature-based shape representation using B-spline wavelets, IEEE Transactions on Image Processing, Vol. 8, No 11, pp 1586-1592.

Sikora, T. (2001). The MPEG-7 Visual Standard for Content Description—An Overview, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, June 2001

Veltkamp, R C and Tanase M. (2001). Content-based retrieval systems: a survey, Technical Report UU-CS-2000-34, citeseer.ist.psu.edu/veltkamp00contentbased.html

Veltkamp, R. C.; Burkhardt, H. & Kriegel, H.-P. (2001). *State-Of-The-Art in Content-Based Image and Video Retrieval*, ISBN 1-40200-109-6, Kluwer Academic Publishers.

Veltkamp, R. C, & Hagedoorn, M. (2001). *State-of-the-art in shape matching*. In *Principles of Visual Information Retrieval*, M. Lew (editor), Springer, ISBN 1-85233-381-2, pp 87-119.

Venguerov, M. & P. Cunningham, P. (1998). Generalised Syntactic Pattern Recognition as a Unifying Approach in Image Analysis, LNCS, Vol. 1451, pp913-920, Springer Verlag, Sydney, (Australia)

Watanabe, S. (1985). *Pattern recognition: human and mechanical*. Wiley, 1985

Zhang, D., and Lu, G. (2004). Review of shape representation and description techniques, Pattern Recognition, Vol.37, pp 1–19.

Zhang, D. (2002). Image Retrieval Based on Shape, PhD dissertation, Faculty of Information Technology, Monash University, Australia

Liu, J., Sun, J. and Wang, S. (2006). Pattern Recognition: An overview, International Journal of Computer Science and Network Security (IJCSNS), Vol. 6, No.6, June 2006

11

Robust Microarray Image Processing

Eugene Novikov, Emmanuel Barillot
Service Bioinformatique, Institut Curie
26 Rue d'Ulm, 75248 Paris Cedex 05,
France

1. Introduction

High-density microarrays are a rapidly developing technology in molecular biology allowing one to measure simultaneously the activity of thousands of biomolecules in the cell under different experimental conditions. Two-color comparative microarray experiment is a key point of transcriptome (Yang et al., 2002; Herzel et al., 2001; Hegde et al., 2000), CGH (comparative genome hybridization, Pinkel et al., 1998, Ishkanian et al., 2004) and, more recently, protein (Eckel-Passow et al., 2005) microarray technologies.

In a conventional two-color microarray experiment (Fig. 1) two compared samples are labeled using different fluorescent dyes (typically the red-fluorescent dye, Cy5, and the green-fluorescent dye, Cy3), mixed and then co-hybridized to the DNA clones spotted regularly on the microarray. The array is scanned with a high spatial resolution at the corresponding fluorescent wavelengths, and at each scanned pixel the fluorescence intensities are recorded in two color channels (Cy5 and Cy3). The experiment aims to estimate the ratio of the measured intensities for each spot, reflecting differential gene (cDNA technology) or protein expression or a change in DNA copy number (CGH technology) between the test and control samples for the corresponding gene. These ratios are the primary source of information for the subsequent analysis of the microarray data, such as normalization, clustering, classification, differential expression analysis, etc. The main components of the microarray image analysis pipeline for spots include localization, quantification and quality control.

Spot localization involves: (i) identifying the position of each spot on the array to associate it with the spotted clone; and (ii) establishing the borders between the neighboring spots to allow further independent data processing (extracting quantitative information) for each spot. Although spot localization can in principle be done manually, automating this process is essential, as fast and reliable localization increases overall analysis performance and allows high-throughput applications. Many localization algorithms (Buhler et al., 2000; Yang et al., 2002; Jain et al., 2002; Angulo & Serra, 2003; Brändle et al., 2003; Rueda & Vidyadharan, 2006, Ceccarelli & Antoniol, 2006) have been proposed. Some of them require either prior knowledge of some image-specific parameters or direct user participation to find grids. The others are "fully automatic", meaning that different images can be processed without making adjustments for each particular image. However, even for these algorithms, there are always limitations in the automation process because of unpredictable deviations from the assumed array design, high contamination levels or large numbers of missing spots

that cannot be tolerated by the algorithms. In fact, each of the "fully automatic" algorithms has certain limits, and new attempts will never be stopped to push these limits further.

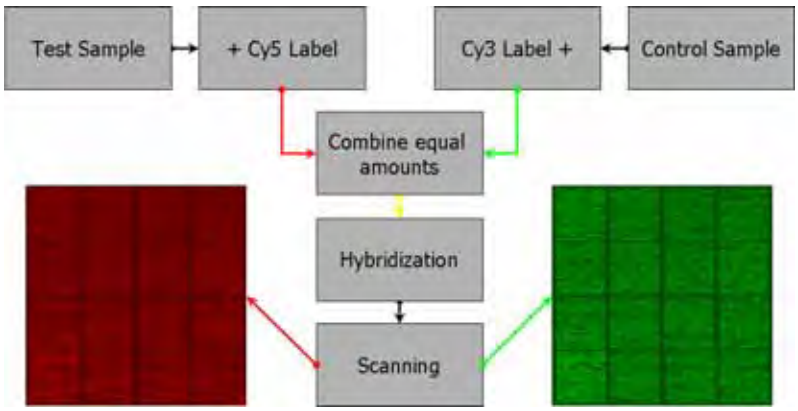


Fig. 1. Two-color comparative microarray experiment.

The aim of the spot quantification is to estimate the ratio. There are two approaches to do that. One is a direct arithmetic ratio of the background-corrected fluorescence intensity estimates in the two color channels (Yang et al., 2002; Bozinov & Rahnenführer, 2002; Angulo & Serra, 2003; Glasbey & Ghazal, 2003; Lehmussola et al., 2006; Axon Instruments, Inc. 2005), and the other is the slope of the linear regression plot of the Cy5 versus Cy3 fluorescence intensities (Jain et al., 2002; Axon Instruments, Inc. 2005). The first approach requires the identification of both the foreground — the measured spot — and the background — typically the level of non-specific hybridization. Large diversity of the algorithms for spot segmentation and background estimation (Lehmussola et al., 2006) highlights the complexity of this problem. The second approach, based on linear regression methods, does not require precise isolation of the spots and identification of the background areas. This method would be rather straightforward, if there were no aberrant or outlier pixels that can strongly affect the slope of the linear regression.

Each ratio estimate should be accompanied by some measure of quality demonstrating the level confidence in the obtained ratios. To determine spot quality we need to have a clear definition of a good spot, or a list of all possible distortions that may spoil the spot. The diversity of instrumental platforms and instrumental and biological factors that may influence the result makes formalization difficult and unlikely to be universal. Several attempts have been made to approach the problem (Buhler et al., 2000; Brown et al., 2001; Wang et al., 2001; Chen et al., 2002; Hautaniemi et al., 2003; Bylesjö et al., 2005). Generally a number of parameters characterizing the spot, such as signal-to-noise ratio, size, circularity, etc., are introduced. These parameters have to be combined into an overall quality value to be used as a confidence level in the follow-up analysis. As individual quality scores generally do not contribute equivalently to the composite quality score, we need to evaluate the weights that control the input of each individual score. For that, training procedures, in which the user classified a set of representative spots into a number of groups ranging from good to bad spots, were proposed (Buhler et al., 2000; Hautaniemi et al., 2003; Bylesjö et al., 2005). This requires an expert to evaluate at least a couple of hundred spots to achieve a good approximation, which is a difficult and time-consuming task.

In this Chapter, we will present a set of advanced algorithms for microarray spot localization, quantification and quality control. We will deal with the rectangular array design. This is the most widespread of the designs used and is also exclusively used within our Institute. In this design, the spots are aligned horizontally and vertically and can be arranged in blocks containing different numbers of spot rows and spot columns. The developed algorithms aim at making analysis more resistant to array contamination and at eliminating user participation at all stages of image processing. The algorithms can be applied to analyze images in one-, two or multi-color microarray experiments. Specific tools have been also developed for ratio evaluation in the two-color comparative experiments.

We present a "fully automatic" spot localization algorithm (Novikov & Barillot, 2006a), which is able to process images of different designs without specific user contribution. We also aimed to make it robust with respect to contamination and missing spots on the array. The developed algorithm is non-supervised and deterministic, ensuring reproducible results. It is assumed that the number of block rows and columns and the number of spot rows and columns within each block are available for analysis as input values.

We have developed a statistical procedure that systematically searches and removes aberrant or outlier pixels (Novikov & Barillot, 2005b). This gives a higher level of confidence in the linear regression ratio estimates. However, as linear regression can give biased estimates when there is a high level of statistical noise (a low correlation between the Cy3 and Cy5 color channels), we still keep estimates from the spot segmentation algorithm. However, after removing aberrant pixels the segmentation algorithm also gives more robust estimates, and there is a greater agreement in the ratio values obtained for both methods.

We have developed a two-level segmentation approach: one intensity level is used to identify spots and the other one separates background areas. Pixels with intensities between these two levels are ignored (buffer zone). We apply the *k*-means adaptive pixel-clustering algorithm (Bozinov & Rahnenführer, 2002) to identify the spot and the background intensity levels. Pixels that are used in the adaptive clustering for the spot and background level estimation are selected from constrained intensity regions. Spot pixels are subject to further geometrical constraints.

We have developed an original set of spot quality characteristics and a model that maps this set into an overall quality value. An automatic training procedure evaluates the contribution of each marginal quality characteristic into the overall quality (Novikov & Barillot, 2005a). This procedure is based on information from replicated spots, located on the same array or over a set of replicated arrays, and assumes that unspoiled replicated spots must have very close intensity ratios, whereas poor spots yield greater diversity in the ratio estimates. Conceptually this approach can be considered as a combination of the "empirical" (based on replicates) and "predictive" (based on quality characteristics) quality assessment methods (Ritchie et al., 2006). The obtained weights can then be used to establish a critical limit for each quality characteristic, such that if a spot's characteristic exceeds its critical limit, the spot is declared a "bad" spot.

The applicability of the developed algorithms has been tested and confirmed using simulated artificial images and experimental images of different array designs used within our Institute and CGH images obtained from the UCSF Cancer Center. These algorithms are included in the software package MAIA (http://bioinfo.curie.fr/projects/maia/), which offers a complete solution for microarray image analysis.

2. Spot Localization

As for other automatic spot localization algorithms (Jain et al., 2002; Angulo & Serra, 2003), we take projections of the intensities in the pixel columns on the X (horizontal) axis and in the pixel rows on the Y (vertical) axis. However, instead of taking the overall intensity directly, we correct it by the amount of regularity in the corresponding row or column, so that bright but very irregular regions are systematically penalized. The developed algorithm transforms fluctuations of the intensity in each pixel row or column of the image into a special parameter that takes into account the regularity of these fluctuations.

2.1 Spot regularity profiles

Regularity components. For each pixel row or column we choose an intensity threshold, T, and isolate continuous regions of pixels with intensities, I_l, higher than T (bright regions): $I_l > T$, and lower than T (dark regions): $I_l \leq T$, l=1,…,m, where m is the number of pixels per row or column. Each bright region can be characterized by its center position $\mu_n(T)$, length $\lambda_n(T)$ and mean intensity $F_n(T)$. For each dark region we estimate its mean intensity, $B_n(T)$.
We then define four components based on these estimates that contribute to the regularity parameter. The most important component is the overall intensity of the bright regions:

$$S(T) = \frac{1}{N(T)} \sum_{n=1}^{N(T)} F_n(T) - \frac{1}{N_B(T)} \sum_{n=1}^{N_B(T)} B_n(T) \tag{1}$$

where $N(T)$ and $N_B(T)$ are the numbers of bright and dark regions at the threshold level, T. The three following parameters deal with the regularity of the bright regions. The first parameter penalizes deviations from the expected spot size, D, of the bright regions:

$$W_1\ (T, D) = \frac{1}{N(T)} \sum_{n=1}^{N(T)} \left(\frac{\lambda_n(T)}{D} - 1 \right)^2 \tag{2}$$

The second parameter ensures that inter-spot distance is not too small. That is, the centers of two bright regions ($\mu_n(T)$ and $\mu_{n+1}(T)$) should not be closer than the expected spot size, D:

$$W_2\ (T, D) = \frac{1}{N(T)} \sum_{n=1}^{N(T)-1} \left(1 - \frac{\mu_{n+1}(T) - \mu_n(T)}{D} \right)_+^2 \tag{3}$$

where $(x)_+ = x$, if $x > 0$ and $(x)_+ = 0$, if $x \leq 0$. The third parameter controls the number of bright regions:

$$W_3\ (T, H) = \left(N(T)/N(H) - 1 \right)_+ \tag{4}$$

where H is the inter-spot distance and $N(H)$ is the expected number of spots in the corresponding pixel row or column. $N(H)$ can be estimated by dividing the number of row or column pixels by H. As we do not expect the number of bright regions to be more than

N(H), this has to be penalized. On the other hand, we cannot impose a lower bound for *N(T),* as some spots may be missing, but the structure is preserved.

Overall regularity parameter. The intensity component (1) and the three regularity components (2), (3) and (4) are combined into an overall regularity parameter:

$$R(T,D,H) = S(T)\exp\left\{-\gamma_1 W_1\ (T,D) - \gamma_2 W_2\ (T,D) - \gamma_3 W_3\ (T,H)\right\} \tag{5}$$

where γ_1, γ_2 and γ_3 are weights determining the contribution of each regularity component. Since all these components are relative quantities, we expect that none will be over-weighted, and hence the weights can be equalized: $\gamma=\gamma_1=\gamma_2=\gamma_3$, where γ is provided by the user. In our analysis we always take $\gamma = 2$, and we have had no problems with the localization for different experimental designs. However, the robustness of the analysis would be increased if γ (or even γ_1, γ_2 and γ_3) were chosen more specifically.

The threshold level, *T,* can be best determined using a special optimization procedure which searches for *T* from the interval $[I_{min};I_{max}]$ maximizing *R(T,D,H):*

$$R(D,H) = \max_{T\in[I_{\min};I_{\max}]} R(T,D,H) \tag{6}$$

where $I_{min} = \min(I_l)$ and $I_{max} = \max(I_l)$, $l=1,\ldots,m$. Eq. (6) represents the final expression for the regularity parameter. We then calculate a set of regularity parameters for each pixel row *i* or column *j*, leading to a regularity profile in the Y ($R_i(D,H)$) and X ($R_j(D,H)$) directions.

Spot size *D* and inter-spot distance *H*. Although possibly available from the experimental design, spot size, *D,* and inter-spot distance, *H,* are not required as prior values. We assume only that *D* and *H* are related as $D = H(1-\alpha)$, where α is the ratio of the inter-spot gap to the inter-spot distance and should be provided by the user. A very precise value of α is not essential. We always take $\alpha = 0.25$, and it appeared to be very stable with respect to different array designs. As *D* is directly available from *H*, we can omit *D* from the notation of the regularity parameter, so that *R(H)* will be used instead of *R(D,H).*

We can obtain H_0, an initial approximation for *H,* by dividing the total number of pixels in the X or Y direction of the array by the total number of spots in the corresponding direction. This is only a rough estimate, but it is sufficient for building the regularity profiles, $R_k(H)$, where $k = i$ for the Y direction and $k = j$ for the X direction (Eqs. (5) and (6)).

We could have, using the profiles obtained, estimated *D* by dividing the number of pixel rows or columns with high regularity by the total number of spots in the Y or X directions, respectively. However, the spots are almost never perfectly aligned and they can get mixed up and become unrecognizable on the one-dimensional axis irrespective of the cutoff level chosen for the regularity profile. This leads to overestimation of the lengths of the regions with high regularity and consequently to an overestimate of *D*.

If all spots within each block overlapped completely in the projections, we could estimate *H* as the ratio of the number of pixel rows or columns with a regularity higher than the selected level to the total number of spots in Y or X directions, respectively. However, as the spots within a block may, even after projecting pixel rows and columns, be separated by dark gaps, the length of the bright regions, needed to evaluate *H*, may be underestimated. To ensure realistic *H* we overlap the spots by superimposing the given profile with itself shifted to the left or right by a certain number of pixels. Complete overlapping of the

neighborhood spots can be achieved by setting the number of pixels used in the profile shifting to the correct value for the inter-spot distance, *H*. We assume that the neighborhood spots are completely overlapped when the number of dips (regions with a regularity lower than the selected level) in the overlapped regularity profile should not be larger a limit defined as the number of blocks plus one. A small number of dips can indicate that neighboring blocks are also indistinguishable.

We search for the highest level of regularity profile that gives the largest number of dips but not larger than the defined limit. The corresponding *H* is then considered as the final estimate. If number of dips is larger than the defined limit for any level of regularity (and correspondingly for any *H*), then the regularity level giving the lowest number of dips is selected, despite being greater than the defined limit. This situation occurs for relatively bright contamination in the positions where there are no spots according to the array design.

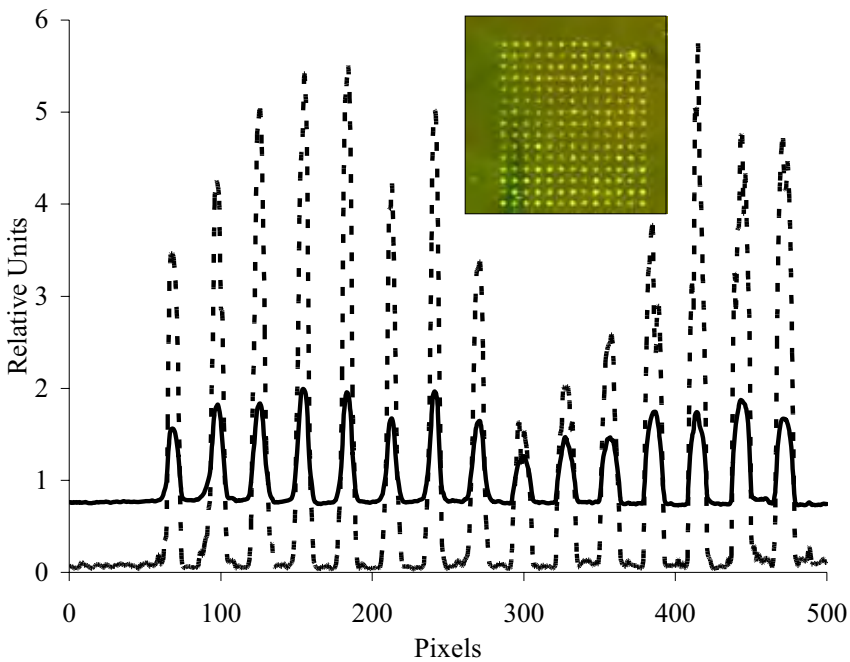


Fig. 2. Intensity (solid line) and regularity (dashed line) profiles for microarray image segment (inset) obtained by projecting on Y axis.

The advantage of using regularity profiles instead of simple intensity profiles is demonstrated in Fig. 2. The regularity profiles (dashed lines) ensure a larger dynamic range (signal to background) than the intensity profiles (solid lines). This leads to better identification of the background regions where it would be expect to find a separation between different spot rows or columns.

Note that each of the approaches that use intensity projections (e.g. Jain et al., 2002; Angulo & Serra, 2003; Brändle et al., 2003) could be reinforced if, instead of simple projections, measures based on the regularity parameter were used.

2.2 Generation of the localization grid

Block separation. First, we use the regularity profiles to look for the borders between the blocks. To increase robustness, the whole array is divided into segments (Fig. 3). If we need to identify the borders between the blocks in the X direction, we take segments in the Y direction with the height of the segment, in pixels, being equal to the height of the image in pixels divided by the number of blocks in the Y direction (NBY). We identify the block borders in the Y direction by taking segments in the X direction with the width of the segment, in pixels, equal to the width of the image in pixels divided by the number of blocks in the X direction (NBX).

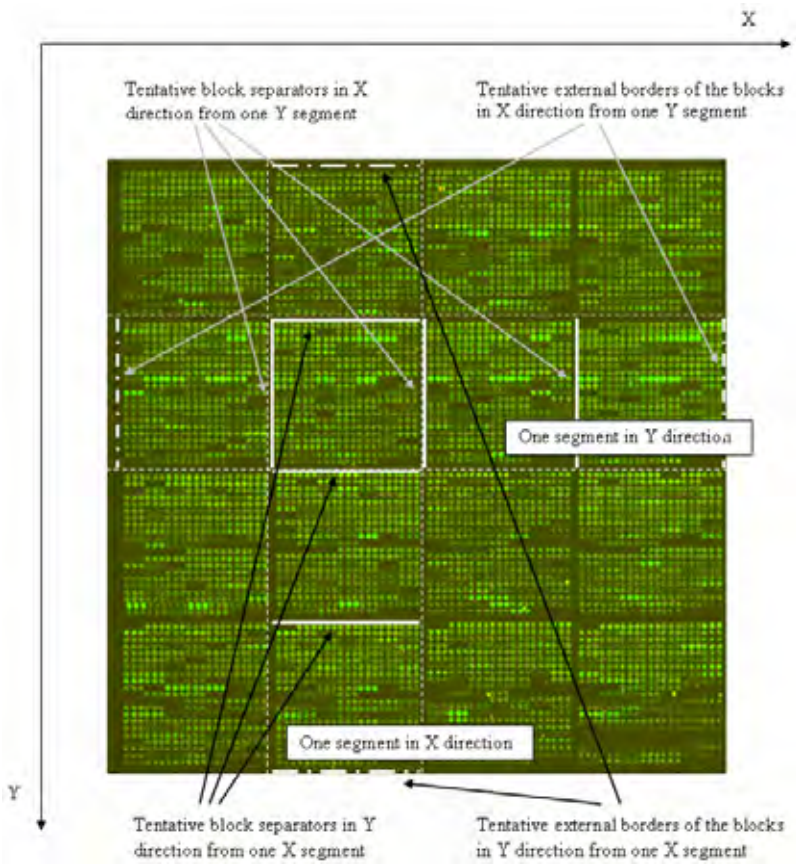


Fig. 3. An example of the separation of the microarray image into segments. There are four sets of tentative block separators and four sets of tentative external borders in the X direction, as four segments (according to the number of blocks) are isolated in the Y direction. Similarly, four tentative block separators and four sets of tentative external borders can be built in the Y direction.

If the blocks are well separated, we can proceed in the following way. For each segment we identify positions separating the blocks by looking for the maximal intervals between the peaks in the regularity profiles. Thus we obtain NBY (in X direction) or NBX (in Y direction)

possible sets of block separations. The best set is the one that has the most regular structure. We calculate the median width of the blocks in NBY sets and the median height of the blocks in NBX sets, and the set, in either the horizontal or vertical separation that gives the smallest deviation from the corresponding median is selected as a final one.

However, this approach is not applicable for arrays where the distance between two neighboring blocks is similar to the distance between the neighboring spots. In this case we take advantage of the fact that the blocks are regularly distributed over the array, and we place the borders equidistant between the external borders of the blocks. These regions have to be long enough to be considered as initial spots in the blocks. We require that the first high-level region must be longer than βD, where β is provided by the user and characterizes the filtering properties on the edges of the array. A default value of $\beta = 0.2$ had been found to be the most relevant for the microarray images of different designs and noise levels that we have tested. The external borders of the blocks are calculated for all segments described above (Fig. 3), and the median estimates are taken. We use two localization iterations to increase the precision of block separation. The first approximation of the grid is used to adjust the borders of the blocks at the second iteration.

Spot localization. After blocks are separated, we have to identify the borders between the spots within each block. Although it may appear straightforward to use regularity (or intensity) profiles to draw lines at the positions of minimal regularity of the corresponding profiles to separate the neighboring spots this often results in errors, because the positions of the minima can be due to random regularity fluctuations. Therefore, we have developed a robust procedure searching for the spot separations. It uses the same optimization procedure as for the overall regularity parameter, but instead of the intensity, I_l, we use regularity profiles in the X ($R_j(H)$) or the Y ($R_i(H)$) directions. An example of the row regularity profile (Y direction) for a one block (shown in inset of Fig. 2) is given in Fig. 2 in dashed line. Applying a set of criteria represented by Eqs. (1), (2), (3) and (4) for each block we can build up a vertical regularity parameter $R_Y(R_i^*,H)$ (Eq. (5)) using a row regularity profile, $R_i(H)$, and a horizontal regularity parameter $R_X(R_j^*,H)$ (Eq. (5)) using a column regularity profile $R_j(H)$. The parameters $R_Y(R_i^*,H)$ and $R_X(R_j^*,H)$ are dependent on the threshold levels R_i^* and R_j^*, and should ensure the highest regularity of the regularity profiles $R_i(H)$ and $R_j(H)$ (see Eq. (6)). However, in difference to Eq. (6), R_i^* in $R_Y(R_i^*,H)$ is determined from the interval between $\min(R_i(H))$ and $\max(R_i(H))$, where i is the row number; and R_j^* in $R_X(R_j^*,H)$ belongs to the interval between $\min(R_j(H))$ and $\max(R_j(H))$, where j is the column number.

Note that the optimized values of $R_Y(R_i^*,H)$ and $R_X(R_j^*,H)$ are of no use in this context. The middle positions of the intervals in the regularity profiles lower than the optimal threshold level are taken as the positions separating spot rows or columns.

3. Spot Quantification

After spot localization step, we assume that the spots are identified and well localized in squares (called spot cells), so that each spot cell can be processed independently of the others. We calculate the ratio of the spot using either a linear regression or a segmentation (spot contouring or spot isolation) approach.

3.1 Ratio estimation based on linear regression

The linear regression approach represents the ratio as the slope of the linear regression fit of the pixel intensities in two channels (Fig. 4). We use orthogonal regression (Kendall & Stuart, 1979, Dissanaike & Wang, 2003) since measured fluorescence intensities are statistically distorted in both color channels. Spot segmentation is unnecessary with this method, as background pixels are concentrated at the origin of the linear regression plot and do not influence the slope of the regression line (Fig. 4). However, outlier or aberrant pixels within the spot cells, even in small numbers, can strongly influence the regression line, thus biasing the ratio. With the aim to fully exploit the advantages of the linear regression approach we tried to reinforce this procedure by systematically filtering out aberrant pixels.

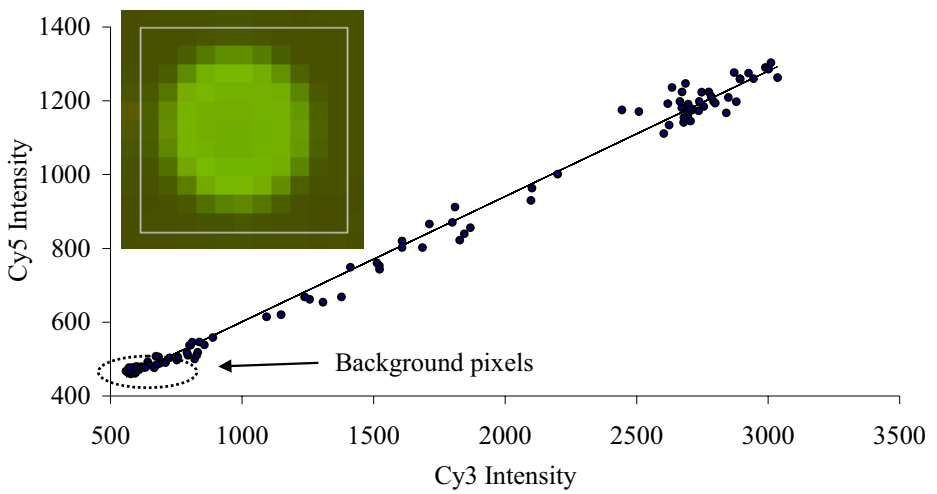


Fig. 4. Estimation of the ratio using linear regression fit for a good spot with a correlation coefficient of 0.99 (ratio = 0.339). The background pixels are grouped near the origin of the linear regression plot.

Different approaches exist to detect statistical outliers in experimental data (Rousseeuw & Leroy, 2003; Atkinson & Riani, 2000). Well-advanced high-breakdown algorithms (Rousseeuw & Leroy, 2003) or forward search algorithms (Atkinson & Riani, 2000) are based on repetitive resampling of experimental data and iterative linear regression approximation. This makes these algorithms computationally infeasible for microarray image analysis, where thousands of spots, each one containing 100-500 data points (pixels), should be processed in seconds. Therefore, we have to look for more approximate algorithms, which, however, can ensure higher efficiency. For microarray images, we expect that the majority of the spots should not have outliers, and the number of outliers for possibly contaminated spots should not be too high. Therefore it would be advantageous to have an algorithm that could quickly identify outlier presence, without being involved in time-consuming iterations. With this aim we have adopted the backward search algorithm with single-case diagnostics (Rousseeuw & Leroy, 2003). The advantage of this algorithm is that if the procedure can not identify an outlier at the first iteration, it proceeds to the next spot, thus saving processing time. Although single-case diagnostics are known to be less efficient

(Rousseeuw & Leroy, 2003) for the data with tight groups of outliers, in our work we rarely had problems: in microarray image, even if several aberrant pixels form a spatial cluster (Fig. 5), they are often very different at the intensity scale (at least in one of two color channels). As outlier intensities are widely distributed, the removal of even one of them changes the quality of the linear regression noticeably, facilitating the one-pixel (or single-case) backward search procedure for spot quantification.

The backward search procedure, in our implementation, examines suspicious pixels by evaluating the quality of the linear regression fit with and without the suspicious pixel. We quantify the fit quality by the residual variance, s^2. The smaller s^2 is, the closer the linear regression line is to the experimental data. The ratio of the s^2 values is calculated for the fit with the tested pixel and for the fit without. If this ratio is larger than a critical value of the F-distribution at a user-defined confidence level, the pixel will be marked as aberrant. We select pixels with the highest intensity in either of two channels first and then select pixels having the largest deviation from the fitted regression line. To take into account the fact that the distortions caused by pixels from the top of the intensity scale and by pixels lying off of the linear regression line, may be different, we apply different confidence levels for the F-statistics for these pixels. In our analysis we use 0.01 as a confidence level for the pixels from the top of the intensity scale and 0.1 for the pixels lying off of the linear regression line.

For the high-intensity pixels we also perform another test to determine how far their intensities are from the averaged intensity of the other pixels within the spot cell. This detects pixels, far away from the other pixels, that do not distort the linear regression line. Although these pixels may not change the ratio, they could be considered as aberrant pixels, as we expect to see an almost continuous distribution of pixels intensity (Fig. 4). The procedure performs iteratively until no more aberrant pixels are detected.

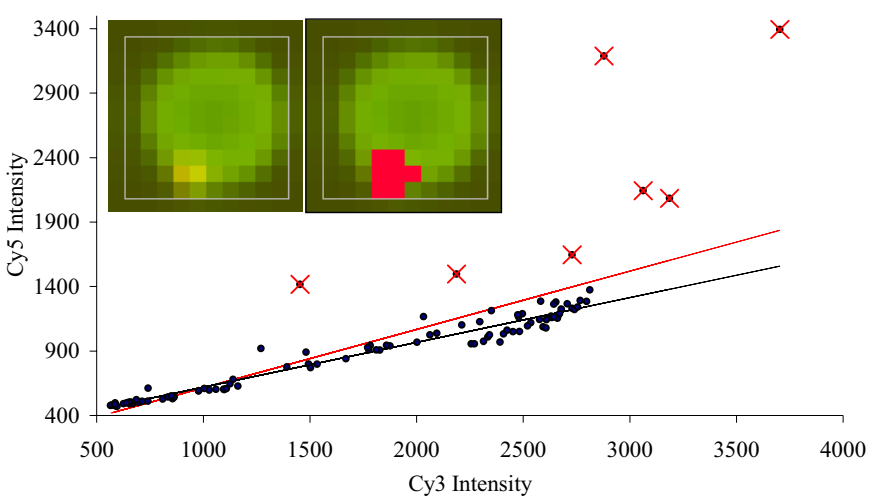


Fig. 5. Estimation of the ratio using linear regression fit for a spot with aberrant pixels (red crosses). The estimated ratio with the aberrant pixels is 0.45 (a), when the aberrant pixels are removed it decreases to 0.37 (b). The estimated ratios for the other two spots from the same triplicate are 0.342 and 0.332.

An example of the outlier detection is presented in Fig. 5. It is important to note that the regression approach is capable of detecting contamination pixels that are geometrically inseparable from the spot. Therefore, the developed procedure can be considered not only as a procedure for correcting ratio recovery, but also as a procedure to repair the spot and to improve the quality of experimental material. It requires, however, that the contamination clearly deviates from the straight regression line, which is defined by the majority of "good" pixels from the spot. The filtering procedure can detect up to ~30% of aberrant pixels with respect to the number of spot pixels. For the spots with larger number of aberrant pixels, a safer way would be to flag out these spots rather than to try to identify all aberrant pixels. Besides much higher computational complexity (and hence processing times), high-breakdown filtering algorithms may have difficulties to distinguish between contaminating pixel clusters and useful spots, when these become comparable in size, and contamination is highly correlated in two color channels.

One potential problem of linear regression approach is when one image (Cy3) is shifted relative to the other (Cy5). As this shift increases, the correlation between the two channels decreases rapidly, and linear regression fit becomes poorly defined. To solve this problem we have developed a special procedure for the automatic identification and removal of shift between two images. The procedure moves one image with respect to the other one to obtain the largest correlation coefficient for a number of representative spots. These spots are selected according to two criteria: they should be bright enough, but not beyond the dynamic range of the registered intensities; and they should not contain pixels a lot brighter than most of the pixels in the corresponding spot cell.

3.2 Ratio estimation using spot segmentation

The spot segmentation approach identifies spots and background areas. The ratio is then defined as

$$r = (F_{Cy5}-B_{Cy5})/(F_{Cy3}-B_{Cy3}) \tag{7}$$

where $F_{Cy5}(F_{Cy3})$ is either the mean or median estimate of the spot intensity in the Cy5(Cy3) channel, and $B_{Cy5}(B_{Cy3})$ is either the mean or median estimate of the background intensity in the Cy5(Cy3) channel.

We have developed a multi-level segmentation approach where a segmentation algorithm is first applied to isolate spots and then to identify background pixels. The algorithm is applied to the combined image: $F_i = F_i^{Cy5}A_{Cy5}+F_i^{Cy3}A_{Cy3}$, where F_i is the combined intensity of the i-th pixel, $F_i^{Cy5}(F_i^{Cy3})$ is the intensity of the i-th pixel in the Cy5(Cy3) color channel, and A_{Cy5} and A_{Cy3} are the normalization constants: $A_k = \min(M_{Cy5},M_{Cy3})/M_k$, $k = \{Cy5,Cy3\}$, where $M_{Cy5}(M_{Cy3})$ is the mean intensity of the pixels located along the borders of the given spot cell in the Cy5(Cy3) color channel.

The spot is isolated by establishing the signal level, L_s, such that all pixels with intensities higher than L_s will be classified as potentially belonging to the spot. We used the k-means adaptive pixel-clustering algorithm (Bozinov and Rahnenführer, 2002) to do this. However, we had problems when this algorithm was applied to segment spots with relatively smooth edges. Some pixels may be clearly brighter than the background, but not bright enough to be included into the spot. To regularize the solution, we establish an intensity limit, U, such that only pixels with the intensities higher than U participate in the spot segmentation.

We use Chebyshev's inequality (Fisher & van Belle, 2003) to define U as $M+W/(1.35p^{1/2})$, where p is a user-defined confidence level for the intensity distribution of background

pixels, M is the median and W is the inter-quartile distance of pixel intensities located along the borders of the given spot cell (these pixels are expected to be purely background pixels). Then pixels with the intensities higher than U are classified according to the k-means adaptive pixel clustering algorithm to estimate L_s.

After selecting the bright pixels some geometrical constraints need to be imposed. We define a spot circle, centered on the center of mass of all the bright pixels from the given spot cell, with the radius $(0.5Z/\pi)^{1/2}$, where Z is the number of pixels with intensities higher than L_s. If it turns out that the number of bright pixels within the circle is relatively small ($<0.5Z$), we increase the radius by one until the number of pixels covered by the circle becomes equal or higher than $0.5Z$. For spots with a circular shape it should happen at the first trial. More attempts are needed for spots with more peculiar shapes (e.g. donut-like). The bright pixels within this circle are considered as belonging to the spot. All other bright pixels in the same spot cell are considered as potential space outliers. Further steps resemble the seeded region growing (Yang et al., 2002). The space outliers are converted into spot pixels only if one of their neighbors is already a spot pixel. It performs iteratively building up a cluster of bright pixels, which are geometrically inseparable from the originally defined spot pixels. These pixels constitute a spot and the remaining bright pixels are considered as space outliers that should be ignored during further analysis.

Spot pixels with excessively high or low intensity with respect to the majority of spot pixels can also be discarded. The admissible range is defined as "median of spots pixels" ± "inter-quartile distance of spot pixels"/$(1.35p^{1/2})$, where p is a user-defined confidence level for spot pixels. This filtering is appropriate for flat spots with large amount of pixels.

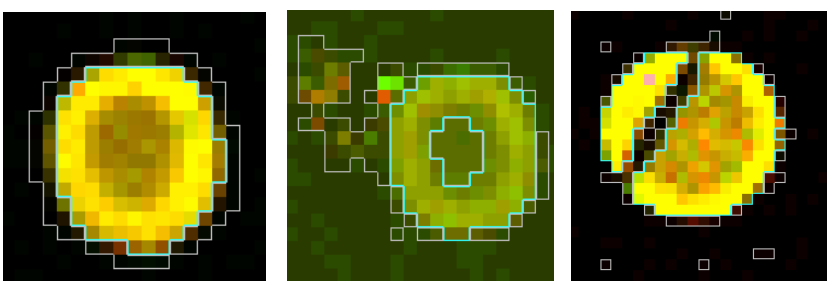


Fig. 6. Segmentation examples: pixels within turquoise contours represent spots and pixels outside gray contours represent background areas.

Finally, we identify the areas used to calculate the background levels, B_{Cy5} and B_{Cy3}. The different approaches for calculating the background vary considerably (Yang et al., 2002; Bozinov & Rahnenführer, 2002; Bengtsson & Bengtsson, 2006; Axon Instruments, Inc. 2005). We search for the background level, L_b, such that all pixels with intensities lower than L_b are classified as background, and pixels with intensities from the interval $[L_b;L_s]$ comprise the buffer zone ignored in further quantification. L_b, in our implementation, is estimated from the k-means adaptive clustering applied to pixels with intensities from the interval $[M;U]$. This procedure identifies background areas within a spot cell. Similar to (Axon Instruments, Inc. 2005) the background estimates, B_{Cy5} and B_{Cy3}, are taken from all background areas within approximately two spot-cell-size regions centered at the current spot.

Several examples of segmentation for spots of different shapes and geometries are shown in Fig. 6. As one can see, the developed algorithm is able to produce predictable contours for broad range of different spots.

3.3 A combined approach to unique ratio estimation

The performance of the linear regression approach depends on the level of statistical noise in the detected images and hence on the level of correlation between two (Cy3 and Cy5) color channels. For images with a high correlation coefficient (~0.90), the linear regression approach is often better than the segmentation approach, and filtering is more effective, as any contamination is better recognized by the linear regression fit. For noisier images, the regression approach is less efficient in filtering and may also produce biased estimates. For such images, the segmentation algorithm generally demonstrates better performance.

A general strategy to estimate the ratios can be composed of two steps. First, linear regression filtering is applied to each spot. This removes aberrant pixels for highly correlated signals, and leaves the data largely unaltered for noisy images. Then segmentation approach is used for the final ratio estimation according to Eq. (7), where $F_{\{Cy5,Cy3\}}$ and $B_{\{Cy5,Cy3\}}$ are the mean estimates for the spot and background intensities, respectively. Mean estimates are more precise (Fisher & van Belle, 2003), but can be affected by outliers. However, as the outliers have been already removed by the linear regression filtering, we can use the mean values. Although estimation using the segmentation estimator may be not as good as the linear regression estimator for highly correlated spots, the difference is generally so unimportant that we can sacrifice some quality for generality. We call this two-step algorithm the regression filtered segmentation estimator (RFSE).

In general, the idea to perform preliminary filtering of microarray images is not new. There have been a number of publications reporting application of the median filter (Glasbey & Ghazal, 2003), top-hat filter (Yang et al., 2002; Glasbey & Ghazal, 2003) or a set of morphological operators (Angulo & Serra, 2003). However, all these techniques, while reducing noise in images, also change intensity levels of the majority of pixels on the array, regardless of whether these pixels are outliers or not. For example, existing filtering procedures may dissolve micro-cluster of aberrant pixels (like the one shown in Fig. 5), so that it will not be seen any more. However, exceptionally high intensities from the outlier cluster will implicitly influence the intensity of both, the neighboring "good" pixels and the new "good" pixels that will substitute the outliers. This may result in biased intensity and ratio estimates. Contrary to that, our approach specifically eliminates outlier pixels, otherwise not distorting data. It also allows for visual examination of the contaminating pixels to evaluate sources of possible problems in microarray experiment.

4. Spot Quality

Each ratio estimate should be accompanied by some value of quality reflecting the level confidence in the obtained ratios. This value is derived from a set of quality characteristics generated by spot quantification procedures (linear regression and spot segmentation).

4.1 Spot characterization by quality parameters

The generated quality characteristics (x) may be defined on any domain, but we scale them ($q(x)$) to fit the range between 0 (bad spot) and 1 (good spot). This facilitates further quality analysis. For scaled quality characteristics we use another term: quality parameters.

Coefficient of determination (*CD*) of linear regression signifies the degree of linear relationship between the intensities in the Cy3 and Cy5 channels. High values of *CD* (approaching 1) are expected for good spots. Low values suggest either relatively bright but non-correlated contamination, or strong statistical noise normally characterizing low-level (or missing) spots. $q(CD) = CD$.

Durbin-Watson statistic (*DWS*) evaluates the presence of the first-order autocorrelation in the residuals of the linear regression fit. It ranges from 0 to 4, 0 being a positive correlation and 4 being a negative correlation. A *DWS* value close to two indicates that the residuals are uncorrelated and the model is appropriate. Large deviations from two, resulting from systematic patterns in the residuals plot suggest that the spot cannot be modeled in terms of a simple linear regression. $q(DWS) = 1-|DWS-2|/2$.

Spot contamination is the number (*SC*) of the aberrant pixels (within the spot contours) flagged out by the filtering procedure. $q(SC) = 1-SC/Z$, where Z is the number of pixels within the spot contour.

Diameter of the spot: $D = 2(Z/\pi)^{1/2}$. As the true value for the spot diameter may be difficult to establish, we use a typical value taken as the median diameter over all spots on the array. Spots with exceptionally small or large diameters should be penalized. $q(D) = exp\{D_T-D\}$, if $D>D_T$ and $q(D) = exp\{D-D_T\}$, if $D<D_T$ where D_T is the typical diameter.

Geometrical symmetry parameter measures deviation of the contoured spot from the ideal circle. We divide both the real spot and the ideal circle into eight segments (pie slices defined as $[k\pi/4;(k+1)\pi/4]$, $k = 0,\ldots,7$) and we count the number of pixels belonging to the spot (Z_{si}, $i = 1,\ldots,8$) and to the circle (Z_{ci}, $i = 1,\ldots,8$) for each segment. The sum of the absolute relative differences $GS = \sum|Z_{si}-Z_{ci}|/Z_{ci}$ is then taken as an indicator of quality. For ideal circular spots *GS* should approach 0, whereas highly deformed (un-circular) spots can be recognized by high *GS* values. $q(GS) = exp(-GS)$.

Intensity symmetry of the spot is defined as $IS = \sum|F_i-F|/F$, where F_i, $i = 1,\ldots,8$ are the mean intensities for the same 8 segments and F is the mean intensity within the spot. Although a spot may have perfect circular shape, it may contain very bright (or dark) and highly concentrated groups of pixels originating from pieces of dust or other contamination. $q(IS) = exp(-IS)$.

Coefficient of variation of two ratio estimates: $CVR = 2^{1/2}|RR-RS|/(RR+RS)$. Despite the different methods of ratio estimation (one by the linear regression approach (*RR*), and the other by the segmentation algorithm (*RS*)), the variation between the two obtained ratios should be as small as possible. Large variations between the two estimates may indicate a problematic spot. $q(CVR) = exp(-CVR)$.

Uniformity of the background along the grid lines separating neighborhood spots is defined as $UB = \sum|B_i-B|/B$, where B_i, $i = 1,\ldots,8$ are the mean intensities in 8 segments of the grid line around the spot, and B is the mean intensity for the whole grid line around the spot. Large *UB* values may discover presence of relatively bright contamination around the spot, large variability in the background or merged neighboring spots. $q(UB) = exp(-UB)$.

Absolute level of background (*AB*) calculated from the local area around the spot ($AB = \max(B_{Cy5},B_{Cy3})$) is compared to the median background level over all spots on the array.

Spots with exceptionally high *AB* values may indicate the presence of the contamination areas, which are larger than the size of the spot. $q(AB) = exp(1-AB/AB_T)$, if $AB>AB_T$ and $q(AB) = exp(AB/AB_T-1)$, if $AB<AB_T$. where AB_T is the typical background level.
Signal (*S*) is defined as $S = \min(F_{Cy5} - B_{Cy5}, F_{Cy3} - B_{Cy3})$. $q(S) = 1$, if $S>S_T$ and $q(S) = exp(S/S_T-1)$, if $S<S_T$, where S_T is the median signal over all spots on the array.
The developed quality parameters, although not optimal, have led to reasonable results for most of the experimental and simulated situations we tested. Of course, there may be a possibility to formalize some of these parameters more precisely and/or to develop new parameters accounting for other types of distortions.

4.2 Spot quality analysis

We consider two aims of spot quality analysis. The first is to combine the marginal quality parameters into an overall quality value. This value can be used either to flag out directly spots with a quality lower than a user-defined threshold, or, in the follow-up image analysis procedures (normalization, classification, clustering, etc.) as a parameter characterizing the level of confidence in the obtained Cy5/Cy3 ratios. The second aim is to identify a critical range for each quality characteristic. If a certain quality characteristic of the spot falls in this range, the corresponding spot is classified as a "bad" spot.

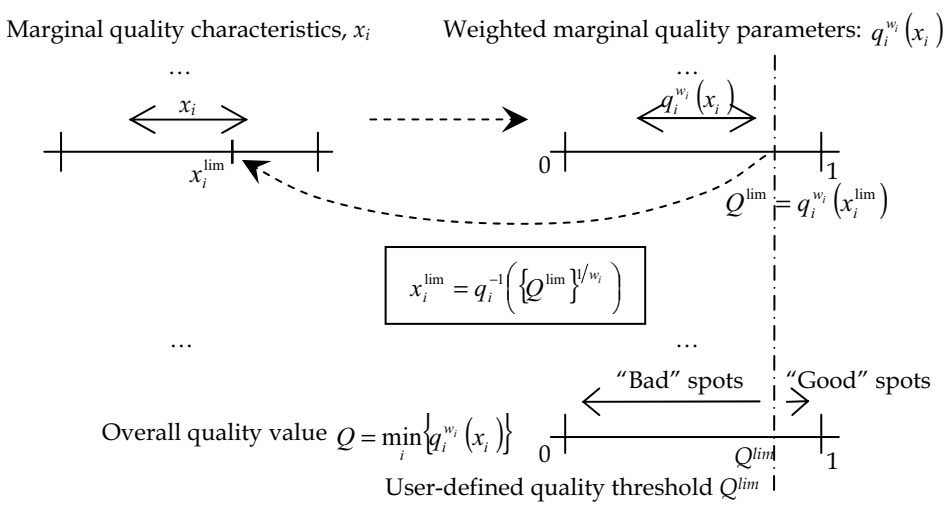


Fig. 7. The correspondence between the quality characteristics, quality parameters and overall quality value.

Overall quality. We used the following definition for the overall quality value:

$$Q = \min_i\{q_i^{w_i}\}, \tag{8}$$

where $q_i = q_i(x_i) \in [0;1]$ are the marginal quality parameters for $x = \{CD, DWS, SC, D, GS, IS, CVR, UB, AB, S\}$ and w_i are the weights that control the input of the corresponding quality

components into the overall quality value. A link between the weight w_i and the critical value x_i^{lim} can be established for each quality characteristic:

$$w_i = \ln\{Q^{\lim}\} / \ln\{q_i(x_i^{\lim})\} \text{ or } x_i^{\lim} = q_i^{-1}\left(\{Q^{\lim}\}^{1/w_i}\right) \tag{9}$$

where $Q^{lim} \in [0;1]$ is the user-defined overall quality threshold, and $q_i(x_i^{lim})$ is the quality parameter calculated for x_i^{lim}. The critical value x_i^{lim} sets up the limit such that if a certain characteristic i exceeds this limit, the corresponding quality parameter $q_i(x_i^{lim})$ will become lower than Q^{lim}. The correspondence between x_i, x_i^{lim}, $q_i(x_i)$, $q_i(x_i^{lim})$, w_i, Q and Q^{lim} is demonstrated in Fig. 7.

Quality weights w_i. The experimental quality parameters, q_i, are directly available from the quantification procedure, whereas the weights w_i (or the critical values x_i^{lim}) are unknown and are not easily guessed or derived from theory. Therefore, the problem of spot quality analysis becomes a problem of weights (w_i) estimation. This can only be solved if additional information is available. Here we consider three possibilities:

1. The additional information may come, for example, from the user expertise. The user has to classify the spots manually (Buhler et al., 2000; Hautaniemi et al., 2003; Bylesjö et al., 2005) and assign a quality value to each spot from a representative subset. These values are then used for training the model (8) leading to a combination of the weights (w_i) such that the overall quality values reproduce the user classification reasonably well.
2. We can manually apply different combinations of the weights w_i and visually appreciate, which spots have been flagged out. The trials must be continued until most of the user classified "bad" spots are eliminated by the chosen combinations of the weights.
3. The weights can be estimated automatically using information available from replicated spots on the same array or over a set of replicated arrays. Unspoiled replicate spots should have very similar ratio values. Large differences between the observed ratios in the replicate spots would signal that some spots from this replicate were irregular. We formalize this approach by first defining the quality value for the replicate:

$$Q_k = \min_j\left\{\min_i\left\{q_{kji}^{w_i}\right\}\right\} \tag{10}$$

where q_{kji} is the i-th quality parameter of the j-th replicated spot in the k-th replicate. Then we require that the ratio variation coefficient in the k-th replicate, V_k, is proportional to the logarithm of Q_k:

$$V_k \sim -\ln\left[\min_j\left\{\min_i\left\{q_{kji}^{w_i}\right\}\right\}\right] \tag{11}$$

The log transform is the most "natural" way to convert [0;1] scale of Q_k into [0;∞) scale of V_k. Finally, exponential transform of Eq. (11) yields

$$\exp\left(-V_k / V\right) = \min_j\left\{\min_i\left\{q_{kji}^{w_i}\right\}\right\} \tag{12}$$

where V is the user-defined characteristic ratio variation coefficient. The weights w_i can be estimated from the best fit of the experimental quality values Q_k to the exponentially transformed ratio variation coefficient V_k (Novikov & Barillot, 2005a). If certain quality factors do not influence the shape of the experimental quality curve Q_k (Eq. (10)), the corresponding weights will be set close to 0. If a certain effect shows up in only a small number of spots, it may be neglected by the optimization procedure, and the corresponding weight will be erroneously small. In this case, manual correction of the weights would be necessary.

In our quality analysis algorithm, user participation is limited to the definition of the characteristic ratio variation coefficient, V. This is somewhat simpler than deciding on the quality of several hundred spots, which is used to teach the algorithm in the manual approach. However, as with other solutions, this algorithm requires representative images to train the model. It is impossible to evaluate confidently the weight of the contribution of the diameter quality parameter, for example, if all spots in the array have the same diameter. Therefore, a careful selection of training images containing a realistic diversity of all possible distortions and artifacts is needed.

In (Novikov & Barillot, 2005a) we have also demonstrated possibilities to perform quality analysis based on replicated spots from different arrays and a possibility to apply quality weights obtained from the analysis of one training image, which should contain replicated spots, to other arrays, which may not contain replicates. The latter example attempts to reproduce an important possibility of designing microarray experiments. A small number of training arrays with replicated spots and representative diversity of possible artifacts can be measured and analyzed. The obtained results can then be used to evaluate the quality of other arrays of similar design, which may not contain replicated spots.

Follow-up image analysis. As it was mentioned earlier, the overall quality value, Q (Eq. (8)), can be used as a parameter characterizing the level of confidence in the obtained Cy5/Cy3 ratios. If, for example, n ratios should be averaged, the weighted mean would ensure a more robust estimate for the average:

$$r = \sum_{l=1}^{n} Q_l \, r_l \Big/ \sum_{l=1}^{n} Q_l \tag{13}$$

where r_l is the Cy5/Cy3 ratio and Q_l, is the corresponding overall quality value (l = 1,…,n). The weighted coefficient of variation is defined as

$$V = \frac{1}{r}\sqrt{\sum_{l=1}^{n} Q_l \left(r_l - r\right)^2 \Big/ \sum_{l=1}^{n} Q_l} \tag{14}$$

Note that the ratio variation coefficient V_k can be determined from Eq. (14), if we set Q_l = 1, l = 1,…,n, with n being the number of spots in a replicate.

5. Testing image processing algorithms

5.1 Image Simulation

In (Novikov & Barillot, 2005b) we have described a software component for Monte-Carlo simulation of microarray images. The simulator accounts for statistical noise and different types of distortions, such as non-specific hybridization and dust. As the values of the ratios are exactly known in the simulation experiments, it allows us to test and compare

objectively different ratio estimation algorithms. The general model for the two-color (Cy3, Cy5) microarray image is given by:

$$F_{Cy3}(i,j) = \sum_{k=1}^{N_S} g\left(i,j,c_k^{sx},c_k^{sy},\rho^s,I^s\right) + \sum_{k=1}^{N_D} g\left(i,j,c_k^{dx},c_k^{dy},\rho^d,I^d\right) \tag{15}$$

$$F_{Cy5}(i,j) = r\sum_{k=1}^{N_S} g\left(i,j,c_k^{sx},c_k^{sy},\rho^s,I^s\right) + \sum_{k=1}^{N_D} g\left(i,j,c_k^{dx},c_k^{dy},\rho^d,I^d\right) \tag{16}$$

where N_S is the number of spots and N_D is the number of dust clusters, c_k^{sx} and c_k^{sy} are the coordinates of the center of a spot, c_k^{dx} and c_k^{dy} are the coordinates of the center of a dust cluster, ρ^s and ρ^d are the approximate radiuses of the spot and dust cluster, respectively, I^s and I^d are the fluorescence intensity in the center of the spot in the Cy3 color channel and in the center of the dust cluster, respectively, and r is the ratio of the test and control samples. Dust is represented by the random distribution over the array of clusters of pixels of varying brightness. We consider that these pixel clusters have an identical shape to the spots and therefore the same analytical representation is used for an ideal spot shape and dust cluster:

$$g(i,j,c^x,c^y,\rho,I) = I\exp\left(-\left\{\left(i-c^x\right)^4 + \left(j-c^y\right)^4 + \left(i-c^x\right)^2\left(j-c^y\right)^2\right\}\Big/2\rho^4\right) \tag{17}$$

The parameters characterizing the spots (c_k^{sx}, c_k^{sy}, ρ^s, I^s and R) are user-defined. For example, the coordinates c_k^{sx} and c_k^{sy}, the radius ρ^s and the ranges for x and y for each spot are defined from a user-defined array design. The user should also specify the number of dust clusters N_D on the array. The other parameters characterizing the dust are random variables, and the probability laws for their generation is a matter of choice. We use uniform distributions for ρ^d (in the interval 0 to ρ_m) and I^d (in the interval 0 to I_m), where ρ_m and I_m are a user-defined maximal dust cluster radius and maximal dust intensity, respectively. We also assume that c_k^{dx} and c_k^{dy} are uniformly distributed over the array. Statistical laws of the dust characteristics can generally be different in the two (Cy3, Cy5) channels.

In the developed simulation model we also account for the nonspecific hybridization and statistical noise:

$$\widetilde{F}_k\,(i,j) = F_k\,(i,j) + B_k + \eta_{Bk}B_kG_B + \sigma(i,j)G_S \tag{18}$$

where k represents either Cy3 or Cy5, B_i and η_{Bk} are the user-defined average and noise-to-signal ratio of nonspecific fluorescence intensity in the color channel k, $\sigma(i,j)$ is the standard deviation of the pixel statistical noise, and G_B and G_S are independent Gaussian random variables with zero mean and unit standard deviation. The exact representation for $\sigma(i,j)$ is defined by the experimental set-up. There are currently three possibilities: $\sigma(i,j)$ can be (i) constant, (ii) proportional to the signal, or (iii) proportional to the square root of signal. The type and quantitative characteristics of the statistical noise are defined by the user.

5.2 Evaluation of the noise resistance using artificial images

All artificial images were generated using the same array design: 4x12 blocks and 21x21 spots within each block with the inter-spot distance of 15 pixels and the inter-block gap of 20 pixels. For all spots in the generated arrays the spot radius, ρ^s, was about 4 pixels, the intensity, I^s, in the Cy3 color channel was 5000 and the ratio, r, of the Cy5 and Cy3 channels was 3. Non-specific hybridization was generated using B_k = 1000 and η_{Bk} = 0.5. The standard

deviation of the statistical noise, $\sigma(i,j)$, at each pixel was proportional to the signal at the corresponding pixel with the noise-to-signal ratio of 0.1. We also added randomly distributed dust clusters with the maximal intensity, I_m = 65535, and maximal radius, ρ_m = 2 pixels. Generated images differ in the number of dust clusters, N_D.

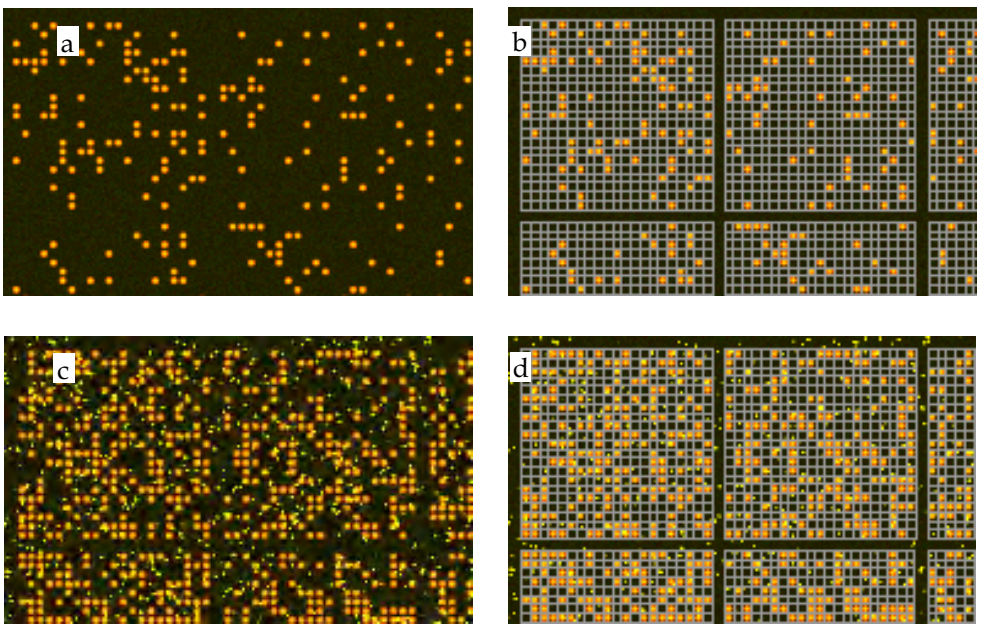


Fig. 8. Fragments of artificial microarray images with 4x12 blocks and 21x21 spots per block: a) the fraction of the bright spots is equal to 15%; no contamination; b) the same image with the generated grid; c) randomly distributed contamination spots are added; the percentage of the bright correct spots is 40% and the number of the contamination spots is equal to the number of the correct spots (*NS* = *ND*); d) the same image with the generated grid.

Localization. We studied the influence of the amount of bright (visible) spots and the level of contamination on the spot localization. Two exemplary artificial images are presented in Fig. 8. One (Fig. 8a) containing only 15% of bright spots randomly distributed over the image, and the other one (Fig. 8c) with randomly distributed contamination spots. For the contaminated array, and the number of dust clusters was equal to the number of true spots ($N_S = N_D$).

Grid placement depends on the distribution of the spots over the array. Therefore, we generated 100 images, each with a random spot distribution, and counted the amount of grids that needed user intervention. For the images without contamination, only 10 of 100 images gave misplaced grids. This happened when first or last spot rows or columns are empty, so that the algorithm shifted the grid by one row or column. For contaminated images grid misplacement occurred in 7 of 100 images. This took place when false spots were recognized as the real spots by the algorithm. Examples of the correctly generated grids in both cases are given in Figs. 8b and 8d.

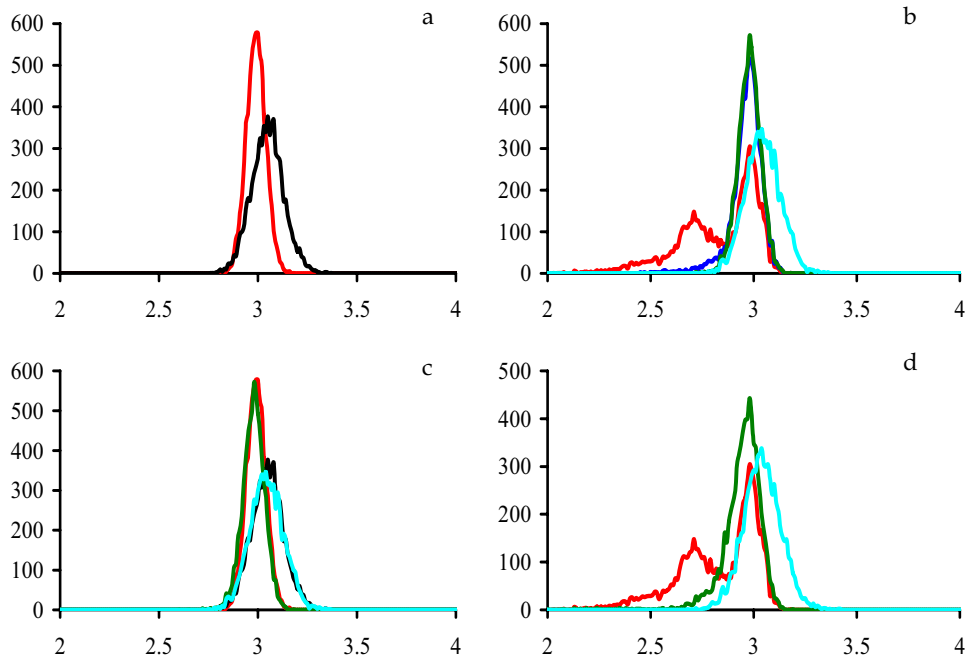


Fig. 9. Histograms of the ratio estimates: a) ratio of means (red) and ratio of medians (black) for the dust-free image; b) ratio of means (red), un-weighted RFSE (blue), weighted RFSE (green) and weighted ratio of medians (turquoise) for the contaminated image; c) weighted RFSE (green) and weighted ratio of medians (turquoise) for the contaminated image, ratio of means (red) and ratio of medians (black) for the dust-free image; d) ratio of means (red), weighted ratio of means (green) and weighted ratio of medians (turquoise) for the contaminated image.

Quantification and Quality. We investigated the influence of the level of contamination on the spot quantification. We used the same array design as before (Fig. 8) with one exception: all true spots were bright and visible. We compared RFSE ratio with the ratio (7) where $F_{\{Cy5,Cy3\}}$ and $B_{\{Cy5,Cy3\}}$ are either the mean (ratio of means) or median (ratio of medians) estimates. We also compared the weighted and un-weighted mean estimates for the average r (Eq. (13)). The un-weighted characteristics were obtained from Eq. (13) by setting all Q_l, l = 1,…,n to 1. The weighted characteristics were calculated with the overall quality values Q_l available from the quality analysis algorithm. As all spots from the simulated image can be considered as replicates, having the same theoretical ratio (r = 3), we artificially split up the total number of spots into the groups of three closely placed spots. These groups, regarded as independent triplicates, can be used to calculate the experimental quality values Q_k (Eq. (10)) and to build up the corresponding quality plot, Q_k *versus* V_k, according to Eq. (12). The weights w_i are estimated from the best fit in Eq. (12). For each group we calculated the weighted and un-weighted means of ratios using Eq. (13). These averaged ratios were

collected in histograms presented in Fig. 9. We expect the best estimators to provide distributions centered on the true ratio ($r = 3$) with the least spread around this value.
As expected, the ratio of medians gave a broader distribution for the dust-free image (Fig. 9a). Neither regression filtering nor quality control could improve observed estimates: the histograms of obtained ratios with or without filtering or with or without quality control were indistinguishable in the figure. For the contaminated image (Fig. 9b), ratio of means without filtering or quality control produced an additional peak (red line) reflecting contribution of dust clusters. RFSE estimate eliminates that peak (blue line) and the application of quality weights further improves the estimation (green line). These measures are so efficient that the resulting histogram after regression filtering and quality weighting became almost equivalent to the histogram of the ratios for the dust-free image (Fig. 9c). The ratio of medians is a robust estimate, but less accurate than RFSE. Fig. 9d demonstrates the power of quality control. Linear regression filtering was not applied in this case. The histogram of ratios of means had the same peak of aberrant ratios. Once weights have been applied, the peak disappeared.
Depending on the image, or even on each particular spot, different ratio estimators, such as the ratio of means or ratio of medians, may ensure a better performance; however, in practice it is difficult to predict with confidence the best estimator. RFSE approach gives a unique ratio estimate, which is always comparable to the best of other ratio estimators.

5.3 Robust processing of experimental images

Localization. We tested spot localization algorithm for arrays with different spot sizes, experimental designs and levels of contamination (numerous examples can be found on our web site http://bioinfo.curie.fr/projects/maia/). In all cases the spot localization procedure was carried out automatically with no user intervention. We only supplied the number of blocks in rows and columns and the number of spots in rows and columns within each block when switching from one image to another one. Comparison of the performance of our spot localization algorithm with others can be found in (Novikov & Barillot, 2006a). Although the developed procedure has proved to be very robust with respect to different types of microarray distortions, there is no guarantee that it will perform well for any array. Therefore, interactive tools are available to repair erroneous grids.
Quantification and Quality. We quantified two experimental images (Fig. 10) of different array design and signal-to-noise levels. One image (Fig. 10A) was provided as demonstration example for UCSF Spot 2.0 (downloadable from http://jainlab.ucsf.edu/Downloads.html). It contains 4x4 blocks with 21x21 spots per block, with a spot cell size of about 10 pixels. Cy3 and Cy5 color channels are strongly correlated, with the average correlation coefficient for the spots being about 0.97. Bright contamination spots can be seen irregularly scattered over the array. The magnified image of one such spot is shown in Fig. 5. Each clone was spotted in triplicate. The replicated spots are placed as neighbors in a row. The second image (measured in the Institute Curie, downloadable from http://bioinfo.curie.fr/projects/maia/) contains 12x4 blocks with 15x15 spots per block (Fig. 10B), with a spot cell size of about 30 pixels. The average correlation between the channels in the spots was about 0.85, being somewhat lower than for the first image, although there are no obvious contamination spots. Each clone was prepared in triplicate with the replicated spots put in three vertically distributed sub-arrays.

It is difficult to remain objective while doing comparative study for the experimental images. As the true ratio values are unknown, the only useful measure of quality is the variation in ratio estimates between the replicated spots, which should be reasonably low. Therefore we take the coefficient of variation (Eq. (14)) of the replicates as a quantitative measure of the ratio estimation consistency. However, this measure may not be totally objective: (i) the estimates may be consistent, but systematically biased (the true values of the ratios are unknown); (ii) three replicated spots of very poor quality may give very similar ratio values just by chance (the number of replicates is low). The average over all replicates at the given array coefficient of variation is taken as a global indicator of the Cy5/Cy3 ratio consistency of the array.

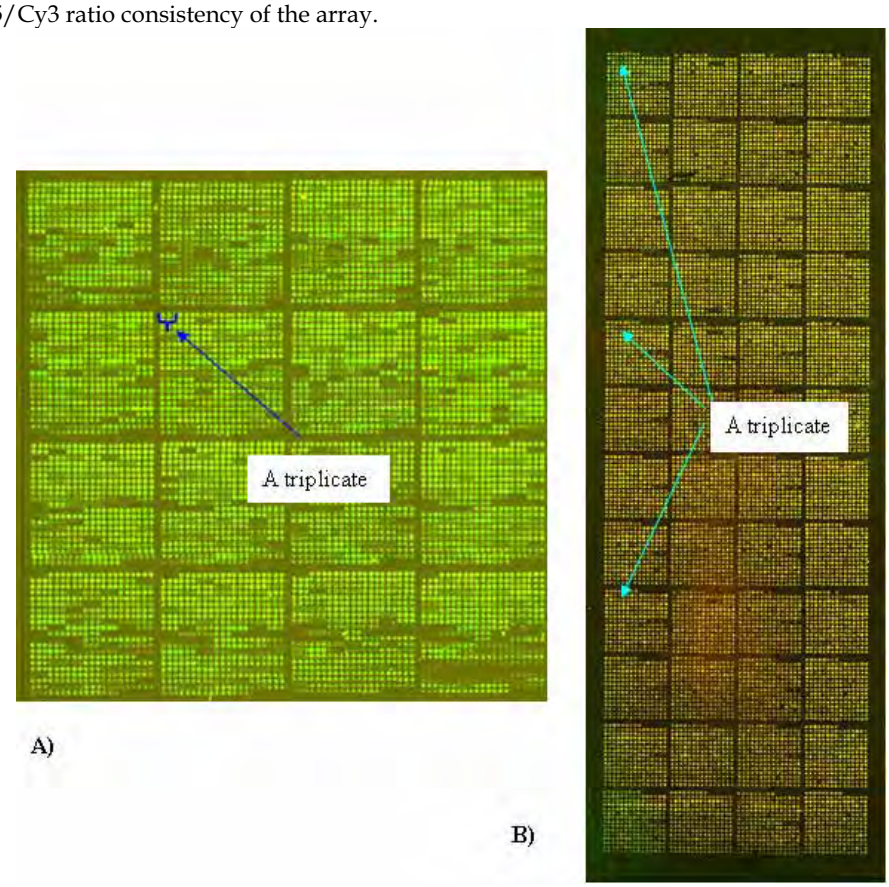


Fig. 10. Experimental images used for evaluation: A) 4x4 blocks with 21x21 spots per block, spot cell size is about 10 pixels; B) 12x4 blocks with 15x15 spots per block, spot cell size is about 30 pixels. The locations of triplicates are indicated.

We compared the averaged coefficient of variation for three ratio estimates (RFSE, ratio of means and ratio of medians) with or without quality control. The weights, w_i, of the marginal quality parameters for Q_k were identified using Eq. (12) with $V \approx 0.07$ for image A, and with $V \approx 0.2$ for image B.

The results are summarized in Table 1. RFSE algorithm ensures the smallest coefficient of variation for both images and quality control improves performance for all three ratio estimates. We found a greater improvement for image A than for image B. This was not a surprise, as image B is characterized by a reasonably high signal-to-noise level, and it does not contain any obvious contaminated spots. However, even in this case the quality measures cannot be ignored, as there are still a few low-intensity spots that need to be specially treated (probably rejected). By contrast, image A has obvious randomly distributed pieces of dust, and the developed filtering procedure (RFSE) and quality measures proved to be powerful enough to repair or to disregard the contaminated spots, thus increasing the consistency of the Cy5/Cy3 ratio estimates. The fact that quality control does not show up much better performance is due to rather good general quality of the images, and a few problematic triplicates cannot influence very much the averaged coefficients of variation. For example, in image A, we have less than 9% of triplicates with the ratio variation coefficients larger that the selected V (~0.08), and 7% for image B ($V \approx 0.2$).

Image	Quality weights	RFSE	Ratio of means	Ratio of medians
A	Without	0.0196	0.0324	0.0410
	With	0.0172	0.0245	0.0381
B	Without	0.119	0.120	0.133
	With	0.108	0.109	0.122

Table 1. The averaged coefficient of variation of the ratio triplicates for two images A and B (see Fig. 10).

Results on comparison of the performance of our quantification approaches with the approaches available from other image analysis packages can be found in (Novikov & Barillot, 2005a; Novikov & Barillot, 2005b).

6. Software

The developed algorithms have been implemented in the MAIA (microarray image analysis) software package (Novikov & Barillot, 2006b). Demonstration version of the software can be downloaded from http://bioinfo.curie.fr/projects/maia/. A full version is freely available to non-commercial users upon request from the authors. The package is written in Java (interface) and C++ (algorithms), and. runs on Windows 95/98/Me/NT/2000/XP platforms (may be used under Unix after recompiling C++ code) and needs the Java Runtime Environment. The whole quantification procedure (including filtering, segmentation and ratio estimation) for one 4Mb image pair (Cy3/Cy5, ~7300 spots; each spot cell is ~10 pixels) takes ~3 sec on 3.00GHz Pentium® 4 CPU with 1 GB of RAM; for a 40Mb image pair (~10800 spots; each spot cell is ~30 pixels) takes up to 20 seconds of processing.

7. Conclusions

In this work we have presented a complete solution for robust, high-throughput, two-color microarray image processing comprising procedures for automatic spot localization, spot quantification and spot quality control.

The spot localization algorithm is fully automatic and robust with respect to deviations from perfect spot alignment and contamination. As an input, it requires only the common array design parameters: number of blocks and number of spots in the x and y directions of the array. Although fully automatic, there is no guarantee that it will perform well for any array. Therefore, we offer some interactive tools to repair grid in case if it is erroneous.

Robust ratio estimation comprises two steps. First, linear regression filtering is used to identify and remove aberrant pixels, and then more traditional segmentation approaches are applied for final estimation. Using the two-step quantification algorithm, we ensure a unique ratio estimate, which is as robust as estimates based on medians and as precise as estimates based on means. Linear regression filtering relies on the fact that the two color channels are expected to be highly correlated. Any contamination, which is uncorrelated in the two channels, can be easily recognized by the algorithm and removed. For noisy (weakly correlated) data, the filter is transparent for the data. Moreover, in this case, linear regression estimates can be biased. Therefore we apply a spot segmentation step to establish the final estimate.

The spot quality algorithm provides a value of spot quality reflecting the level of confidence in the obtained ratio estimate at each spot. The unique spot quality value is derived from a set of ten marginal quality parameters characterizing certain features of the spot. The contribution of each quality parameter in the overall quality is automatically evaluated based on the visual classification of the spots, or using information available from the replicated spots, located on the same array or over a set of replicated arrays. Therefore the developed procedure allows us not only to quantify spot quality, but also to identify different types of spot deficiency occurring in microarray technology. The quality values can be used either directly to flag out some spots with the quality lower than the user-defined threshold, or in the follow-up analysis as a weight controlling the contribution/influence of the obtained ratio estimates.

There are many possibilities to advance the developed algorithms. For example, several spot localization parameters (γ, α and β), that are currently fixed in predefined values, can be iteratively adjusted to achieve the highest regularity of the generated grid. To enhance spot quantification, we can envisage more sensitive (than the single-case diagnostics for the linear regression model) algorithms for aberrant pixel detection. These perspectives are facilitated by further standardizing microarray technology, so that images are becoming more regular, and more specific models for spots and arrays can be developed and justified. As it was shown, different features of the spot (intensity, size, circularity, etc.) can be quantitatively characterized. These characteristics, besides ratios, may contain useful information for the follow-up analysis. One possibility to utilize this information is presented in this paper: we used them to derive spot quality values. However, we believe that more sophisticated analytical tools can be applied to use spot information in other applications. Exploration of these possibilities creates an interesting perspective for future developments.

8. Acknowledgements

We would like to thank our colleagues from the different laboratories of the Institute Curie: (F. Radvanyi, CNRS/IC 144; O. Delattre, INSERM/IC 830; M. Dutreix, CNRS/IC 2027) and Prof. D. Pinkel (UCSF Comprehensive Cancer Center), who have provided numerous microarray images allowing considerable improvement of the algorithms.

9. References

Angulo, J. & Serra, J. (2003). Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, Vol. 19, 553-562.

Atkinson, A. & Riani, M. (2000). *Robust Diagnostic Regression Analysis*, Springer.

Axon Instruments, Inc. (2005). GenePix Pro 6.0. http://www.axon.com, User's Guide and Tutorial.

Bengtsson, A. & Bengtsson, H. (2006). Microarray image analysis: background estimation using quantile and morphological filters. *BMC Bioinformatics*, Vol. 7, 96.

Bozinov, D. & Rahnenführer, J. (2002). Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering, *Bioinformatics*, Vol. 18, 747-756.

Brändle, N.; Bischof, H. & Lapp, H. (2003). Robust DNA microarray image analysis. *Machine Vision and Applications*, Vol. 15, 11-28.

Brown, C.S.; Goodwin, P.C. & Sorger, P.K. (2001). Image metrics in the statistical analysis of DNA microarray data. *Proceedings of the National Academy of Sciences*, Vol. 98, 8944-8949.

Buhler, J.; Ideker, T. & Haynor, D. (2000). Dapple: improved techniques for finding spots on DNA microarrays. *UW CSE Technical Report UWTP 2000-08-05*.

Bylesjö, M.; Eriksson, D.; Sjödin, A.; Sjöström, M.; Jansson, S.; Antti, H. & Trygg, J. (2005). MASQOT: a method for cDNA microarray spot quality control. *BMC Bioinformatics*, Vol. 6, 250.

Ceccarelli, M. & Antoniol, G. (2006). A deformable grid-matching approach for microarray images. *IEEE Transactions on Image Processing*, Vol. 15, 3178-3188.

Chen, Y.; Kamat, V.; Dougherty, E.R.; Bittner, M.L.; Mel*tzer, P.S. & Trent, J.M. (2002). Ratio statistics of gene expression levels and applications to microarray data analysis. *Bioinformatics*, Vol. 18, 1207-1215.

Dissanaike, G. & Wang, S. (2003). A critical examination of orthogonal regression. http://ssrn.com/abstract=407560.

Eckel-Passow, J.E.; Hoering, A.; Therneau, T.M. & Ghobrial I. (2005). Experimental design and analysis of antibody microarrays: applying methods from cDNA arrays. *Cancer Research*, Vol. 65, 2985-2989.

Fisher, L.D. & van Belle, G. (1993). *Biostatistics. A Methodology for the Heath Sciences.* John Willey & Sons.

Glasbey, C.A. & Ghazal, P. (2003). Combinatorial image analysis of DNA microarray features, *Bioinformatics*, Vol. 19, 194-203.

Hautaniemi, S.; Edgren, H.; Vesanen, P.; Wolf, M.; Järvinen, A.K.; Yli-Harja, O.; Astola, J.; Kallioniemi, O. & Monni, O. (2003). A novel strategy for microarray quality control using Bayesian networks. *Bioinformatics*, Vol. 19, 2031-2038.

Hegde, P.; Qi, R.; Abernathy, K.; Gay, C.; Dharap, S.; Gaspard, R.; Hughes, J.E.; Snesrud, E.; Lee, N. & Quackenbush, J. (2000) A concise guide to cDNA microarray analysis. *BioTechniques*, Vol. 29, 548-562.

Herzel, H.; Beule, D.; Kielbasa, S.; Korbel, J.; Sers, C.; Malik, A.; Eickhoff, H.; Lehrach, H. & Schuchhardt, J. (2001) Extracting information from cDNA arrays. *Chaos*, Vol. 11, 98-107.

Ishkanian, A.S.; Malloff, C.A.; Watson, S.K.; DeLeeuw, R.J.; Chi, B.; Coe, B.P.; Snijders, A.; Albertson, D.G.; Pinkel, D.; Marra, M.A.; Ling, V.; MacAulay, C. & Lam, W.L. (2004). A tiling resolution DNA microarray with complete coverage of the human genome. *Nature Genetics*, Vol. 36, 299–303.

Jain, A.N.; Tokuyasu, T.A.; Snijders, A.M.; Segraves, R.; Albertson, D.G. & Pinkel, D. (2002). Fully automated quantification of microarray image data. *Genome Research*, Vol. 12, 325-332.

Kendall, M.G. & Stuart, A. (2003). *The Advanced Theory of Statistics,* Vol. 2, McMillan, 1979.

Lehmussola, A.; Ruusuvuori, P. & Yli-Harja, O. (2006). Evaluating the performance of microarray segmentation algorithms. Bioinformatics, Vol. 22, 2910-2917.

Novikov, E. & Barillot, E. (2005a). An algorithm for automatic evaluation of the spot quality in two-color DNA microarray experiments. *BMC Bioinformatics*, Vol. 6, 293.

Novikov, E. & Barillot, E. (2005b) A robust algorithm for ratio estimation in two-color microarray experiments. *Journal of Bioinformatics and Computational Biology*, Vol. 3, 1411-1428.

Novikov, E. & Barillot, E. (2006a). A noise-resistant algorithm for grid finding in microarray image analysis. *Machine Vision and Applications*, Vol. 17, 337-345.

Novikov, E. & Barillot, E. (2006b). Software package for automatic microarray image analysis (MAIA). *Bioinformatics*, Vol. 23, 639-640.

Pinkel, D.; Segraves, R.; Sudar, D.; Clark, S.; Poole, I.; Kowbel, D.; Collins, C.; Kuo, W.L.; Chen, C.; Zhai, Y.; Dairkee, S.H.; Ljung, B.M.; Gray, J.W. & Albertson, D.G. (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics*, Vol. 20, 207–211.

Ritchie, M.E.; Diyagama, D.; Neilson, J.; van Laar, R.; Dobrovic, A.; Holloway, A. & Smyth, G.K. (2006). Empirical array quality weights in the analysis of microarray data. *BMC Bioinformatics*, Vol 7, 261.

Rousseeuw, P.J. & Leroy, A.M. (2003). *Robust Regression and Outlier Detection*, John Willey & Sons.

Rueda, L. & Vidyadharan, V. (2006). A hill-climbing approach for automatic gridding of cDNA microarray images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 3, 72-83.

Wang, X.; Ghosh, S. & Guo, S.W. (2001). Quantitative quality control in microarray image processing and data acquisition. *Nucleic Acids Research*, Vol. 29, e75.

Yang, Y.H.; Buckley, M.J.; Dudoit, S. & Speed, T.P. (2002). Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistics*, Vol. 11, 108-136.

12

Computer Vision for Microscopy Applications

Nikita Orlov, Josiah Johnston, Tomasz Macura, Lior Shamir, Ilya Goldberg
Laboratory of Genetics, National Institute on Aging/NIH
USA

1. Introduction

The tremendous growth in digital imagery has introduced the need for accurate image analysis and classification. The applications include content based image retrieval in the World Wide Web and digital libraries (Dong & Yang, 2002; Heidmann, 2005; Smeulders et al., 2000; Veltkamp et al., 2001) scene classification (Huang et al., 2005; Jiebo et al., 2005), face recognition (Jing & Zhang, 2006; Pentland & Choudhury, 2000; Shen & Bai, 2006) and biological and medical image classification (Awate et al., 2006; Boland & Murphy, 2001; Cocosco et al., 2004; Ranzato et al., 2007). Although attracting considerable attention in the past few years, image classification is still considered a challenging problem in machine learning due to the complexity of real-life images. This chapter discusses an approach to computer vision using automated image classification and similarity measurement based on a large set of general image descriptors. Classification results as well as image similarity measurements are presented for several diverse applications.

1.1. Image classification and computer vision

Image analysis can be partitioned into two major approaches. In one, it is assumed that the image is of something that can be modeled a-priori and recognized within the image. This approach uses a model of the subject to drive segmentation followed by extraction of features from the segmented data that correspond to model parameters (size, shape, intensity, distribution, etc). This approach lends itself very well to using imaging for quantitatively measuring defined aspects of a pre-conceived model (Dong & Yang, 2002; Huang et al., 2005; Smeulders et al., 2000). However, a model is not always available, or when available is not always easily reconciled with image data or is not readily useable to extract the relevant subject out of the image (i.e. segmentation). In many applications of image processing, the observed parameters of a model are used to answer questions about the degree to which a particular observation differs from previous observations (i.e. image similarity), or the degree to which an observation agrees with several alternative models (i.e. image classification). Thus an alternative to model-based image analysis for the purposes of computing image similarity or classification is to use pattern recognition and supervised machine learning to answer these questions directly. The focus of this chapter is an approach where a model of what is imaged is built up out of examples consisting of training images rather than an image-independent pre-conceived notion of what is being imaged.

Although the image plane is the carrier of various patterns, the pixels themselves are not normally used directly as inputs to machine learning algorithms. Instead, image content is derived through computation of numerical values that represent quantitative measures of various pixel patterns (Gurevich & Koryabkina, 2006; Heidmann, 2005). These numerical features of the image are based on different algorithms that extract a wide variety of patterns present in the image, such as edges, color (Funt & Finlayson, 1995; Stricker & Orengo, 1995; Tieu & Viola, 2004), textures (Ferro & Warner, 2002; Livens et al., 1996; Smith & Chang, 1994; Smith & Chang, 1996), shapes (Mohanty et al., 2005), histograms (Chapelle et al., 1999; Flickner et al., 1995; Qiu et al., 2004), etc.

Biological microscopy is an emerging application for pattern recognition that presents many diverse problems and image modalitites (Awate et al., 2006; Boland et al., 1998; Boland & Murphy, 2001; Duller et al., 1999; Murphy, 2004; Orlov et al., 2006; Ranzato et al., 2007; Rodenacker & Bengtsson, 2003; Swedlow et al., 2003). When pattern recognition has been used, the tendency is to tailor the image descriptors as well as the classification algorithm to a specific type of imaging problem. Biological microscopy can produce images of many kinds ranging from structural studies of sub-cellular compartments, to the morphology of cells, to tissues and entire organisms. Methods for generating contrast (i.e. the imaging techniques) vary as much as the scale – from fluorescently-tagged protein-specific probes, to various colorimetric stains, to the differential scattering properties of molecular structures. For these reasons, there is no typical imaging problem in biological microscopy and therefore no typical set of image content descriptors. The very nature of the application field requires using a broad variety of algorithms for describing relevant image content (Awate et al., 2006; Boland et al., 1998; Boland & Murphy, 2001; Cocosco et al., 2004; Livens et al., 1996; Ranzato et al., 2007).

A growing demand in pharmaceutical as well as basic research is the use of high-throughput image analysis to score High Content Screens (HCS). In these experiments, a large bank of manipulations (tens of thousands of genes or chemical compounds) is applied one by one to cells grown under defined conditions. Generally the screen is a hunt for genes or compounds that mimic a particular cellular response that can be pre-arranged using positive controls. These screens are typically highly automated using robots for plate and liquid handling, as well as image acquisition. The variety of possible visual assays, combined with the very high demands on the robustness of the processing algorithms makes image analysis in these types of screens the primary bottleneck.

Rodenacker and Bengtsson (Rodenacker & Bengtsson, 2003) have surveyed a large collection of content descriptors for the analysis of grayscale microscopy images. They differentiated feature types into four major categories: intensity, size and shape, texture, and structure. Their suggested scheme for computing signatures includes two pre-processing steps, segmentation (selection of ROIs) and transforms. The use of image transforms is seen as an essential part of feature extraction, where the next-order extraction algorithms (histograms and others) would operate on transforms to produce feature vectors. Many of the feature algorithms given by Rodenacker and Bengtsson could also be used without prior segmentation, and are applicable outside of biological microscopy. The number of descriptors discussed in the paper is quite large, so the authors provide suggestions about which features they found most useful and recommend avoiding textural and structural features for data with strong variation in size and intensity. For feature selection, they recommend hand picking features instead of using independent statistical methods.

Manual feature selection relies on considerable expertise, because of its dependence on the specifics of the experiment as well as the preprocessing steps used.
Lehmann *et al.* (Lehmann et al., 2005) have developed an automated system for categorization and retrieval of images in a medical context. The system includes feature computation and selection as well as classification based on supervised learning using a k-NN algorithm. The set of descriptors they used was limited to Tamura textures as well as several other texture-based descriptors applicable in this domain. The sensitivity of the system was quite good, being able to distinguish 81 distinct categories.
Gurevich and Koryabkina (Gurevich & Koryabkina, 2006) undertook probably the most ambitious survey of existing image descriptors. They developed and adopted from the literature a broad range of features and classified them by scope, method, purpose, etc. While the authors made suggestions of applicability of descriptor types to specific domains, no automated mechanism of feature selection was proposed.

1.2. Digital images: properties and meaning

A given image is not merely an undifferentiated 'bag of features'. The meaning of the image, or the relevant information it contains, can be derived from these features only once their relative importance is determined in a specific context. In supervised machine learning, context is determined by associating a given image with others in a class. The set of classes and the example images they are comprised of defines the context of a specific imaging problem. A given image may be viewed in different contexts by associating it with different groups, which results in a different relative importance of the features, and consequently different interpretations of the image's meaning.
Typical approaches to machine learning emphasize optimizing classification in just one particular problem. Because of this, typical implementations of pattern recognition algorithms only allow for a limited set of descriptors (Awate et al., 2006; Boland & Murphy, 2001; Cocosco et al., 2004; Dong & Yang, 2002; Jing & Zhang, 2006; Ranzato et al., 2007; Rosenfeld, 2001; Shen & Bai, 2006; Smeulders et al., 2000). A limited number of features is desirable because it lowers the computational cost, and reduces the dimensionality of the feature space used in classification. The features selected and their relative weights are problem-specific. The feature set can become inapplicable when new images deviate significantly from those the classifier was trained on, or if they are from a different imaging modality.
A general computer-vision approach requires an alternative to task-specific or manual feature selection (Rosenfeld, 2001). It should use a large feature set in an application-specific context to automatically pick patterns crucial for the given recognition problem (Fig. 1)(Felsenstein, 1989).

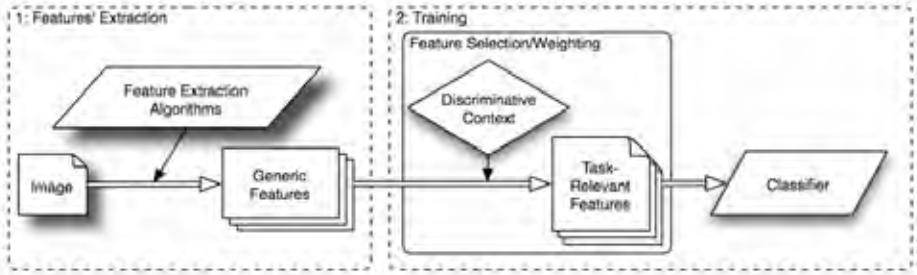


Fig. 1. Image classification scheme. Panel 1: feature extraction; panel 2: feature selection and pattern recognition.

Two antagonistic principles play important roles: context-independent and context-dependent. On the one hand, the system must not ignore details capable of discriminating patterns, which requires having a comprehensive set of context-independent descriptors. On the other hand, irrelevant information (weak features) should be discarded depending on the image context. A general approach to computer vision must balance these two principles. Automating the selection and weighing of features prevents the introduction of anthropogenic bias, which combined with a comprehensive set of descriptors, leads to both generality and objectivity.

Three alternative pathways are available for introducing discriminative context into the initial feature set. The first is to evaluate the discriminative power for all features of the initial set independently of a classifier, keeping features with the highest classification power and discarding the rest. Examples of classifier-independent feature reduction include principal component analysis (PCA) and linear discriminant analysis (LDA, e.g. Fisher discriminant). The second approach is to combine feature selection with classifier building and training, resulting in a feature subset concurrently with the classifier itself. Lastly, there are classifiers capable of performing in high dimensionality feature spaces, essentially by being tolerant of many features with low weights. These include weighted nearest neighbor methods (Parades & Vidal, 2006; Ricci & Acesani, 1999), though these have not been evaluated in feature spaces higher than a few hundred dimensions.

The variety of images available from biological microscopy sets it apart from typical pattern classification problems. This motivated taking a broader look at the principles of image descriptors to measure a wider variety of image content. Because of the generality afforded by addressing the field of biological microscopy as a whole, this approach also proved effective in completely unrelated fields.

1.3. Chapter outline

This chapter describes an approach to compile and recombine traditional and new image analysis algorithms into a general-purpose hyper-dimensional feature set, and the use of an automated feature selection and training method to reduce this feature bank to a context-dependent subset. This effort resulted in a multi-purpose image classifier that can be applied to a variety of image classification problems.

Section 2 introduces the algorithms used in the feature extraction scheme, and Section 3 describes feature reduction and classifier training. Section 4 presents classification results on a set of diverse image types, and Section 5 discusses techniques for computing context-

specific image similarity. Section 6 presents a computing framework used to calculate the features described in Section 2.

2. Extraction of Image Features

Features fall into four categories: polynomial decompositions, textures, object descriptors and pixel statistics. In the first category, a polynomial approximation of the image is computed, and the polynomial coefficients serve as image features. Three kinds of polynomials are computed: Zernike, Chebyshev and Chebyshev-Fourier. Texture features report inter-pixel variation in intensity for several directions and resolutions. These include Haralick textures, as well as Tamura features and Gabor filters. Object features are calculated from one or more object identification algorithms and comprise statistics about object number, spatial distribution, size, shape, etc. Pixel statistics consist of multi-scale intensity histograms, edge statistics, radon histograms and comb-4 moments. In addition to calculating these features on the original image pixels, they are also calculated on several image transforms (Fourier, wavelet, Chebyshev), as well as transform combinations (see Fig. 2).

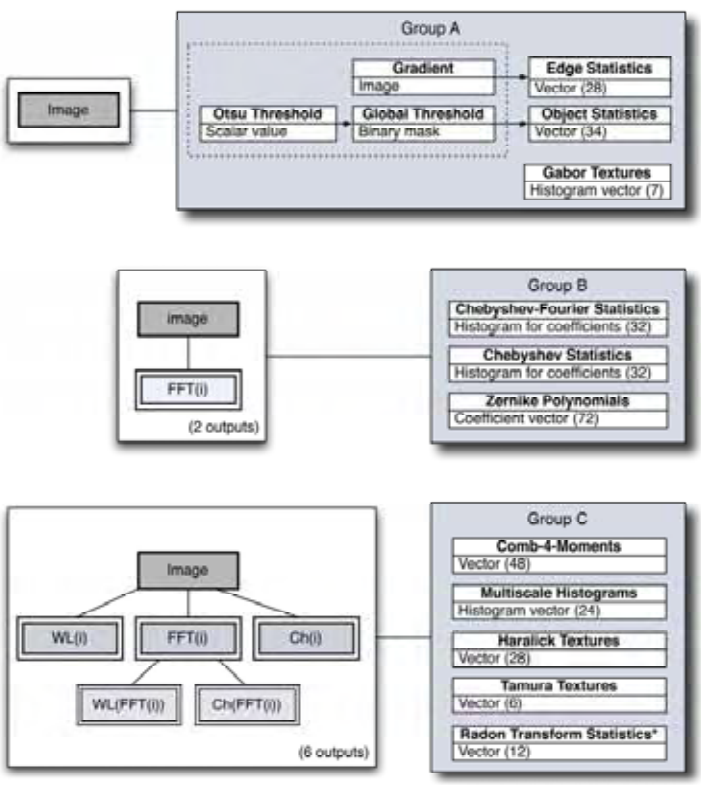


Fig. 2. Proposed scheme for feature extraction. Features are extracted from the original image and several transforms. The Radon Transform Statistics are not computed from the wavelet transform.

Figure 2 illustrates a feature bank (image descriptors) composed of 1025 variables from 11 algorithms, three transforms, and two transform combinations. Each feature measures a different aspect of image content though they cannot be considered strictly orthogonal or independent of each other. All features are based on grayscale images, so that color information is not currently used. Also, though these features cover a broad spectrum of image content, this set cannot be considered to be complete.

2.1. Transforms

In general, transforms allow feature extraction algorithms to be reused to measure very different image content than using them on the original pixels. This algorithm reuse leads to a large expansion of the feature space with a corresponding increase in the variety of image content that can be measured. Each of the three transforms results in a 2D matrix of the same size as the original image pixels; features are computed on image transforms in the same manner as they are on the original images. The Fourier transform is a standard implementation (FFTW), where only the absolute value of the complex transform is used. For the wavelet transform, the standard MATLAB Wavelet toolbox functions 'wavedec2.m' and 'detcoef2.m' were used to compute coefficients for a two-dimensional wavelet decomposition of the image. The Chebyshev transform was implemented by our group and is described in section 2.3 below.

2.2. Pre-processing and color images.

Image pre-processing is a common way to limit noise and improve classification (Hoggar, 2006). Pre-processing is quite common as a prelude for model-based segmentation, but is often unnecessary for the type of scene-based pattern recognition presented in this chapter. All but one of the examples presented in Section 5 were classified without preprocessing. In the example of age-related degeneration of the body-wall muscle, the muscle fibers of the worm's body wall contain significant contributions from the worm's internal structures which were irrelevant for this study. Because the fibers make a regular repeating pattern, a Hamming filter (Hamming, 1989) was utilized to dampen contributions from these less regular structures (see Figure 3).

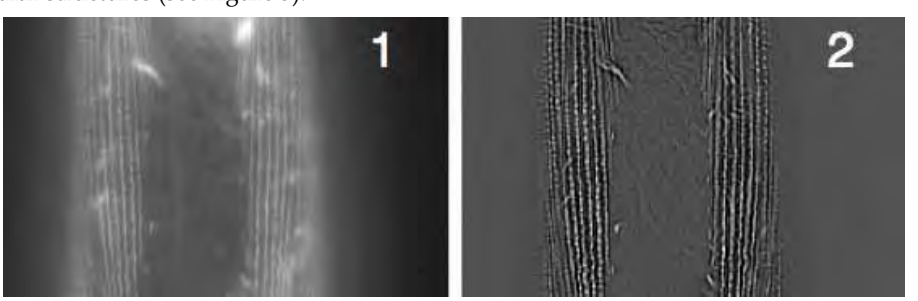


Fig. 3. Image pre-processing. 1: original image (contains irrelevant internal structures); 2: filtered image allows concentrating on morphology of body wall muscle.

Color images also require preprocessing because the feature bank operates on grayscale images. Color information is commonly expressed as several separate color planes in RGB or some other color space. In some cases, the color information is superfluous for classification purposes, and the planes can be combined into a single gray-scale image using

the color's luminance from the NTSC video conversion formula (rgb2gray in MATLAB). In pathology, tissue biopsies are often stained with a pair of dyes called Hematoxylin and Eosin (H&E stain), producing purple cell nuclei with other structures in varying shades of pink. These are normally imaged with an RGB camera, which convolves the H&E channels into an RGB space which is dependent on the camera's spectral response. There are three ways to overcome these difficulties. First, one can compute features on R, G and B channels separately using the existing scheme, with the drawback of treating the 3 channels as independent entities even though each channel is a convolution of H&E. Second, one can use a color deconvolution algorithm (Ruifrok & Johnston, 2001) or similar techniques to approximate the original 2D color space and then use the feature bank on the resulting H&E channels. Lastly, one can introduce feature extraction algorithms that specifically measure color information (e.g. color histograms).

2.3. Chebyshev transform and related features

Chebyshev polynomials (Gradshtein & Ryzhik, 1994)

$$T_n(x) = \cos(n \arccos(x)) \tag{1}$$

are widely used for approximation purposes. For any given smooth function one can generate its Chebyshev approximant, like

$$f(x) \cong \sum_{n=0}^{N} \alpha_n T_n(x). \tag{2}$$

Chebyshev polynomials are orthogonal (with a weight (Gradshtein & Ryzhik, 1994)); therefore, the expansion coefficients α_n can be expressed via the scalar product:

$$\alpha_n = \langle f(x), T_n(x) \rangle. \tag{3}$$

By analogy with Fourier space (formed by the transform coefficients), one can consider the collection of coefficients $\{\alpha_n\}$ as members of some spectral space – the Chebyshev space. Similarly to the 1D case (2), for a given image I_{ij} its two-dimensional approximation through the Chebyshev polynomials is

$$I_{ij} = I(x_i, y_j) \cong \sum_{n,m=0}^{N} \alpha_{nm}\, T_n(x_i)\, T_m(y_j). \tag{4}$$

The fast algorithm was used in the transform computation; it takes two sequential 1D transforms, first for rows of the image, then for columns of the resulting matrix (similarly to the implementation of 2D FFT).

The Chebyshev transform is designed to characterize all ranges of the image spectral domain – from low to high frequency features. The idea is to retain a finite number of expansion terms, with the expansion coefficients being used as image descriptors. Chebyshev is used both as a transform (with orders matching image dimensions) and as a set of statistics. The maximum transform order does not exceed N = 20, so that the resulting coefficient vector has dimensions (1x400). The feature vector is produced from the coefficients by applying a 32-bin histogram.

2.4. Features based on Chebyshev-Fourier transform

This 2D transform is defined in polar coordinates, and it uses two different kinds of orthogonal transforms for its two variables: the radial coordinate of the image is approximated with Chebyshev polynomials, and Fourier harmonics are used for the azimuth variable:

$$\Omega_{nm}(r,\phi) = T_n(2r/R-1)\times\exp(im\phi), \quad 0 \le r \le R\,. \tag{5}$$

In this sense it shares similarities with Zernike transform where the power polynomials are used in radial direction, and harmonic functions for the angle. For the given image (I_{ij}) the transform generates an image approximant in the form

$$I_{ij} \Rightarrow I(r_k,\phi_l) \cong \sum_{m=-N/2}^{N/2}\sum_{n=0}^{N}\beta_{nm}\,\Omega_{nm}(r_k,\phi_l)\,. \tag{6}$$

In the presented descriptor system (Fig. 2), features based on coefficients β_{nm} of the Chebyshev-Fourier transform capture low-frequency components of the image content (large-scale areas with smooth intensity transitions). The highest order of polynomial used is N = 23, and the coefficient vector is then reduced by binning to 1x32 length.

2.5. Features based on Gabor wavelets

Gabor wavelets (Gabor, 1946) are used to construct spectral filters for segmentation or detection of certain image texture and periodicity characteristics. Gabor filters are based on Gabor wavelets, and the Gabor transform of an image $I(x,y)$ is defined as

$$GT(x,y;f) = \iint_{W_X,W_Y} I(x-w_X, y-w_Y)\,G(w_X,w_Y;f)\,dw_X\,dw_Y \tag{7}$$

where the kernel $G(w_X,w_Y;f)$ often takes the form (Gregorescu et al., 2002) of a convolution of the Gaussian with the harmonic function:

$$\begin{aligned} &G(w_X,w_Y;f_0) = \exp\{-(X^2+\gamma Y^2)/2\sigma^2\}\times\exp\{j(f_0X+\phi)\},\\ &\begin{cases} X = w_X\cos\theta + w_Y\sin\theta \\ Y = -w_X\sin\theta + w_Y\cos\theta \end{cases} \end{aligned} \tag{8}$$

The parameters of the transform are rotation (θ), ellipticity (γ), frequency (f_X, being related to the wavelength) and σ (related to the bandwidth). The parameters γ and σ were chosen to be $\gamma = 0.5, \sigma = 0.56\times 2\pi/f_X$ (Gregorescu et al., 2002). In the feature bank (see Fig. 2) the Gabor Features (GF) are defined as an area occupied by the Gabor-transformed (GT) image:

$$GF(f_X) = \frac{1}{G_L}\iint_{x,y:GT>0} GT(x,y;f_X)\,dx\,dy \approx \frac{1}{G_L}\sum_{i,j:GT>0} GT_{ij}(f_X), \quad GT_{ij}(f_X) = GT(x_j,y_i;f_X)\,. \tag{9}$$

To minimize frequency bias, these features are computed in a frequency range ($f_X \in \{f_k\}_{k=1}^{K}$), and normalized with the low frequency component $G_L = GF(f_L)$. The frequency values used were $f_L = 0.1$ and $f_X = [1\ 2 \ldots 7]$.
In the feature bank in Figure 2, the Gabor features belong to a group of textural descriptors and measure image content corresponding to the high and highest spectral frequencies, especially grid-like image textures.

2.6. Radon transform based features

The Radon transform computes a projection of pixel intensity onto a radial line from the image center at a specified angle (radon.m is a built in MATLAB function). The transform is typically used for extracting spatial information where pixels are correlated to a specific direction or angle. The Radon feature computes a series of Radon transforms for angles 0, 45, 90, and 135, and then convolves each transform into a 3-bin histogram; the resulting vector therefore totals 12 entries.

2.7. Multi-scale histograms

This set of features computes histograms with varying numbers of bins (3, 5, 7, and 9) (Hadjidementriou et al., 2001). Each frequency range best corresponds to a different histogram, and thus variable binning allows measuring content in a large frequency range. The maximum number of counts is used to normalize the resulting feature vector, which is 1x24 elements.

2.8. Four-way oriented filters for first four moments

For this set of features, the image is subdivided into a set of stripes in four different orientations (0°, 90°, +45° and -45°). The first four moments (mean, variance, skewness, and kurtosis) are computed for each "stripe", and each set of stripes is sampled as a 3-bin histogram. Four moments in four directions with 3-bin histograms results in a 48-element feature vector.

2.9. Tamura features

Three basic textural properties of an image – contrast, coarseness, and directionality– were proposed by Tamura in 1978 (Tamura et al.). We used these definitions as they were given in Tamura's paper and coded them without modifications. Coarseness gave 4 values (1 for total coarseness and 3 for the histogram), directionality and contrast each contributed 1 entry, totaling 6 features for this group.

2.10. Edge, Zernike and Haralick features

These features were computed as described in (Murphy et al., 2001). Briefly, Edge features measure several statistics on the image's Prewitt gradient. Zernike features are the coefficients of the Zernike polynomial approximation of the image. Haralick features are statistics computed on the image's co-occurrence matrix.

2.11. Object Statistics

Object statistics are calculated from a binary mask of the image resulting from applying a global threshold using Otsu's method. Thirty-four basic statistics about the segmented objects are extracted with MATLAB's 'regionprops.m' function. The statistics include: number of objects, "Euler Number" (the number of objects in the region minus the number of holes in those objects), and image centroid (x and y). Additionally, minimum, maximum, mean, median, variance, and a 10-bin histogram are calculated on both the objects' areas and distances from objects' centroids to the image centroid.

3. Feature Evaluation and Training

Feature extraction is followed by evaluation of each feature's classification power in a given training context. Many classification algorithms such as neural networks, Bayesian belief networks, Markov chain networks, support vector machines, etc, operate in low-dimensional space and neither need nor can function with extraneous or irrelevant features (Bishop, 1996). The problem of mismatch between number of features used for image description and the number useable by these classifiers is often referred to as the 'curse of dimensionality' (Bishop, 1996). Common techniques for reducing dimensionality include Fisher Discriminant and Linear Discriminant Analysis, as well as Principal Component Analysis, Independent Component Analysis, etc. (Bishop, 1996; Fukunada, 1990; Jain & Zongker, 1997; Kudo & Sklansky, 2000; Yang & Wu, 2004; Yu & Yang, 2001). Dimensionality reduction is still considered an active research topic.

Having a large collection of features implies that for every particular classification problem a majority of the features will be sensitive to irrelevant image content and therefore represent noise. Such features unnecessary add to computational complexity and degrade the performance of the classifier when a finite number of training samples are used (Kudo & Sklansky, 2000). Including non-representative features in a classification problem can also lead to over-training with a resulting loss of predictive power.

Dimensionality reduction is a form of hard thresholding, where features below a certain classification power are completely eliminated from subsequent training and classification. An alternative approach (soft thresholding) can be realized with a family of weighting algorithms (Parades & Vidal, 2006; Ricci & Acesani, 1999). The examples presented in Section 4 use a hard thresholding approach which is discussed below.

One way to evaluate the expected usefulness (i.e. discriminative power) of each feature is by computing its ability to separate data between classes while minimizing its within-class variation. This scoring is based on the Fisher linear discriminant which can be formulated as follows (Fukunada, 1990)

$$F = S_B / S_W\,, \quad S_B = \underset{c=1..C}{\text{mean}} \left(\mu - \mu_c\right)^2, \quad S_W = \underset{c=1..C}{\text{mean}}\ \sigma_c^2 \tag{10}$$

where S_B is the variance of class means from the pooled mean, S_W is the mean of within-class variances, μ_c is the mean of class c, μ is the pooled mean (i.e. mean of all samples), and σ_c^2 is the variance of class c.

The first round of dimensionality reduction calculates the Fisher score separately for each feature, and eliminates 80% of the features with the lowest scores. The second round of

dimensionality reduction tests each feature's discriminative power in a classifier context and iteratively builds the classifier using a greedy-hill climbing algorithm. Naïve Bayesian networks were chosen because they are quickly trained and do not require optimizing the network topology when adding new features to a growing network. In the initial pass, a single-node Bayesian network is constructed from each of the features, and its classification performance and predictive power is scored. To score each network, the initial set of training images is split into a training and test set multiple times, and the classification performance is averaged for all splits. Typically, 35% of the images are reserved for testing in each split, and 35 splits are averaged together for the network score. These parameters have not been systematically calibrated, but proved effective for the datasets evaluated. The feature producing the highest-scoring network is retained for the second pass. In subsequent passes, each of the remaining features is added to the network one by one and the network score is determined. These passes continue until the network score no longer improves by the addition of new features. Generally the number of features selected depends on the number of classes in a classification problem, yielding classifiers with about as many nodes/features as there are classes. This also varies to some extent depending on the separability of the image classes.

4. Experimental Results

In order to evaluate the efficiency of the proposed approach four distinct image data sets were chosen based on application diversity. This small diverse set includes images of fluorescently-labeled cells grown in culture, optical sections of aerogel used to characterize particle traces in the Stardust space probe experiment, fluorescently-labeled muscle tissue, and an organ imaged using differential interference contrast.

The first pair of examples illustrates the effectiveness of the classifier using two very different image types that are particularly challenging for segmentation-based image analysis. The first example (Figure 4) is from an HCS experiment looking for genes that disrupt the formation of fringes or "ruffles" surrounding cells. These ruffles are thought to be important for cell migration, and consequently for the invasiveness of cancer cells as well as developmental abnormalities. The images tested are the control images for the screen, where absence of ruffling is induced by a gene known to have this effect, and these images are contrasted with those of normal cells that display the ruffling. The complete experiment would look for additional genes from a 35,000-gene library that mimic the "no ruffling" morphology (i.e. appear similar to the "no ruffling" control). Various aspects of the images are irrelevant for the purposes of the screen – the cell density and distribution, the overall intensity of the images, etc. These irrelevant variations are included in the set of control images for the screen, and are therefore averaged out in the course of training the classifier. It is not immediately obvious what sorts of image descriptors one would manually chose to differentiate these two classes. Additionally, these images do not appear to be amenable to segmentation-based approaches. Instead, a data driven approach of using a training set to identify relevant features from a large diverse feature set effectively addresses this type of imaging problem (see Table 1).

The second application (Figure 5) involves control images from the Stardust comet dust project. In control experiments, iron grains were shot into aerogel using a dust accelerator. Images were generated by collecting optical sections of aerogel on a microscope. The goal of this classification task is to differentiate images that contain tracks left by dust particles (see

Figure 5, box) from images lacking these tracks. This problem is complicated because all images contain artifacts. While this type of problem can probably be easily addressed with a segmentation algorithm and a discriminator based on the form factor of the segmented objects, this would take significant manual tinkering. However, the same algorithms and parameters used in the previous example worked well without requiring any new software or manual parameter adjustment to work equally well on this very different imaging problem.

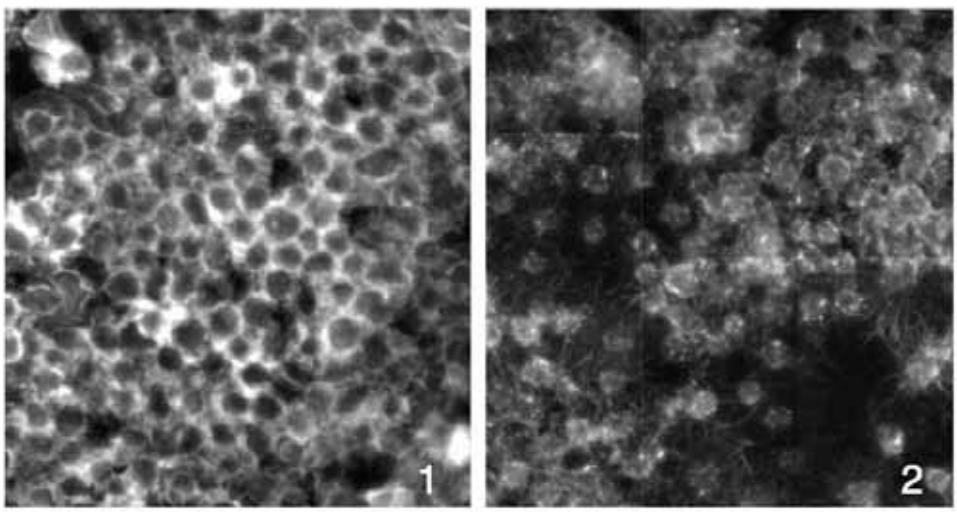


Fig. 4. Images of Ruffling phenotype. Panel 1 shows a field of cells exhibiting the normal ruffling behavior. Panel 2 shows the effect of a knocking down the expression of a gene known to be required for the ruffling phenotype.

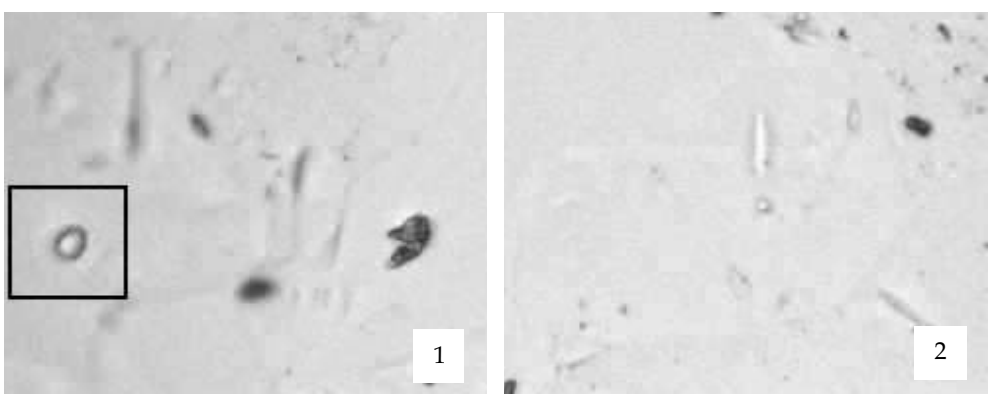


Fig. 5. Images of aerogel from the Stardust project. Panel 1 shows an image containing both a track caused by a dust particle (surrounded by the black frame) and artifacts. Panel 2 shows an image containing only artifacts.

Classification accuracy for these two problems is shown in Table 1. Accuracy is defined as the ratio of correct class assignments to the number of test samples.

Data set	Accuracy
Ruffling cells	0.92
Stardust	0.87

Table 1. Classification results for Ruffling cells and Stardust data sets. These accuracies are averages of 5 separate divisions of training and test images. Ruffling dataset had 616 training and 2648 test images while the Stardust dataset had 2072 training and 1358 test images.

It is of considerable medical interest to identify compounds and genes that affect the aging process. The effects of these compounds or genetic mutations can be quantified if a morphological age can be computed independently of temporal age. *C. elegans* is a small earth-dwelling worm used as a model organism in genetic, development, aging and behavioral studies. Its transparency to visible light, short lifespan, and its well characterized development and genetics make it an ideal organism for aging studies. In the next two studies, the goal is to determine a morphological age based on a microscope image of the worm.

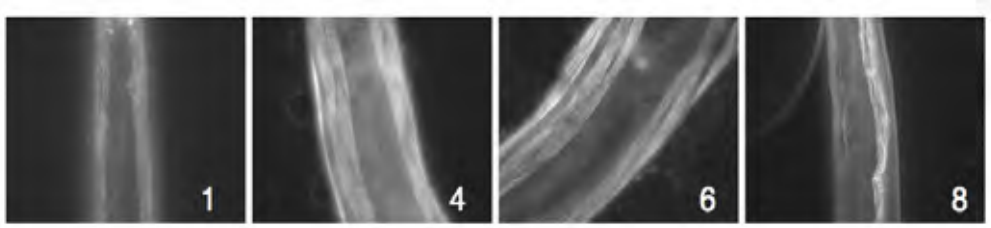


Fig. 6. Aging of body wall muscle in *C. elegans*. Progressive muscle degeneration in the body wall muscle corresponds to 1, 4, 6, and 8 days after molting.

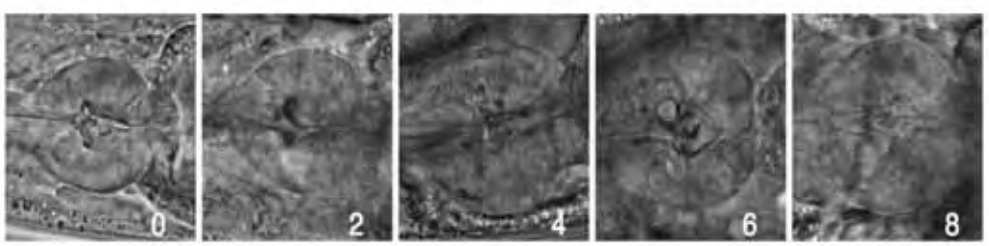


Fig. 7. Aging of pharynx terminal bulb in *C. elegans*. Progressive degeneration of this organ corresponds to 0, 2, 4, 6, and 8 days after molting.

These two studies illustrate a general problem of using images to determine the degree of progression through a continuous morphological process. A distinguishing characteristic of

this type of problem is that there can be considerable uncertainty about the "ground truth" with regard to individual images. On average, the images in a class belong to a known stage along this process, but the exact stage of individual images within the class is far less certain. The individual images in a class may not be precisely synchronized, and the morphological process itself can be subject to substantial stochastic effects. In the specific case presented here, it is known that individual worms can be synchronized at best to within 4 hours. Additionally, individual variability is easily seen in the images collected on a given day, and it is known that a synchronized genetically identical population does not die synchronously, but over a span of 7 to 15 days, which indicates a further loss of synchrony during the process. This supports the prediction that individual images will not be readily classified, and that there will be considerable spill-over between adjacent classes. This would not be a problem in an actual experiment because the treatment would be applied to many individuals, allowing for the use of averaging to evaluate its effect.

These two applications differ in the aging effect being assayed and in the types of images collected. In the first study (see Figure 6), the actin fibers of muscle cells throughout the worm are fluorescently labeled and imaged on a fluorescence microscope. In the second study (see Figure 7), a non-invasive imaging technique that does not require stains or dyes for contrast development (differential interference contrast – DIC) is used to visualize a part of the worm's eating apparatus – the pharynx terminal bulb.

Classification accuracy is not a relevant measure of classifier performance for these two applications because the goal is to measure change rather than class assignment. Instead, the relevant measure is the correlation between the known and computed ages of the classes. The age of each class is computed by averaging the computed ages of its member images. Formula (12) defines an age metric (m) for an individual image in terms of a weighted sum of known class ages (a_c). Weights are the probability (p_c) reported by the Naïve Bayesian classifier of the image belonging to a particular class (c)..

$$m = \sum p_c a_c \,, \tag{12}$$

The averaged marginal probabilities for each class are reported in Tables 2 and 3 for body wall muscle and pharynx respectively. The corresponding correlation factors were 0.95 and 0.88 for muscle and pharynx images respectively.

Test Data	Day 1	Day 4	Day 6	Day 8
Day 1	0.39	0.27	0.22	0.12
Day 4	0.24	0.47	0.15	0.14
Day 6	0.24	0.15	0.39	0.22
Day 8	0.12	0.15	0.25	0.48

Table2.Confusion matrix of averaged marginal probabilities for body wall muscle.

Test Data	Day 0	Day 2	Day 4	Day 6	Day 8
Day 0	0.95	0.03	0.01	0	0.01
Day 2	0.04	0.48	0.21	0.14	0.13
Day 4	0.01	0.25	0.29	0.22	0.23
Day 6	0.00	0.14	0.24	0.35	0.27
Day 8	0.01	0.16	0.24	0.29	0.30

Table 3. Confusion matrix of averaged marginal probabilities for pharynx.

From inspecting the marginal probabilities in tables 2 and 3, it is clear that there is considerable spill-over to neighbouring classes. However, the probabilities on the diagonal are still strongest, which indicates that accurate classification can still be achieved if a sufficient number of individual images are averaged together.

5. Image Similarity

The last two experiments in the previous section established a baseline of the aging process from which to evaluate effects of experimental manipulations. Aging, being a continuous morphological process, presents a useful means of validating image similarity measures. Simultaneously, image similarity can refine our understanding of aging by answering questions like "Are there distinct stages in this morphological process?" Additionally, reliable image similarity measures are necessary for quantifying class separability, identifying new morphological clusters, or using image queries for content-based image retrieval.

Image similarity is most commonly computed as a distance between vector representations of a pair of images. Each vector defines a point in a high-dimensional space. The feature vectors described in Section 2 form one such space. This raw feature space suffers from the "curse of dimensionality" – a large number of possibly noisy dimensions. This problem can be addressed by constructing a calibrated subspace to emphasize signal and dampen noise.

In such a calibrated space, meaningful distances between images can be determined. Distance matrices can be used to construct dendrograms that visualize population trends. Dendrogram algorithms build a minimal spanning tree from distance information. Branch lengths in a dendrogram indicate similarity between images, and branch angles are inconsequential. Every pair of images is connected by a path; the more similar a pair of images, the shorter the path. In the results given below (Figure 8), a heatmap was applied to represent the known age of each node as a color.

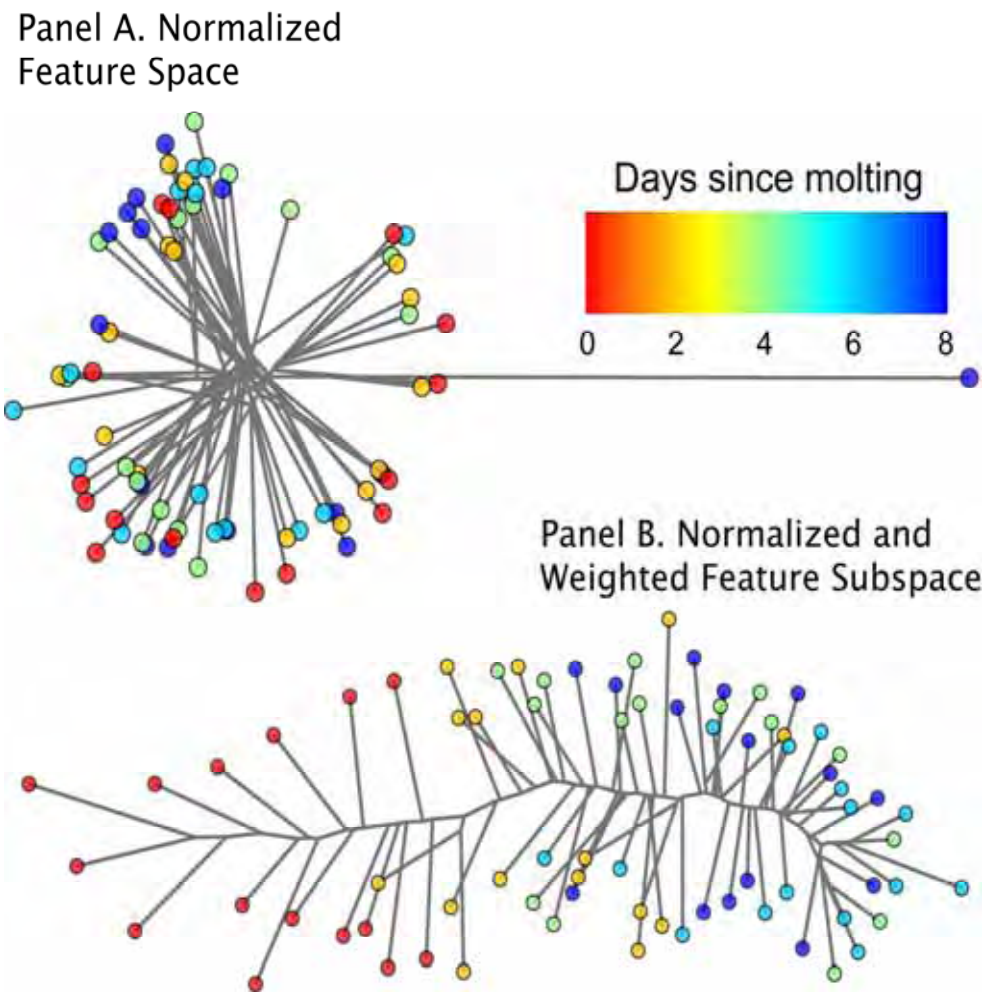


Fig. 8. Individual images of worm pharynx at different ages are represented by colored circles and plotted on dendrograms using distance measures from two types of feature space (see text). The color of each circle represents the known age of each worm in the image. Panel A was constructed from a normalized subspace, and Panel B from a normalized and weighted subspace.

In Figure 8, individual images of worm pharynx (see Figure 7 for image examples) are represented by colored circles in dendrograms computed from two different feature spaces. Normalization of the feature space was accomplished with a linear offset and scaling to a uniform range. Panel B of this figure was constructed from a normalized and weighted

subspace. The subspace was formed by first normalizing then scaling each dimension by its Fisher Discriminant score (see Section 3). Thirty-five percent of the lowest-scoring dimensions were excluded from this space. Dimensional weighting is commonly used in pattern recognition to construct a subspace in which meaningful distances can be determined (Parades & Vidal, 2006; Ricci & Acesani, 1999). Pair-wise Euclidean distance matrices between individual images were computed in these two spaces. The unrooted dendrograms were calculated from the distance matrices with the Fitch-Margoliash method implemented in the Phylip software package (Felsenstein, 1989). The heatmap representing the known age of the worms was applied subsequently.

The normalized full feature space places nearly all images equidistant to each other, which is not meaningful. An exception not shown in the figure is one image whose mean distance to other images was 10,000 times the median value. The FD-weighted feature subspace produces a gradient from young (red, left) to old (blue, right). The FD scores were calculated from unordered class information, yet the subspace yields class ordering. This indicates that the FD-weighted subspace can be used for biologically meaningful measures of similarity.

6. Algorithm Execution Frameworks and the Open Microscopy Environment

Several practical problems emerge when calculating features on large image sets. Full computation of features can take several days for a thousand-image dataset. To avoid recomputation, extracted features need to be stored systematically. To save time when classifying experimental images, it is desirable to calculate only the subset of features identified during dimensionality reduction. Additionally, when a feature bank is extended with more algorithms or permutations, reusing previous results can save considerable amounts of time. A workflow manager and informatics infrastructure addresses these problems better than simply unleashing all of these algorithms on folders full of image files.

The Open Microscopy Environment (OME) provides an image informatics infrastructure which includes facilities for archiving images, meta-data and analysis results. It also provides an Analysis System which executes image-processing algorithms in complex workflows, stores algorithm state and results in a database, and distributes algorithm execution across multiple networked computers. Results from any point in the workflow can be exported out of the OME database for subsequent analysis. The image features presented in Section 2 were computed in OME, and exported into MATLAB. Subsequent dimensionality reduction, signature weighting, dendrograms, and machine learning were performed with other software. Future plans are to integrate this functionality more tightly into OME.

OME is an open source software suite with a thin-client/server architecture (Goldberg et al., 2005; Swedlow et al., 2003). Users access OME using an internet-browser connecting to an extensible, dynamically generated web interface (Johnston et al., 2006). Based around a PostgreSQL database, OME has a middleware layer that provides functionality such as access control, user settings, and image annotation. OME data and computations are performed entirely server-side, optionally using a dedicated cluster. OME is designed to handle gigabytes of high-dimensional microscopy and medical imaging formats, as well as generic TIFF images.

OME algorithm wrappers are called AnalysisModules and are defined using the eXtensible Markup Language (XML) (Achard et al., 2001). AnalysisModule definitions are comprised of two sections: (1) the data modeling section describes the module's name, description,

inputs and outputs and (2) the execution instructions section specifies the interface to the algorithm's implementation (Macura et al., 2005).

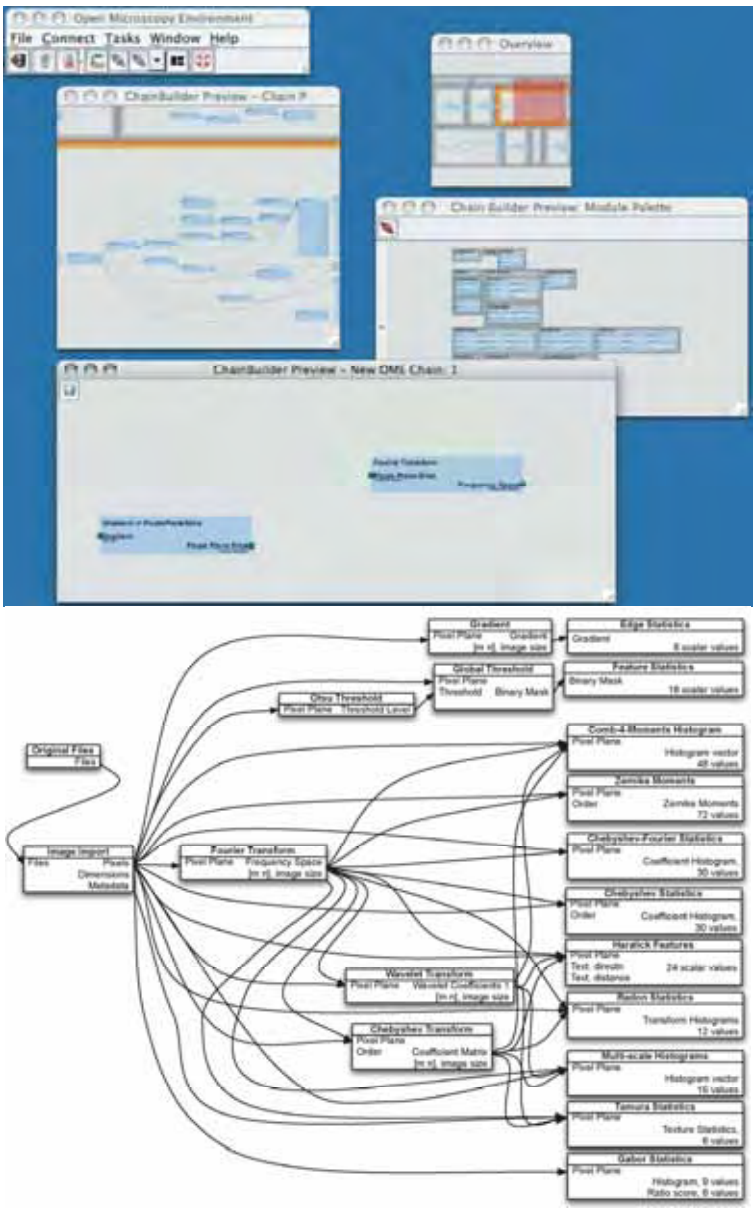


Fig. 9. Top Panel: the ChainBuilder Tool is a user interface for building complex workflows. Bottom Panel: Schematic of the feature extraction workflow (See Fig. 2, Section 2) built using ChainBuilder.

AnalysisModules' inputs and outputs can be connected interactively using the ChainBuilder GUI Tool (See Figure 9), to form workflows called AnalysisChains. After the image processing workflow has been modeled as an analysis chain, the workflow is executed by the OME Analysis System against images managed by OME. The OME Analysis System exploits branching in AnalysisChains to realize that some AnalysisModules can be executed concurrently. This concurrent execution can occur on a local multi-core node as well as remote nodes connected on a network.

OME overhead on a single processor is insignificant (5% of the execution time) and distributed computing is significantly faster (6x using 4 dual-core processors) than executing the algorithms natively in MATLAB. The results from executing algorithms in OME and storing intermediary results in a database and middle-layers agree to native algorithm execution results to the precision expected for 32 bit floating-point representation.

The OME software, implementations of feature extraction algorithms discussed in Section 2, along with AnalysisModule wrappers and AnalysisChains are all available for download from openmicroscopy.org.

7. Summary and Conclusions

This chapter discusses a general computer vision approach to the problem of automatic image classification and similarity measurements. The approach is based on using a large number of image descriptors followed by automated dimensionality reduction and classifier training. Calculation of feature descriptors was implemented as a component of OME, a system for collecting, archiving, annotating and analyzing images and metadata. High dimensional abstract feature sets are needed to allow universal context-independent description of image content. These descriptors in themselves are not sufficient to fully describe image content because their relative importance must first be determined in a given context. Context can be provided by example using training sets and supervised learning to determine which descriptors are important and which are not. This context-dependent weighing of descriptors leads to feature sets that correlate well with independently determined measures of image similarity.

8. Acknowledgements

The images in Figure 4 were from Pamela Bradley (NIH). Images in Figure 5 were from Andrew J. Westphal (UC-Berkeley). Images from Figures 6 and 7 were from Catherine A. Wolkow (NIH). Harry Hochheiser wrote the ChainBuilder tool presented in Figure 9. D. Mark Eckley (NIH) helped with editing of this manuscript. This research was supported by the Intramural Research Program of the NIH, National Institute on Aging.

9. References

Achard, F.; Vaysseix, G. & Barillot, E. (2001). XML, bioinformatics and data integration. *Oxford Bioinformatics,* 17, 2, 115-125, 1460-2059.

Awate, S. P.; Tasdizen, T.; Foster, N. & Whitaker, R. T. (2006). Adaptive Markov modeling for mutual-information-based, unsupervised MRI brain-tissue classification. *Medical Image Analysis,* 10, 726-739, 1361-8415.

Bishop, C. (1996) *Neural networks for pattern recognition*. Oxford University Press.

Boland, M.; Markey, M. & Murphy, R. (1998). Automated recognition of patterns characteristic of subsellular structures in florescence microscopy images. *Cytometry,* 33, 366-375, 1097-0320.

Boland, M. V. & Murphy, R. F. (2001). A Neural Network Classifier Capable of Recognizing the Patterns of all Major Subcellular Structures in Fluorescence Microscope Images of HeLa Cells. *Oxford Bioinformatics,* 17**,** 12, 1213-1223, 1460-2059.

Chapelle, O.; Haffner, P. & Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks,* 10, 1055-1064, 1045-9227.

Cocosco, C. A.; Zijdenbos, A. P. & Evans, A. C. (2004). A fully automatic and robust brain MRI tissue classification method. *Medical Image Analysis,* 7, 5513-527, 1361-8415.

Dong, S. B. & Yang, Y. M. (2002) Hierarchical web image classification by multi-level features. *International Conference on Machine Learning and Cybernetics.* pp.663-668.

Duller, A. W. G.; Duller, G. A. T.; France, I. & Lanb, H. F. (1999). A pollen image database for evaluation of automated identification systems. *Quaternary Newsletter,* 89, 4-9, 0143-2826.

Felsenstein, J. (1989). PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics,* 5, 164-166, 0748-3007.

Ferro, C. & Warner, T. A. (2002). Scale and texture in digital image classification. *Photogrammetric engineering and remote sensing,* 68, 51-63, 0099-1112.

Flickner, M.; Sawhney, H.; Niblack, W.; Ashley, J.; Huang, Q.; Dom, B.; Gorkani, M.; Hafner, J.; Lee, D.; Petkovic, D.; Steele, D. & Yanker, P. (1995). Query by image and video content: The QBIC system. *IEEE Computer,* 28, 23-32, 0018-9162.

Fukunada, K. (1990) *Introduction to statistical pattern recognition*. Academic Press, San Diego.

Funt, B. V. & Finlayson, G. D. (1995). Color Constant Color Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 17, 522-529, 0162-8828.

Gabor, D. (1946). Theory of communication. *Journal Institution of Electrical Engineers,* 93, 429-457,

Goldberg, I. G.; Allan, C.; Burel, J.-M.; Creager, D. A.; Falconi, A.; Hochheiser, H.; Johnston, J.; Mellen, J.; Sorger, P. K. & Swedlow, J. R. (2005). The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging. *Genome Biology,* 6**,** 5, 1465-6914.

Gradshtein, I. S. & Ryzhik, I. M. (1994) *Table of integrals, series and products,* 5 edn. Academic Press, San Diego, NY, Boston, London, Sydney, Tokio, Toronto.

Gregorescu, C.; Petkov, N. & Kruizinga, P. (2002). Comparison of Texture Features Based on Gabor Filters. *IEEE Transactions on Image Processing,* 11, 1160-1167, 1057-7149.

Gurevich, I. B. & Koryabkina, I. V. (2006). Comparative analysis and classification of features for image models. *Pattern Recognition and Image Analysis,* 16, 265-297, 1555-6212.

Hadjidementriou, E.; Grossberg, M. & Nayar, S. (2001) Spatial information in multiresolution histograms. *Computer Vision and Pattern Recognition* pp.702.

Hamming, R. W. (1989) *Digital filters,* 3 edn. Englewood cliffs, NJ: Prentice-Hall.

Heidmann, G. (2005). Unsupervised image categorization. *Image and Vision Computing,* 23, 861-876, 0262-8856.

Hoggar, S. G. (2006) *Mathematics of Image Analysis: Creation, Compression, Restoration, Recognition*. Cambridge University Press, Cambridge.

Huang, J.; Liu, Z. & Wang, Y. (2005). Joint scene classification and segmentation based on hidden Markov model. *IEEE Transactions on Multimedia,* 7, 538-550, 1520-9210.

Jain, A. & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19, 153-157, 0162-8828.

Jiebo, L.; Boutell, M.; Gray, R. T. & Brown, C. (2005). Image transform bootstrapping and its applications to semantic scene classification. *IEEE Transactions on Systems, Man and Cybernetics,* 35, 563-570, 0018-9472.

Jing, X. Y. & Zhang, D. (2006). A Face and Palmprint Recognition Approach Based on Discriminant DCT Feature Extraction. *IEEE Transactions on Systems, Man and Cybernetics,* 34, 2405-2414, 0018-9472.

Johnston, J.; Nagaraja, A.; Hochheiser, H. & Goldberg, I. G. (2006). A Flexible Framework for Web Interfaces to Image Databases: Supporting User-Defined Ontologies and Links to External Databases. *Proceedings of 2006 IEEE International Symposium on Biomedical Imaging Proceedings*, Washington D.C., April 2006, IEEE, New York.

Kudo, M. & Sklansky, J. (2000). Comparison of algorithms that select features for pattern classifier. *Pattern Recognition,* 33, 25-41, 0031-3203.

Lehmann, T.; Guld, M.; Deselaers, T.; Keysers, D.; Schubert, H.; Spitzer, K.; Ney, H. & Wein, B. (2005). Automatic categorization of medical images for content-based retrieval and data mining. *Computerized Medical Imaging and Graphics,* 29, 143-155, 0895-6111.

Livens, S.; Scheunders, P.; Wouwer, G.; Dyck, D.; Smets, H.; Winkelmans, J. & Bogaerts, W. (1996). A Texture Analysis Approach to Corrosion Image Classification. *Microscopy Microanalysis Microstructures,* 7, 143-152, 1154-2799.

Macura, T. J.; Johnston, J.; Creager, D. A.; Sorger, P. K. & Goldberg, I. G. (2005) The Open Microscopy Environment Matlab Handler: Combining a BioInformatics Data & Image Repository with a Quantitative Analysis Environment. *http://www.openmicroscopy.org.uk/publications/MatlabHandler.pdf.*

Mohanty, N.; Rath, T. M.; Lea, A.; Manmatha, R.; Lew, M. S.; Chua, T. S.; Ma, W. Y.; Chaisorn, L. & Bakker, E. M. (2005) Lecture Notes in Computer Science. *International Conference on Image and Video Retrieval.* pp.589-598.

Murphy, R. F. (2004). Automated interpretation of protein subcellular location patterns: implications for early detection and assessment. *Annals of the New York Academy of Sciences,* 1020, 124-131, 0077-8923.

Murphy, R. F.; Velliste, M.; Yao, J. & Porreca, G. (2001) Searching Online Journals for Fluorescence Microscopy Images Depicting Protein Subcellular Location Patterns. *2nd IEEE International Symposium on Bio-Informatics and Biomedical Engineering.* pp.119-128. Bethesda, MD.

Orlov, N.; Johnston, J.; Macura, T.; Wolkow, C. & Goldberg, I. (2006) Pattern recognition approaches to compute image similarities: application to age related morphological change. *International Symposium on Biomedical Imaging: From Nano to Macro.* pp.1152-1156. Arlington, VA.

Parades, R. & Vidal, E. (2006). Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 28, 7, 1100-1110, 0162-8828.

Pentland, A. & Choudhury, T. (2000). Face recognition for smart environments. *IEEE Computer,* 33, 2, 50-55, 0018-9162.

Qiu, G.; Feng, X. & Fang, J. (2004). Compressing histogram representations for automatic colour photo categorization. *Pattern Recognition,* 37, 2177-2193, 0031-3203.

Ranzato, M.; Taylor, P. E.; House, J. M.; Flagan, R. C.; Lecun, Y. & Perona, P. (2007). Automatic recognition of biological particles in microscopic images. *Pattern Recognition Letters,* 28, 31-39, 0167-8655.

Ricci, F. & Acesani, P. (1999). Data Compression and Local Metrics for Nearest Neighbor Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 21, 4, 380-384, 0162-8828.

Rodenacker, K. & Bengtsson, E. (2003). A feature set for cytometry on digitized microscopic images. *Analytic cellular pathology,* 25, 1-36, 0921-8912.

Rosenfeld, A. (2001). From image analysis to computer vision: an annotated bibliography, 1955-1979. *Computer Vision and Image Understanding,* 84, 298-324, 1077-3142

Ruifrok, A. & Johnston, D. (2001). Quantification of histochemical staining by color deconvolution. *Analytical and Quantitative Cytology and Histology,* 23, 291-299, 0884-6812.

Shen, L. & Bai, L. (2006). MutualBoost learning for selecting Gabor features for face recognition. *Pattern Recognition Letters,* 27, 1758-1767, 0167-8655.

Smeulders, A. W. M.; Worring, M.; Santini, S.; Gupta, A. & Jain, R. (2000). Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22, 1349-1380, 0162-8828.

Smith, J. R. & Chang, S. F. (1994) Quad-tree segmentation for texture-based image query. *Second Annual ACM Multimedia Conference.* pp.279-286.

Smith, J. R. & Chang, S. F. (1996) Local color and texture extraction and spatial query. *International Conference on Image Processing.* Lausanne, Switzerland.

Stricker, M. A. & Orengo, M. (1995) Similarity of Color Images. *SPIE Storage and Retrieval for Image and Video Databases.* pp.381-392.

Swedlow, J. R.; Goldberg, I.; Brauner, E. & Peter, K. S. (2003). Informatics and Quantitative Analysis in Biological Imaging. *Science,* 3000, 100-102, 0036-8075.

Tamura, H.; Mori, S. & Yamavaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions On Systems, Man and Cybernetics,* 8, 460-472, 0018-9472.

Tieu, K. & Viola, P. (2004). Boosting image retrieval. *International Journal of Computer Vision,* 56, 17-36, 1573-1405.

Veltkamp, R.; Burkhardt, H. & Krieger, H. (2001) *State-of-Art in Content-Based Image and Video Retrieval*. Kluwer Academic Publishers, Kluwer.

Yang, Y. & Wu, X. (2004). Parameter tuning for induction algorithm-oriented feature elimination. *IEEE Intelligent Systems,* 19, 40-49, 1541-1672.

Yu, H. & Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition,* 34, 2067-2070, 0031-3203.

13

Wavelet Evolution and Flexible Algorithm for Wavelet Segmentation, Edge Detection and Compression with Example in Medical Imaging

Igor Vujović[1], Ivica Kuzmanić[1], Mirjana Vujović[2], Dubravka Pavlović[3] and Joško Šoda[4]

[1] *University of Split, Maritime Faculty*
[2]*Private practice of occupational health, Ploče*
[3]*Health Centre Ploče, Radiological Department*
[4]*University of Split*
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Croatia

1. Introduction

A central goal in signal analysis is to extract information from signals that are related to real-world phenomena. Examples are the analysis of speech, images and signals in medical or geophysical application, to name it a few. One reason to analyze such signals is to achieve better understanding of the underlying physical phenomena. Another is to find compact representations of signals which allow compact storage or efficient transmission of signals through real-world environments. The methods of analyzing signals are wide spread and range from classical Fourier analysis to various types of linear time-frequency transforms and model-based and non-linear approaches.

Wavelet methods in image processing, analysis, compression, superresolution and enhancement are widely present in many researches such as biomedical applications, technology, industry, robotics, space explorations, military, etc. Wavelets have evolved over years. The theory of the first generation of wavelets (FGW) is originated on filter banks theory which includes classical Fourier analysis techniques (Mallat, 1999; Vetterli & Kovačević, 1995). Classical Fourier analysis is an irreplaceable tool in many engineering fields for years, and was solved many problems of linear-time invariant systems that include finding a spectrum of stationary signals (Proakis & Manolakis, 2006). For a non-stationary character of measured signal that spectral content is changing over time, classical Fourier analysis has shown weaknesses. The Fourier analysis only partly solves mentioned problems, a new approach is needed which will give a new insight into signal properties in a different way. Proposed new approach has been time-frequency analysis, i.e. a signal representation in time-frequency plane. The most popular time-frequency analyses are the short-time Fourier Transform (STFT) which is also called the classical method of time-frequency analysis and Wavelet Transform (WT or FGW) which is also called the time-scale analysis (Mertins, 1999). Wavelet transform brought flexible windows for analysis. The

second generation wavelet transform (SGW) is a newly proposed wavelet transform where the filters are not designed explicitly, but the transform consists of application of the lifting scheme. The sequence of lifting steps could be converted to a regular discrete wavelet transform, but it is unnecessary because both design and application is made via the lifting scheme (Sweldens, 1996, Daubechies & Sweldens, 1998). Measured signals of the main interest are not periodic. The area of the interest is not always finite and one-dimensional signals are not always uniformly sampled. At two or more dimensions (i.e. irregular surface) even more complicated situation arises. The FGW localize time-frequency well. Developed fast algorithms for FGW would be adopted in some way, by giving up dilatations and translation. Second generation wavelets (SGW) have updates and predictions instead of filter representation, the SGW have polyphase representation (Jansen & Oonicx, 2005). Factorization by lifting steps was a new approach, which introduces a new quality in computation of wavelet and scaling coefficients. Lifting transform can be applied to FGW as well. Then computationally interesting polyphase matrixes are obtained, which become triangle or scalar for the FGW. It is possible to construct FGW on the SGW settings and vice versa, but the SGW are so powerful that there is no need for transformation of SGW to FGW. The nanotechnology is the reason for improvement of SGW. Namely, research of nanostructures needs better characterization of atoms. The third generation wavelets (TGW) are proposed in (Xiao 2003, Jiang 2003, Vujović et al., 2006a; Vujović et al., 2006b). Wavelets have showed they are unlike numerous techniques which only remain popular for a short period of time – and they demonstrated ability to adopt.

Wavelets have shown great potential and abilities in various technical applications (Šoda, 2005). Nowadays, they are topical in image processing for on and off-line applications (computer vision, robot vision, security systems, etc).

Object segmentation through human-robot interactions in the frequency domain (Arsenio, 2003) was based on segmentation of windowed FFT. But, windowed FFT can be easily transformed to WT. Segmentation of colour images with fast wavelet transform is presented in (Chan et al, 2005).

Interesting application of wavelets for progressive edge detection and edge defection prediction has been developed in the XXI century (Abbas & Alsultanny, 2005). It exploits the observation that wavelet decomposition at higher levels degrades the image in the sense of leaving almost nothing but edges. However, their progressive and predictive detection is based on simple ones. It is not preferable in nowadays science, because everyone tries to find more and more complicated methods. Authors of this chapter evoke for such approach on many occasions. It is the best when you get satisfactory results with simple and elegant methods.

Compression of data, including image compression, is one of the most outstanding applications of wavelets. Some older examples are in references (Heer & Reinfelder, 1990; Said & Pearlman, 1996; Calderbank et al., 1997; Akay, 1998). Nowadays, influence of wavelets in many compression applications is being researched, i.e. in biomedical imaging (Vujović, 2004; Vujović et al, 2003.). Powerful compression possibilities of wavelets have been exploited in many applications, off and on-line, for single images and for image sequences. Wavelets are incorporated in JPEG-2000 standard as well and security (Boles, 1998; Grosbois, 2003; Dai & Yuen, 2006). However, their ability in denoising and compression often depend on thresholding. Automated methods for thresholding are of great interest for wavelets.

Wavelet compression ability gave rise to the idea of reverse process using them for obtaining higher resolutions. A great interest exists for such superresolution issues in the military, security, police, etc., as well as scientific community (Candocia, 1998; Nguyen, 2000; Bose, 2003; Borman, 2004; Chappalli & Bose, 2005).
This chapter describes an interesting approach in wavelet usage for image processing. Superresolution is used for image enhancement before compression by downsampling. The entire process is performed on the wavelet coefficients.

2. Wavelet generations

Heisenberg principle is interesting in the time-frequency domain, because it states that there is a limitation of measurement for time and frequency at the same time. If we can measure time and frequency infinitely precisely, the product of time and frequency is bounded according to Heisenberg principle. Actually, Heisenberg states that we can measure only time or only frequency with infinite precision. The product of time interval, Δt, and frequency interval, Δf, is constant.
This window is area in which it is presumed that amplitude is unchanged (of course, that is only a rough approximation in practice, which introduces error). The consequence of such window size is the worst resolution of time at high frequencies and the worst resolution of frequency at lower frequency range. Wavelet analysis is a multiresolution analysis (MRA): rectangles are vertically elongated at high frequencies, which means better time resolution and horizontally elongated at low frequencies, which means better frequency resolution. This limitation is better described by tiling scheme presented in Fig. 1.

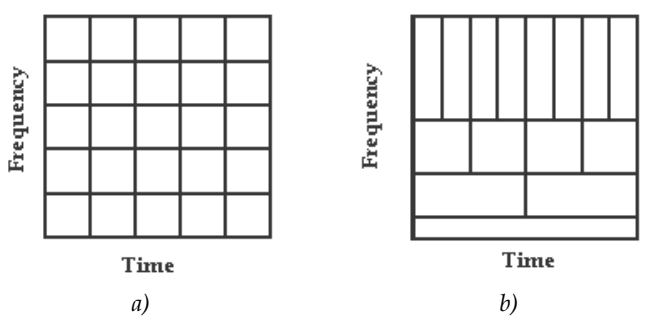


Fig. 1. Tiling scheme: a) STFT – same window for frequency and time for high and low frequency range, b) MRA – windows have the same surface, but different edge lengths

Once a window has been chosen for the STFT, then the time-frequency resolution is fixed over the entire time-frequency plane since the same window is used at all frequencies. To overcome the resolution limitations of the STFT one can imagine letting the resolution Δt and Δf vary in time-frequency plane in order to obtain a multiresolution analysis. The analysis filter bank is then composed of band pass filters with constant relative bandwidth, so called "constant Q-analysis".
The integral transform is one of the most important tools in signal theory (Mertins, 1999). Fourier transform is the best known example, but there are many other transforms, such as

Hartley and Hilbert, that can be derived from the integral signal representation. In the following, we will briefly outline the basic concept of integral transform.
The basic idea of an integral representation is to describe a signal *x(t)*, that is integrable in Lebesque sense and closed on $L_2(R)$, via its density *X(s)*, that is also integrable in Lebesque sense and closed on $L_2(R)$, with respect to arbitrary kernel *φ(t,s):*

$$x(t) = \int_S X(s)\varphi(t,s)ds \quad t \in T \subseteq L_2(R) \tag{1}$$

Using analogous approach, and denoting *θ(s,t)* as reciprocal kernel, the density *X(s)* can be calculates in the form:

$$X(s) = \int_T x(t)\theta(s,t)dt \quad s \in S \subseteq L_2(R) \tag{2}$$

By substituting (2) in (1) it can be obtained:

$$x(t) = \int_T x(\tau) \int_S \theta(s,\tau) \cdot \varphi(t,s) \cdot ds \cdot d\tau \tag{3}$$

In order to state the condition for the validity of (3) in a relatively simple form the so called Dirac impulse *δ (t)* is required. A generalized function *x (t)* then can be presented as follows:

$$x(t) = \int_T x(\tau) \cdot \delta(t-\tau) \cdot d\tau \tag{4}$$

Equations (3) and (4) show that the kernel and reciprocal kernel must satisfy:

$$\int_S \theta(s,\tau) \cdot \varphi(t,s) \cdot ds = \delta(t-\tau) \tag{5}$$

Similarly, by substituting (1) in (2), and then applying the same approach as above, implies:

$$\int_S \varphi(t,\sigma) \cdot \varphi(s,t) \cdot dt = \delta(s-\sigma) \tag{6}$$

A special category is that of self-reciprocal kernels. That corresponds with orthonormal bases in the discrete case and satisfies:

$$\varphi(t,s) = \theta^*(s,t) \tag{7}$$

Transforms that contain a self-reciprocal kernel are also called unitary transforms.
Let *x (t)* be a real or complex-valued continuous-time signal which is integrable in Lebesque sense. For such signals the Fourier transform exists:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j \cdot \omega \cdot t} \cdot dt \tag{8}$$

Here $\omega = 2 \cdot \pi \cdot f$ and *f* is the frequency in Hertz.
If *X (ω)* is also integrable in Lebesque sense, *x (t)* can be reconstructed from *X (ω)* via the inverse Fourier transform:

$$x(t) = \frac{1}{2 \cdot \pi} \int_{-\infty}^{+\infty} X(\omega) \cdot e^{j \cdot \omega \cdot t} \cdot d\omega \tag{9}$$

The kernel used is:

$$\varphi(t, \omega) = \frac{1}{2 \cdot \pi} e^{j \cdot \omega \cdot t} \quad T \in \langle -\infty, +\infty \rangle \tag{10}$$

and for reciprocal kernel we have

$$\theta(\varpi, t) = e^{-j \cdot \omega \cdot t} \quad S \in \langle -\infty, +\infty \rangle \tag{11}$$

From the equations (10) and (11) it can be seen that trigonometric functions form a basis that span the Fourier space. Trigonometric functions satisfy (5), i.e. they form the orthonormal basis on Fourier space. Also, the support of trigonometric functions is infinite in the time domain, which means that localization in the time is poorly determined, i.e. time resolution is poor. Unlike to time domain, in frequency domain Fourier transform gives perfect resolution, since trigonometric functions can be described with Dirac impulse. Heisenberg principle of uncertainty does apply here too.
The wavelet transform *W (a, b)* of a continuous-time signal x (t) is defined as:

$$W(a, b) = |b|^{\frac{-1}{2}} \cdot \int_{-\infty}^{+\infty} x(t) \cdot \psi^*(\frac{t-a}{b}) \cdot dt \tag{12}$$

Thus, the wavelet transform can be viewed, and is computed, as the inner product of *x (t)* and translated and scaled versions of a single function *ψ (t)*, the so-called wavelet. A wavelet function *ψ (t)* is a function of zero average. If *ψ (t)* is considered to be a bandpass impulse response, then the wavelet analysis can be understood as a bandpass analysis. By varying scaling parameter *b* the centre frequency and the bandwidth of the bandpass are influenced. The variation of *a* simple means a translation in time, so for a fixed *b* the transform (12) can be seen as a convolution of *x (t)* with the time-reversed and scaled wavelet

$$W_x(t, b) = |b|^{\frac{-1}{2}} \cdot x(t) * \psi_b(t), \quad \psi_b(t) = \psi^*(\frac{-t}{b}) \tag{13}$$

Time and frequency resolution of WT depends of *b*. For high analysis frequencies, good time localization but poor frequency resolution can be achieved. On the other hand, for low analysis frequencies, good frequency but poor time resolution can be achieved. When using a transform in order to get better insight into the properties of a signal, it should be ensured that the signal can be perfectly reconstructed from its representation. Otherwise the representation may be completely or partly meaningless. For WT the condition that must be met in order to ensure perfect reconstruction is:

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\psi(\omega)|^2}{|\omega|} \cdot d\omega < \infty \tag{14}$$

Where *Ψ (ω)* denotes FT of the wavelet. This condition is known as the admissibility condition for the wavelet *ψ (t)*.

Discrete wavelet transform (DWT) is based on multirate filter banks theory. There are two possible ways to obtain coefficients of DWT, by applying one of the two MRA algorithms, or by sampling CWT coefficients. The following dyadically arranged sampling points are used:

$$b_m = 2^m, \quad a_{mn} = b_m \cdot n \cdot T = 2^m \cdot n \cdot T \tag{15}$$

This yields the values $W_x\ (a_{mn}, bm) = W_x\ (2^m nT, 2m)$. Furthermore,

$$\psi_{mn}(t) = |b_m|^{\frac{-1}{2}} \cdot \psi(\frac{t - a_{mn}}{b_m}) = 2^{\frac{-m}{2}} \cdot \psi(2^{-m} \cdot t - nT) \tag{16}$$

Finally, (12) becomes:

$$W_x(a_{mn}, b_m) = W_x(2^m nT, 2^m) = \langle x, \psi_{mn} \rangle \tag{17}$$

The values $\{W_x\ (2^m nT, 2^m),\ m,n \in R\}$ form the representation of $x\ (t)$ with the respect to the wavelet $\psi\ (t)$ and the chosen grid. We cannot assume that any set $\psi_{mn}(t),\ m,\ n \in R$ allows reconstruction of all signals $x(t) \in L_2(R)$. For this a dual set $\tilde{\psi}_{m,n}(t)$, $m,\ n \in R$ must exist, and both set must span $L_2\ (R)$, any $x(t) \in L_2(R)$ can be written as:

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \langle x, \psi_{mn} \rangle \cdot \tilde{\psi}_{mn}(t) \tag{18}$$

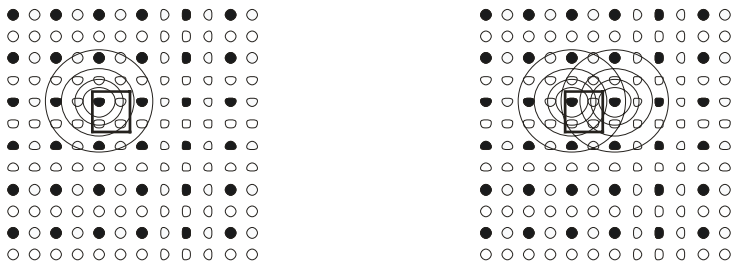
Alternatively, x(t) can be written:

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \langle x, \tilde{\psi}_{mn} \rangle \cdot \psi_{mn}(t) \tag{19}$$

For a given wavelet $\psi(t)$, the possibility of perfect reconstruction is dependent on the sampling interval T. If T is chosen very small i.e. we have oversampling, the values $\{W_x(2^m nT, 2^m),\ m,n \in R\}$ are highly redundant, and reconstruction is very easy. Then the functions $\psi_{mn}(t),\ m,\ n \in R$ are linearly dependent, and an infinite number of dual sets $\tilde{\psi}_{mn}(t)$ exists. The question of whether a dual set $\tilde{\psi}_{mn}(t)$ exists at all can be answered by checking two frame bounds A and B. It can be shown that the existence of a dual set and the completeness are guaranteed if the stability condition:

$$A \cdot \|x\|^2 \le \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |\langle x, \psi_{mn} \rangle|^2 \le B \cdot \|x\|^2 \tag{20}$$

with the frame bounds $0 < A \le B < \infty$ is satisfied (Mertins, 1999). The higher the frame bounds are, the smaller is the reconstruction error. In the case of a tight frame, A = B, perfect reconstruction with $\tilde{\psi}_{mn}(t) = \psi_{mn}(t)$ is possible. With MRA and wavelets resolution is degraded or enhanced by necessity. MRA trades off between both resolutions.



a) b)

Fig. 2. a) Interaction of the single pixel to the neighbours, b) interaction between two pixels

When talking about images, WT is two-dimensional. The error in image analysis begins with digitalization. Namely, in sensors. Sensors are not continuous. They are usually CCD arrays. The basic starting point is that light has the same frequency and amplitude over a single CCD cell. It is not true. However, it is often a good enough approximation. To obtain better image quality, more details must be obtained by some form of interpolation method.
Interpolation method can be primitive and simple or more sophisticated. Transition from a low-resolution image to a more detailed (high resolution image) does not depend only on the observed pixel, but also on its neighbouring pixels. But how can the neighbouring pixels be accounted for? I.e. are the pixels in diagonal positions less influenced by the observed pixel and vice versa? The solution is in introduction of weights for pixels. If this is performed on FGW coefficients we call the product "intuitive wavelets":

$$W(a,b) \cong \int \rho(a,b)x(t)\cdot\psi^{*}_{j,k}(t)dt = d_{j,k} \tag{21}$$

where ρ(a, b) is the weight function. Observe that if ρ(x, y) is the weight of the pixel, then this is propagating through WT into ρ(a, b), because the weight function is just a set of constants. Introduction of weights can be interpreted as primitive type of SGW. Therefore it can be said that this is the SGW on the FGW settings. However, SGW can be introduced for discrete signal and linear filters, which perform perfect reconstruction in z-domain.
Polyphase representation of signal $X(z) = X_p(z^2) + z^{-1}X_n(z^2)$ where X_p i X_n even and odd samples of the signal x and can be written as:

$$X_p(z) = \sum_k x[2k]z^{-k} \text{ and } X_n(z) = \sum_k x[2k+1]z^{-k}$$

The final result is the polyphase matrix of the system:

$$P(z) = \begin{pmatrix} H_p(z) & G_p(z) \\ H_n(z) & G_n(z) \end{pmatrix} \tag{22}$$

In simulations and numerical experiments, the result is the estimated matrix $\widetilde{P}$ and the error P - $\widetilde{P}$ has to be minimized. Filtering is directly performed on either even or odd samples, which breaks down number of operations by factor 2.

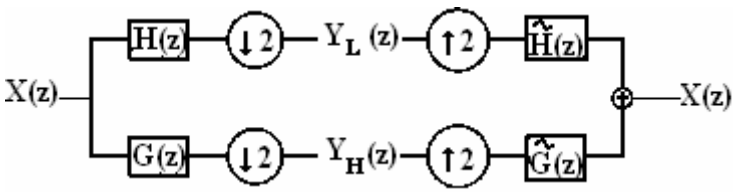


Fig. 3. Perfect reconstruction in Z-domain

The authors propose that FGW and SGW pass the morphology preprocessing in order to emphasize edges. Wavelet coefficients obtained by that manner can be called the third generation wavelets (TGW). This will facilitate further enhancement in different applications. An algorithm for TGW is proposed in (Vujović, Kuzmanić & Vujović, 2006a), but it is not the only way. When talking about TGW, another possibility can be to enhance wavelet coefficient matrixes by i.e. motion field. If stationary image is processed, then quasi-superresolution has to be used.

3. Flexible algorithm

Many algorithms for edge detection, segmentation or compression exist. Some of them are based on wavelets. However, wavelets have some properties which can be used for different operations. The proposed algorithm exploits these properties.

3.1. Wavelet motion field

Let us consider an image sequence $I(p_i, t)$ with $p_i = (x_i, y_i) \in \Omega$ the location of each pixel in the image. The brightness constancy assumption states that the image brightness $I(p_i, t+1)$ is a simple deformation of the image at time t:

$$I(p_i, t) = I(p_i + v(p_i), t + 1) \tag{23}$$

where $v(p_i, t) = (u, v)$ is the optical flow between $I(p_i, t)$ and $I(p_i, t+1)$. This velocity field can be globally modelled as a coarse-to-fine 2D wavelet series expansion from scale L to l (Bruno & Pellerin, 2002):

$$V_\theta(p_i) = \sum_{k_1,k_2=0}^{2^L-1} c_{L,k_1,k_2} \Phi_{L,k_1,k_2}(p_i) +$$

$$+ \sum_{j\geq L}^{l} \sum_{k_1,k_2=0}^{2^j-1} \left[d^H_{n,k_1,k_2} \Psi^H_{j,k_1,k_2}(p_i) + d^V_{n,k_1,k_2} \Psi^V_{j,k_1,k_2}(p_i) + d^D_{n,k_1,k_2} \Psi^D_{j,k_1,k_2}(p_i) \right] \tag{24}$$

where $\Phi_{L,k_1,k_2}(p_i)$ is the 2D scaling function at scale L and Ψ^H_{j,k_1,k_2}, Ψ^V_{j,k_1,k_2}, Ψ^D_{j,k_1,k_2} are wavelet functions which represent horizontal, diagonal and vertical directions. These functions are dilated by 2^j and shifted by k_1 and k_2. The solution can be found by usage of some error function and minimization, i.e. (Bruno & Pellerin, 2002; Bruno & Pellerin, 2001):

$$\begin{aligned} E &= \sum_{p_i\in\Omega} \rho\big(I(p_i + V(p_i,t), t+1) - I(p_i,t), \sigma\big) = \\ &= \sum_{p_i\in\Omega} \rho\big(r(p_i + V), \sigma\big) \end{aligned} \tag{25}$$

and the motion wavelet coefficient vector, θ, is calculated by:

$$\theta = \arg\min_{\theta}(E) \tag{26}$$

Once motion wavelet coefficients have been estimated for each frame f_i of a sequence S containing M frames, anyone can obtain a feature space spanned by the motion feature vectors θ_i, i = 1, ... M. To temporally segment the feature spaces Ω_{seg} (spanned by θ_{seg}), (Bruno & Pellerin, 2002) consider a hierarchical classification with a temporal connexity constraint.

Another approach is only formally different (Wu et al, 1998). Approximation of motion vector, θ = [u(x,y) v (x,y)]^T, by using two-dimensional basis functions, is a natural extension of one-dimensional to two-dimensional basis functions of the tensor product. Accordingly, the two-dimensional basis functions are:

$$\Phi_{0,k_1,k_2}(x,y) = \phi(x-k_1)\phi(y-k_2) \tag{27}$$

$$\Psi^{H}_{j,k_1,k_2}(x,y) = \phi(2^j x - k_1)\psi(2^j y - k_2) \tag{28}$$

$$\Psi^{V}_{j,k_1,k_2}(x,y) = \psi(2^j x - k_1)\phi(2^j y - k_2) \tag{29}$$

$$\Psi^{D}_{j,k_1,k_2}(x,y) = \psi(2^j x - k_1)\psi(2^j y - k_2) \tag{30}$$

where the subscripts j, k_1 and k_2 represent the resolution scale, horizontal and vertical translations and the upper subscript H, V and D represent the horizontal, vertical and diagonal directions. Two dimensional motion vector can be expressed in terms of linear combinations of coarsest-scale function (13) and horizontal, vertical and diagonal wavelets (14 - 16) in finer levels. Motion vectors are (Wu, 1998):

$$u(x,y) = u_{-1}(x,y) + \sum_{,j=0}^{J}\left(u_j^H(x,y) + u_j^V(x,y) + u_j^D(x,y)\right) \tag{31}$$

$$v(x,y) = v_{-1}(x,y) + \sum_{,j=0}^{J}\left(v_j^H(x,y) + v_j^V(x,y) + v_j^D(x,y)\right) \tag{32}$$

where u_{-1} is:

$$u_{-1}(x,y) = \sum_{k_1=-2}^{L_1-2}\sum_{k_2=-2}^{L_2-2} c_{-1,k_1,k_2}\Phi_{0,k_1,k_2}(x,y) \tag{33}$$

where u_j in all directions is expressed as:

$$u_i^{H,V,D} = \sum_{k_1=-2}^{2^j L_1-2}\sum_{k_2=-2}^{2^j L_2-2} c_{j,k_1,k_2}^{H,V,D}\Psi_{j,k_1,k_2}^{H,V,D}(x,y) \tag{34}$$

v_{-1} and v_i are calculated analogly. Maximum likehood estimates [u(x, y) v(x, y)]^T are obtained by minimizing:

$$E = \sum_x\sum_y\left[I_1(x+u(x,y), y+v(x,y)) - I_0(x,y)\right]^2 \tag{35}$$

Equations (31 – 35) are easier for implementation than (23 – 26). They can be approximated as differences of neighbouring approximation, diagonal, vertical and horizontal coefficients. This approximation is used in quasi-superresolution algorithm (Vujović et al., 2006a; Vujović et al., 2006b).

3.2. Superresolution and quasi-superresolution

Superresolution includes restoration as a special case. The restoration equation can be rewritten within the superresolution framework as (Nguyen & Milanfar, 2000):

$$f_k = DC_kE_kx + n_k = H_kx + n_k \tag{36}$$

where p is the number of available frames and $1 \le k \le p$, f_k is an N x 1 vector representing the kth m x n LR image in columnwise order. If l is the resolution enhancement factor in each direction, x is an l^2N x l vector representing the lm x ln HR image in columnwise order, E_k is an l^2N x l^2N warping matrix that represents the relative motion between frame k and a reference frame, C_k is a blur matrix of size l^2N x l^2N, D is the N x l^2N uniform down-sampling matrix, and n_k is the N x 1 vector representing additive noise. Particularly in case of quasi-superresolution, only one image is available (k = 1). Than, superresolution problem can be replaced with filtering and (36) transforms to:

$$f = DCEx + n = Hx + n \tag{37}$$

Since, only in ideal case n = 0, (37) means that HR image is "less clear", which is totally subjective description.

3.3. Algorithm flow

The input image can be processed by morphology operations, but it is optional (block 1 in Fig. 4).

Noise reduction is in the nature of WT, so it is not included in the algorithm. It is also possible to combine the original and processed image. Then it must be chosen which transformation to use (filter or lifting approach). WT is performed between blocks 3 and 4.

Thresholding can be performed if necessary as the preprocessing for the compression or simple for denoising. This option can be performed automatically or manual. Next step is to enhance image incorporating wavelet motion field. When dealing with stationary stand alone image (i.e. in biomedical diagnostic images such as X-rays), motion field calculation is performed in quasi-superresolution manner (Vujović et al, 2006a). This is relative "motion" between wavelet coefficients. In on-line sequences quasi-superresolution can be performed when higher image resolution is necessary and motion can be resolved in some other way if someone do not prefer wavelet motion field. Then we can perform what we need. Edges are obtained by adding all four motion matrixes obtained in quasi-superresolution manner.

When approximation is down-sampled several times and reducing number of colours edges can be pointed out as well. When subtraction of motion matrixes from the enhanced original (previous steps) is performed, a good segmentation is obtained. If enhanced original is put through quasi-superresolution algorithm, HR image can be obtained.

Compression of images can be performed with or without reconstruction at HR grid. Compression can be obtained by thresholding of wavelet coefficients or by downsampling of wavelet coefficients. Multiple downsampling is proven to be useful for compression

(Vujović, 2004) in case study about pulmonary X-rays, when downsampling is performed 6 to 12 times without influence to the medical diagnosis. Of course, it is not generalized.

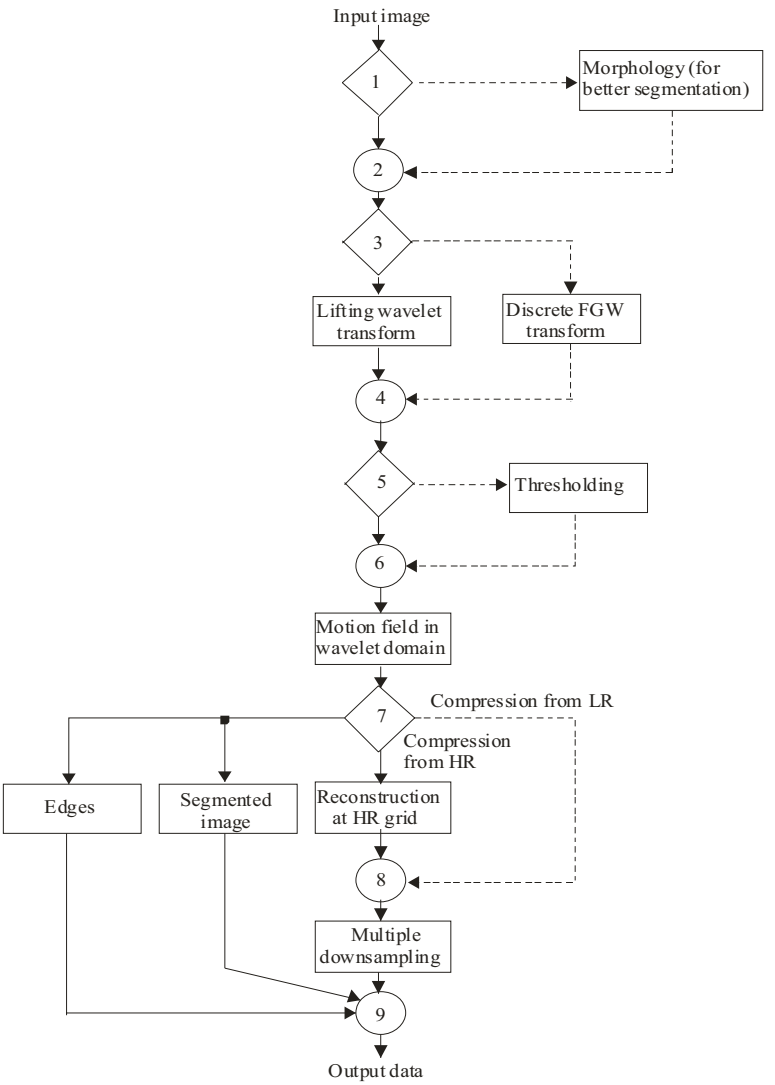


Fig. 4. Flexible algorithm for wavelet segmentation, edge detection and compression

4. Results

Times of execution depend from computer to the computer, so it is very difficult to compare. We executed algorithm on NEC notebook with Athlon XP-M AMD processor with 1.67 [GHz] and 480 [MB] RAM size with Windows XP operating system. Hard disk is half full and Norton Antivirus is active.

Fig. 5. Motion field execution on wavelet coefficients in stand alone image with quasi-superresolution reconstruction to HR grid

Type of wavelets (Matlab designation)	Time of execution of filtered WT [s]	Time of execution of lifted WT [s]	Improvement in percentage [%]
bior1.3	12.768	11.978	6.18
rbio1.3	12.598	12.528	0.55
haar	12.128	11.536	4.88

Table 1. Comparison of wavelet quasi-superresolution execution time

Fig. 5 to 12 shows some of the results. Figures are chosen to open discussion. There are better and worse examples.

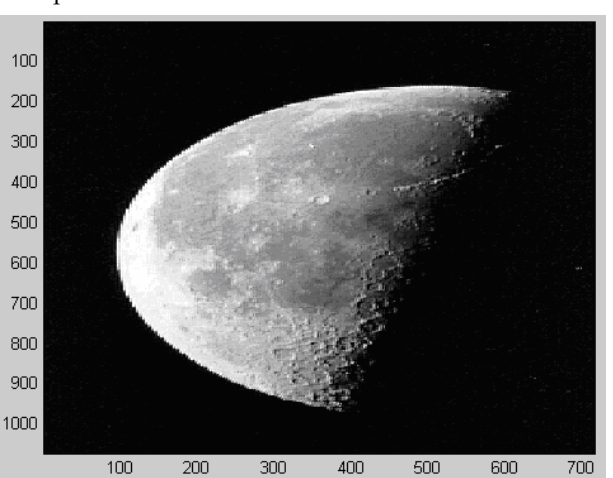


Fig. 6. Reconstruction after wavelet motion field for FGW haar

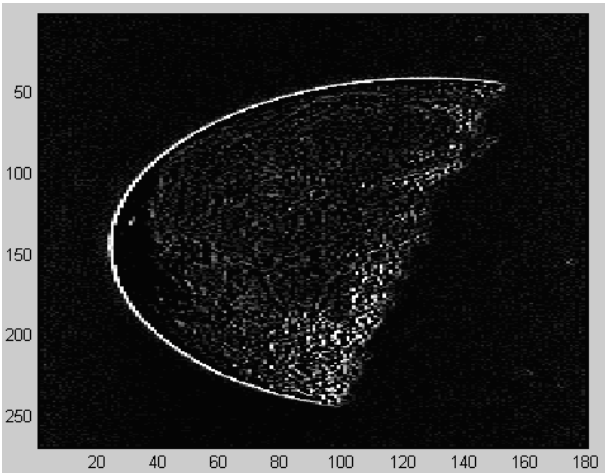
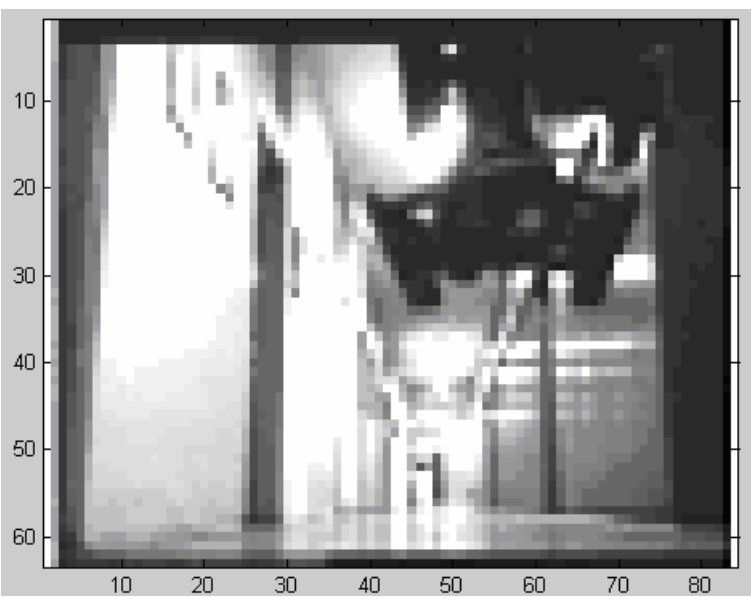
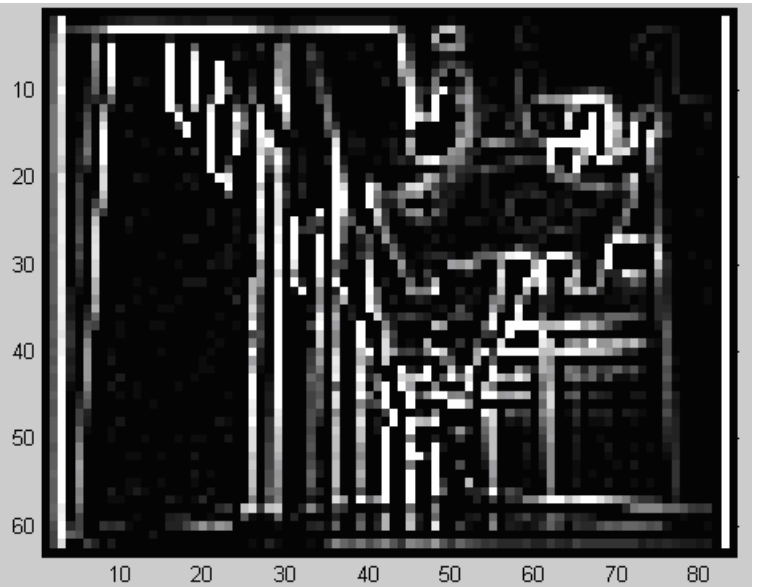


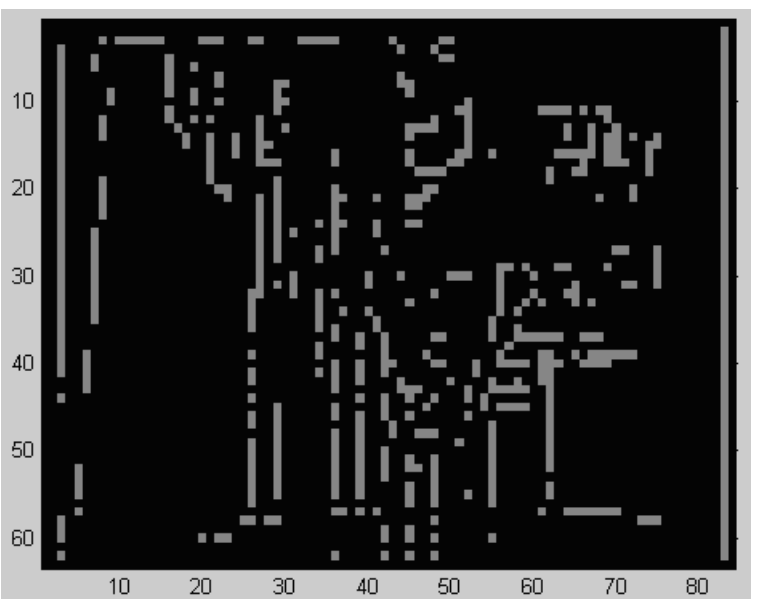
Fig 7. Simple edge detection by usage of only motion field vectors without the original from downsampled approximation coefficients



a)

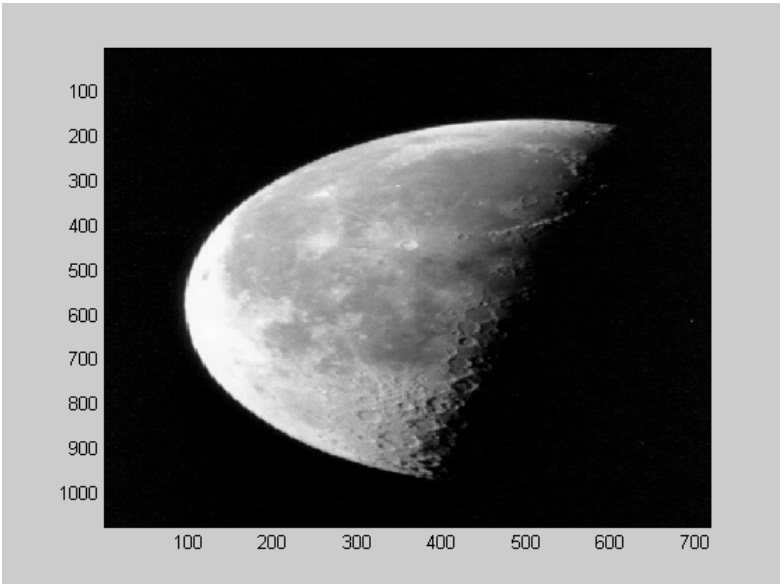


b)



c)

Fig. 8. A robot perspective: a) original image, b) edge detection by wavelet motion vectors with the original colour map, c) edge detection by wavelet motion vectors with the increased number of colours


100
200
300
400
500
600
700
800
900
1000
100
200
300
400
500
600
700

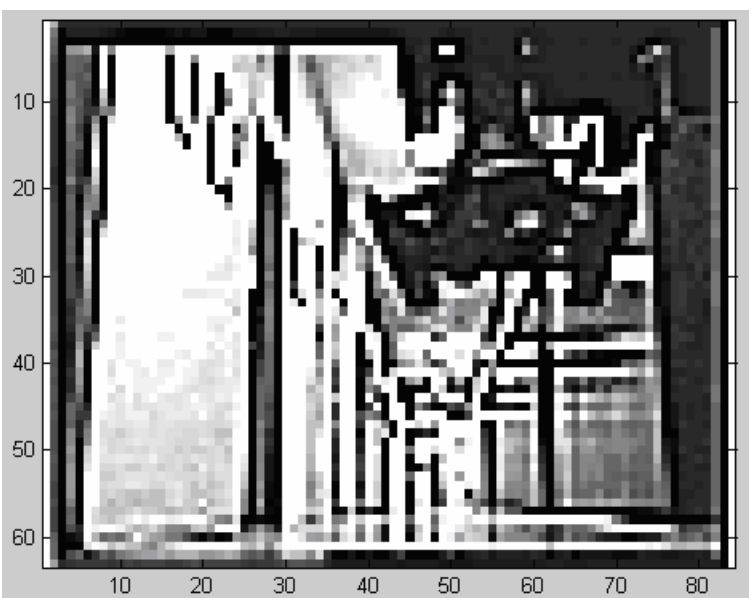
a)

100
200
300
400
500
600
700
800
900
1000
100
200
300
400
500
600
700

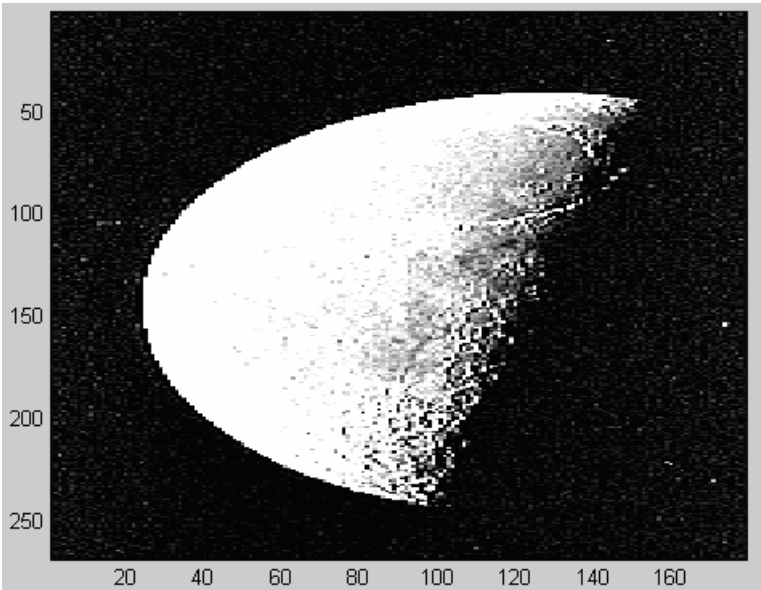
b)

c)

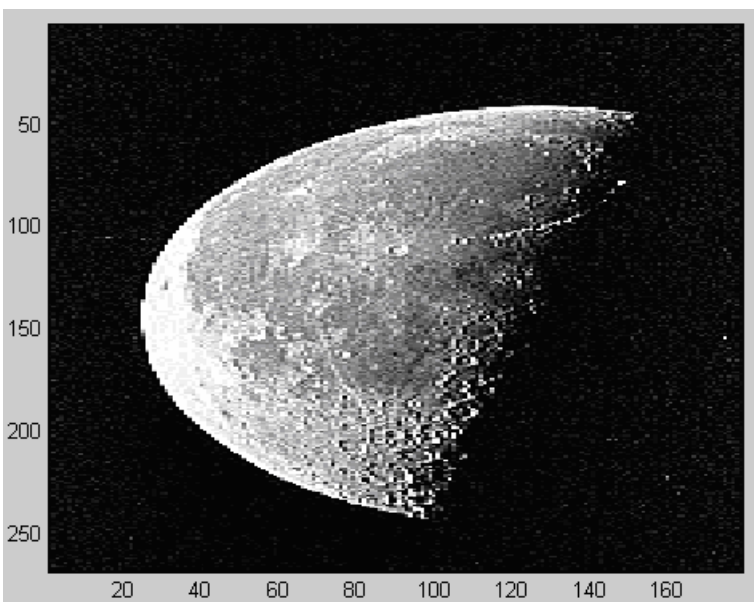
Fig. 9. a) Classic zoom of the approximation, b) quasi-superresolution on approximation with FGW, c) quasi-superresolution on approximation by SGW



a)

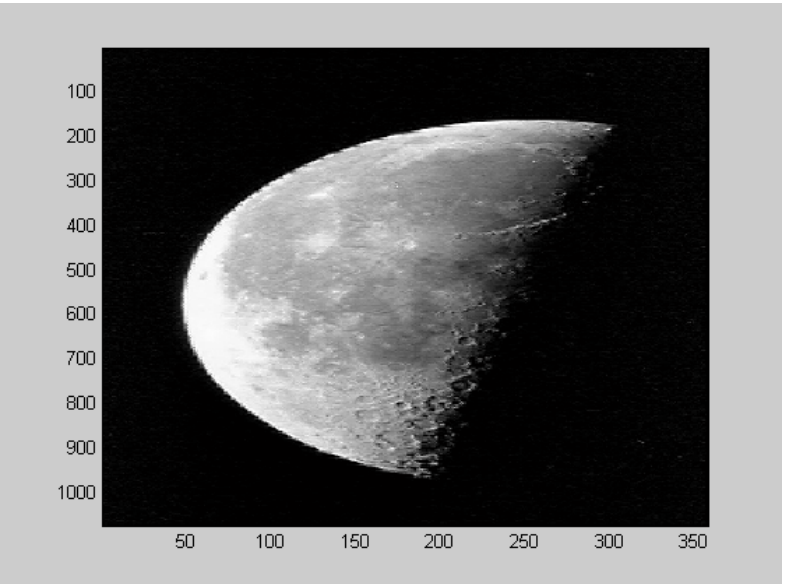


b)

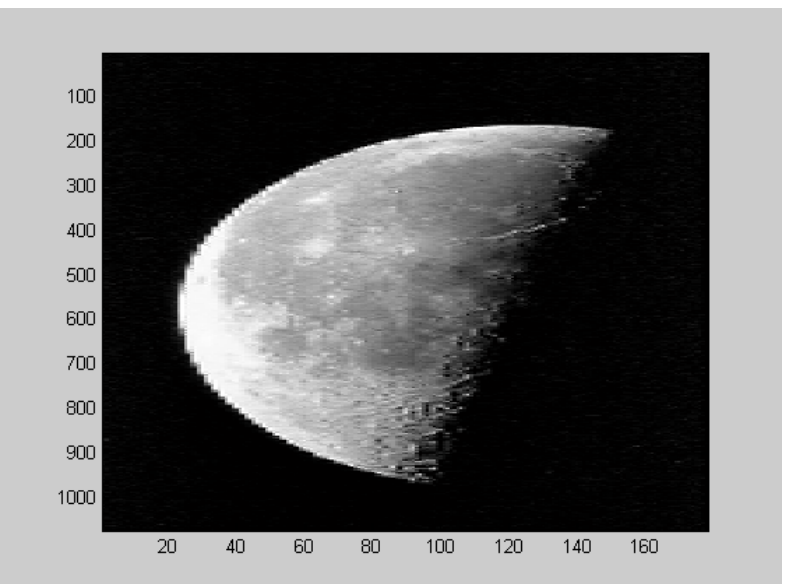


c)

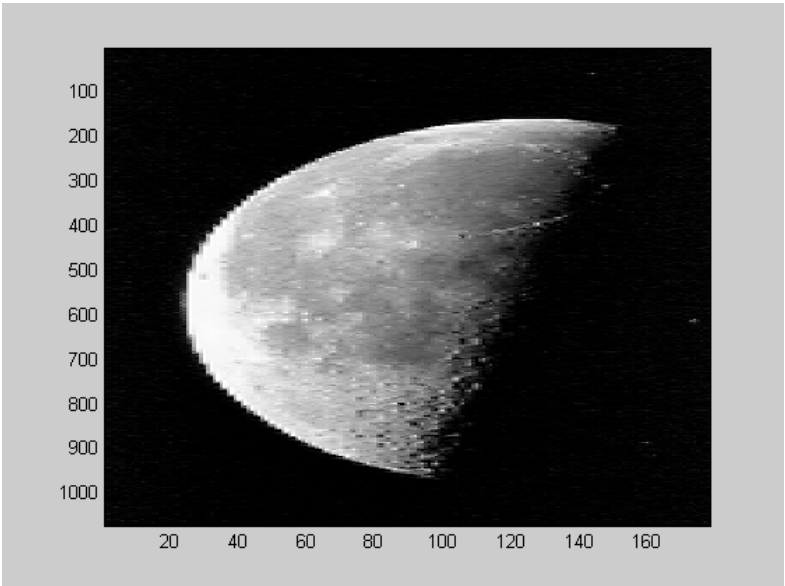
Fig. 10. Addition of motion fields in all directions subtracted from the approximation at the first level: a) robot's view, b) lifting WT, db2, c) lazy wavelet


100
200
300
400
500
600
700
800
900
1000
50
100
150
200
250
300
350

a)

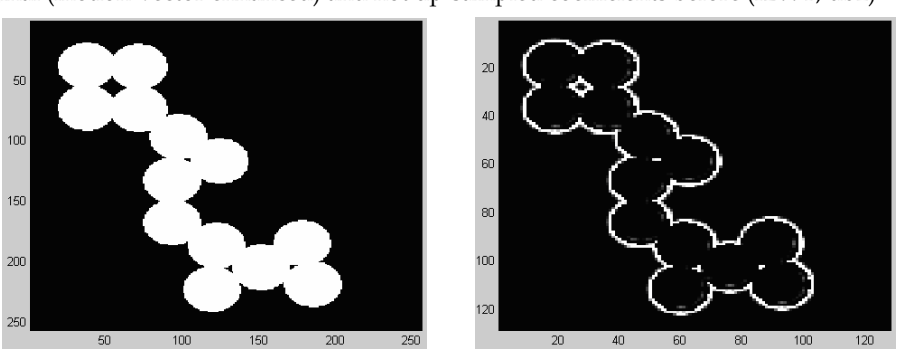

100
200
300
400
500
600
700
800
900
1000
20
40
60
80
100
120
140
160

b)



c)

Fig. 11. Image reconstructed from: a) down-sampled original (motion-vector enhanced before), coefficients were not up-sampled (IDWT, db2), b) twice down-sampled original (motion-vector enhanced), not up-sampled coefficients (IDWT, db2), c) twice down-sampled original (motion-vector enhanced) and not up-sampled coefficients before (ILWT, db2)



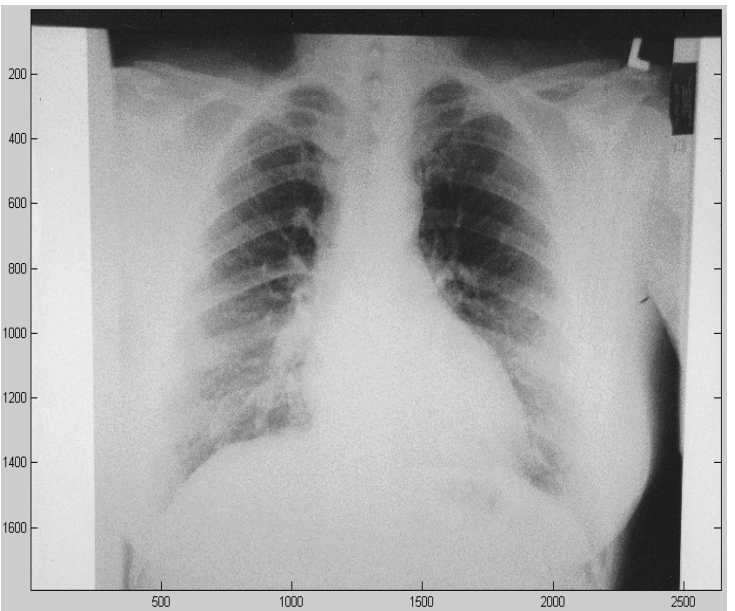
a) b)

Fig. 12. a) Original image, b) wavelet motion field edge detection

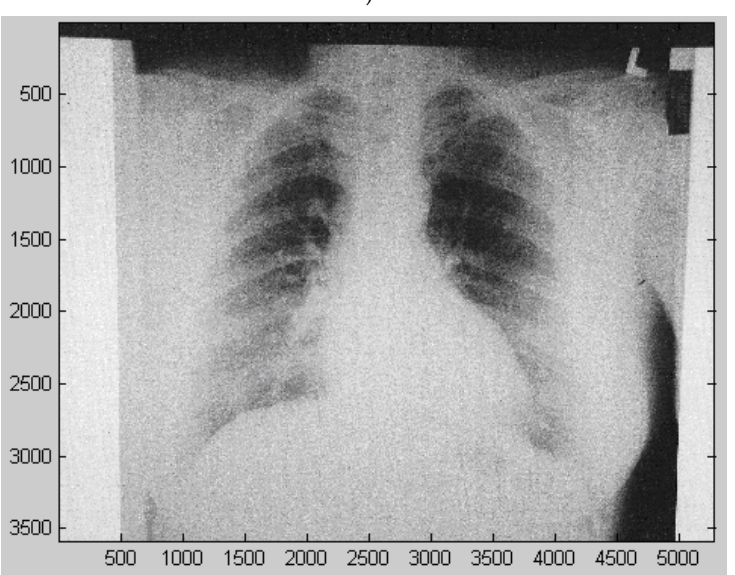
5. Example in medical imaging (Vision system for X-rays)

One of applications of vision systems is in medicine. Every modern hospital has Hospital Information System (HIS) or Picture Archiving and Compression System (PACS) at least in rudimental way. Telemedicine is old news. Our research started with compression of pulmonary X-rays for asbestosis infected patients. The problem was how to compress images without changing the diagnosis. In (Vujović, 2004) the goal is reached for lossy

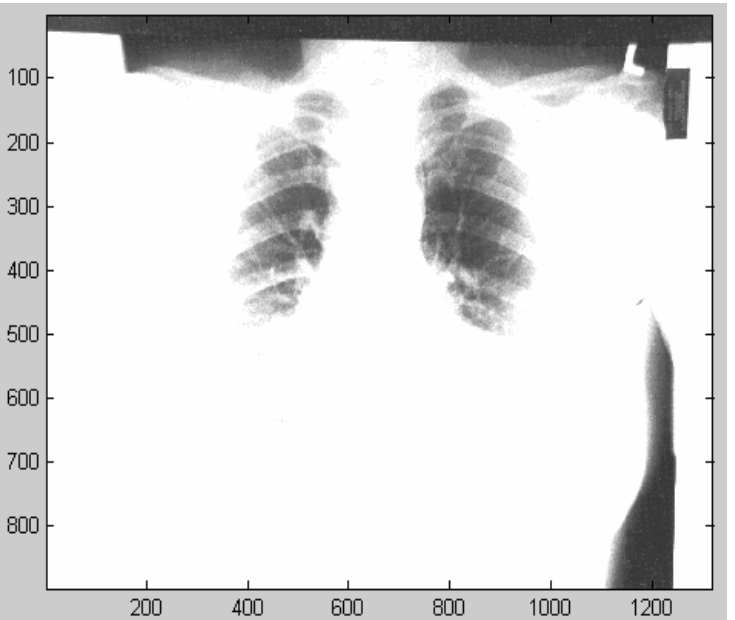
compression by down-sampling. Images were degraded in quality, but diagnostic value is preserved. Compression ratio obtained was 1:128 or higher (depending on type of wavelets). This was confirmed by three independent medical experts, as required by International Labour Organization.



a)



b)

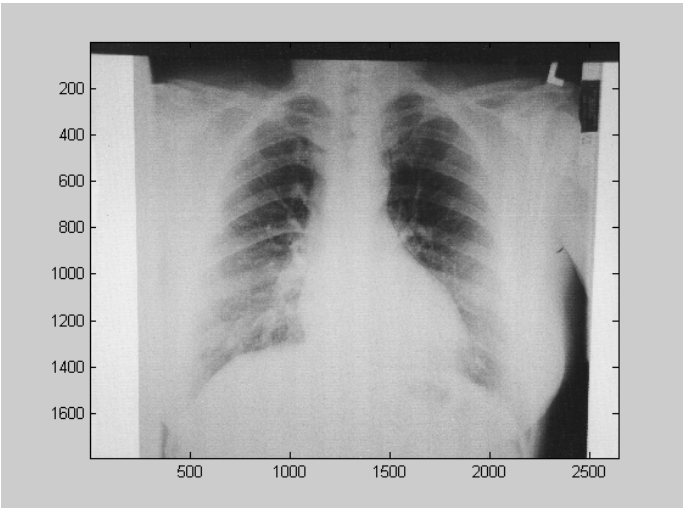


c)

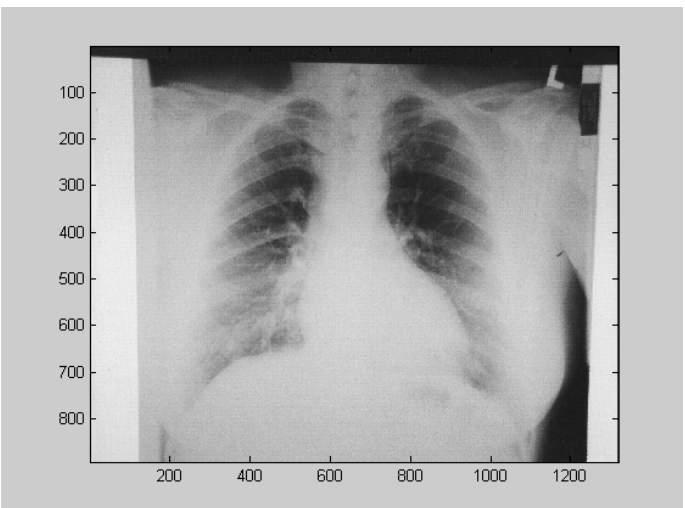
Fig. 13. a) Original X-ray of randomly chosen patient, b) motion field in wavelet domain (quasi- superresolution), c) approximation coefficients

Fig. 13.a shows original of randomly chosen patient. In Fig. 13.b motion-field enhanced, quasi-superresolution image is shown. Fig. 13.c shows approximation coefficients. Fig. 14. shows results on compression for wavelet motion field enhanced X-rays. Compression ratio for lossless compression is 1 : 8.0211 in Fig. 14.d and 1 : 4.0189 for Fig. 14.c.

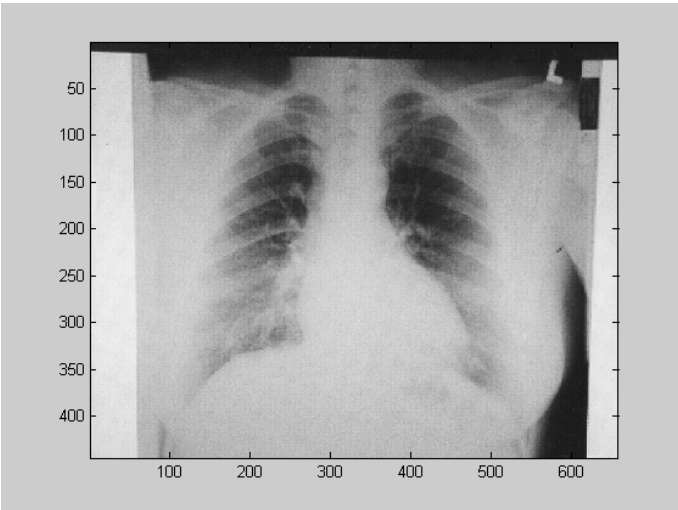
Superresolution and quasi-superresolution are, in nature, processes of obtaining higher resolutions and more details. The question in this case is what do the new details mean. Can it be beginning useful in prevention of diseases by early diagnosis (when medical experts still can not see the illness)? Or is it a cause of error, because the new details do not mean illness. The new details could be only math creation without meaning in nature. Which of this is true? The second danger is in thresholding, because small shadows (which mean illness) can be deleted if not carefully used. Medical diagnosis is not changed in such compression as illustrated.


200
400
600
800
1000
1200
1400
1600
500
1000
1500
2000
2500

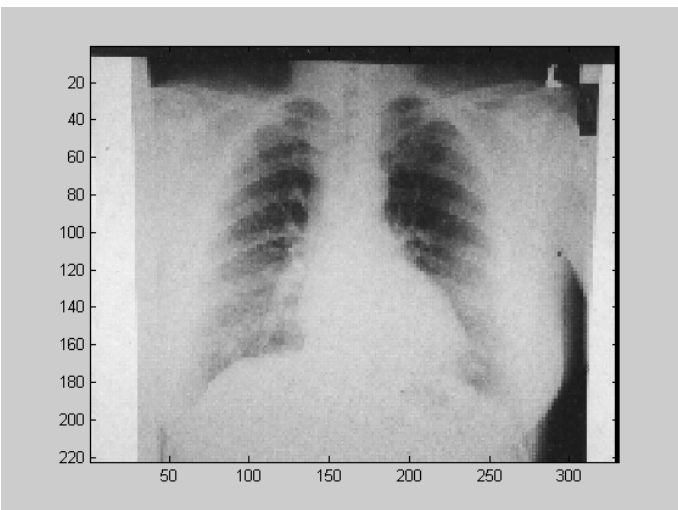
a)


100
200
300
400
500
600
700
800
200
400
600
800
1000
1200

b)



c)



d)

Fig. 14. a) Original X-ray, b) image down-sampled two times after wavelet motion field enhancement , c) image down-sampled three times after wavelet motion field enhancement, d) image down-sampled four times after wavelet motion field enhancement

6. Conclusion

Wavelets have evolved over years. FGW and SGW are still used and they are applied in more and more areas of research. Proposed algorithm is flexible, because of many options which can be used. It can be simple, but also complex. Disadvantage of many image

processing algorithms is that they do not give the same results on every class of images. So, they are not generalized. This algorithm has the same fate. It gives the best results for pulmonary X-rays with gray scale.

Time of execution is with active Norton Antivirus in Windows and Matlab with half-full hard disk. It would be considerable faster if it is executable stand alone application isolated and without antivirus application. There are a lot of programming solutions to make faster the algorithm. Since it is still in developing phase, we had the main interest in operation algorithm. Further work should include improvement of execution time.

Potential area of application is biomedical imaging, because there is no need to take care of execution time. However, it could be used in virtual reality systems and systems of augmented reality. This is possible, because it is not necessary to execute the algorithm in real-time all the time. Algorithm can be performed occasionally, i.e. when scene is changed. In the meantime, only differences in frames can be processed. This can be improved by choosing only limited regions of interest for processing.

It is important not to mix up motion field in an image sequence and in a stationary image. Motion field in the image sequence is defined as in section 3.1. Motion field in the stationary image is without sense, because there are no two frames to look for motions. However, quasi-superresolution states that we can find motion between neighbouring wavelet coefficients. So, this motion does not correspond to real motion in the observed scene. This is a novel idea, which helps in i.e. medical imaging, finger print analysis, human iris recognition, face recognition, etc. Fig. 12. shows potential of wavelet motion vectors in edge detection. Further research should be inclusion of colours and colour segmentation.

Vision systems in medicine must be carefully used, because of misdiagnosis danger. I.e. in superresolution, when an un-seen detail shows up, it could mean that illness is discovered before medical expert could see it. But, it can be a false positive. If a vision system is used instead, the system must be checked by medical experts for any possible case. The algorithm could be incorporated in computer hardware and sell as medical vision system. It has to be checked by appropriate bodies in different countries before.

7. References

Abbas, Y & Alsultanny, K. (2005). Edge Defection Prediction by Using Wavelet Compression. *Journal of Computer Science*, Vol. 1., No. 3., pp. 310-315, ISSN 15493636

Akay, M. (Ed.) (1998). *Time Frequency and Wavelets in Biomedical Signal Processing*, IEEE Press, Piscataway, ISBN 0-7803-1147-7, New York

Arsenio, A. M. (2003). Object Segmentation Through Human-Robot Interactions in the Frequency Domain. On-line: www.groups.csail.mit.edu/lbr/hrg/2003/arsenio_sib.pdf (URL last checked on December 2006)

Boles, W. W. (1998). A Security System Based on Human Iris Identification Using Wavelet Transform. *Engineering Applications of Artificial Intelligence,* Vol. 11., No. 1., pp. 77-85 ISSN 0952-1976

Borman, S. (2004). *Topics in Multiframe Superresolution Restoration*, PhD Thesis, University of Notre Dame, Indiana, USA

Bose, N. K. (2003). *Iterative Blind Second Generation Wavelet Superresolution and Role of Moving Least Squares*, Pennstate University, Army Research Office Grant DAAD 19-03-1-0261, Presentation.

Bruno, E. & Pellerin, D. (2002). Video Structuring, Indexing and Retrieval Based on Global Motion Wavelet Coefficients. *Proceedings of International Conference of Pattern Recognition* (ICPR), pp. 132-137, ISBN 0-7695-1695-X, Quebec City, Canada, 11-15. August 2002, IEEE Computer Society Press, Piscataway

Bruno, E. & Pellerin, D. (2001). Global Motion Model Based on B-spline Wavelets: Application to Motion Estimation and Video Indexing. *Proceedings of the 2nd Int. Symposium on Image and Signal Processing and Analysis ISPA'01*, pp. 102-107, S. Lončarić, H. Babić (Ed), Pula, Croatia, June 2001

Calderbank, A. R.; Daubechies, I.; Sweldens, W. & Yeo, B. L. (1997). Lossless Image Compression Using Integer to Integer Wavelet Transforms, *Proceedings of International Conference on Image Processing ICIP,* Vol. 1, pp. 596-599, ISBN 0-8186-8183-7, Washington, DC, USA, 26-29th October 1997.

Candocia, F. M. (1998). *A Unified Superresolution Approach for Optical and Synthetic Aperture Radar Images*, PhD Thesis, University of Florida.

Chan, D. Y.; Lin, C. H. & Hsieh, W. S. (2005). Image Segmentation with Fast Wavelet-Based Colour Segmenting and Directional Region Growing. *IEICE Transactions on Information & Systems*, Vol.E88–D., No.10., pp. 2249-2249, ISSN 0916-8532

Chappalli, M. B. & Bose, N. K. (2005). Simultaneous Noise Filtering and Super-resolution with Second Generation Wavelet. *IEEE Signal Processing Letters*, Vol. 12., No.11., pp. 772-775, ISSN 1070-9908

Dai, D. Q. & Yuen, P.C. (2006). Wavelet Based Discriminant Analysis for Face Recognition. *Applied Mathematics and Computation*, Vol. 175., No. 1., pp. 307–318, ISSN 0096-3003

Daubechies, I. & Sweldens, W. (1998). Factoring Wavelet Transforms Into Lifting Steps. *Journal of Fourier Analysis Applications*, Vol. 4., No. 3., pp. 247-269, ISSN 1069-5869

Grosbois, R. (2003). *Image Security and Processing in the JPEG 2000 Compressed Domain.* PhD Thesis, Université Paris, France

Heer, V. K. & Reinfelder, H-E. (1990). A Comparison of Reversible Methods for Data Compression. *Proceedings of SPIE "Medical Imaging IV"*, SPIE, Vol. 1233, pp. 354-365, ISBN 0-8194-0277-X

Jansen, M. & Oonincx, P. (2005). Second Generation Wavelets and Applications, Springer-Verlag, ISBN 1-85233-916-0, London

Jiang, X. Q. & Blunt, L. Third Generation Wavelet Model for Surface Texture. *Proceedings of 9th International Conference on Metrology and Properties of Engineering Surface*. Halmstad University, Sweden, 10-11 September 2003. ISBN 91-631-5455-2

Mallat, S. (1999). *A Wavelet Tour of Signal Processing* 2nd Edition. ISBN 0-12-466606-X Academic Press, London, UK

Mertins A. (1999). *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications* .John Wiley & Sons Ltd., ISBN 0-471-98627-7, Baffins Lane, Chichester, West Sussex, England

Nguyen, N. X. (2000). *Numerical Algorithms for Image Superresolution*. PhD Thesis, Stanford University, USA

Nguyen, N. & Milanfar, P. (2000). A Wavelet-Based Interpolation-Restoration Method for Superresolution, on line: www_sccm.stanford.edu/pub/sccm/sccm00-03.ps.gz (URL last checked, June 2006)

Proakis G. J. & Manolakis G. D. (2006). *Digital Signal Processing, Principles, Algorithms, and Applications.* 4th Ed., Pearson Prentice Hall Inc., ISBN 0-13-187374-1, Upper Saddle River, NJ, USA

Said, A. & Pearlman, W. A. (1996). An Image Multiresolution Representation for Lossless and Lossy Compression. *IEEE Transactions on Image Processing*, Vol. 5., No. 9., pp. 1303-1310, ISSN 1057-7149

Šoda, J. (2005). *Time-frequency Analysis of Measured Signal.* (In Croatian). Master Thesis, University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia

Sweldens, W. (1996). The Lifting Scheme: A Custom-design Construction of Biorthogonal Wavelets. *Application of Computer Harmoning Analysis*, Vol. 3., No. 2, pp. 186-200

Vetterli M. & Kovačević J. (1995). *Wavelets and Subband Coding.* Prentice-Hall Signal Processing Series. ISBN 0-13-097080-8, London, UK

Vujović, M.; Vujović, I. & Kuzmanić, I. (2003). The Application of New Technologies in Diagnosing Occupational Asbestosis. *Arhiv za higijenu rada i toksikologiju*, Vol. 54., No. 4., pp. 245-252, ISSN 0004-1254

Vujović, I. (2004). Application of Wavelets in Biomedical Signal Processing with Example in Compression of X-ray Images of Occupational Asbestosis Infected Patients. (in Croatian). *Master Thesis,* University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia.

Vujović, I.; Kuzmanić, I. & Vujović, M. (2006.a). Algorithm for Combined Wavelet Quasi-Superresolution. *Proceedings of 5th Int. Symposium Communication Systems Networks and Digital Signal Processing,* pp. 469-473, ISBN 960-89282-0-6, Patras, Greece, 19-21 July 2006, CSNDSP, School of CEIS, Northumbria University, Newcastle upon Tyne

Vujović, I.; Kuzmanić, I. & Kezić, D. (2006.b). Wavelet Superresolution and Quasi-Superresolution in Robot Vision. *Proceedings of 10th International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology"* TMT-2006, pp. 597-600, ISBN 9958-617-30-7, Barcelona-Lloret de Mar, Spain, 11-15th September 2006., Publishers: Faculty of Mechanical Engineering, Zenica, Bahçeşehir University Istanbul, Mühendislik Fakültesi, Turkey, Escola Tecnica Superior D'Enginyeria Industrial de Barcelona, Universitat Politecnica de Catalunya, Zenica, Barcelona, Istambul

Wichmann, E. H. (1988). *Quantum physics.* Textbook from University of Berkeley (translation to Croatian). ISBN 86-7059-056-5, Tehnička knjiga, Zagreb, Croatia

Wu, Y. T.; Kanade, T.; Cohn, J. & Li, C. C. (1998). Optical Flow Estimation Using Wavelet Motion Model. *Proceedings of 6th IEEE International Conference on Computer Vision*, pp. 992-998, Bombay, India, 4-7 January 1998.

Xiao, S. J.; Jiang, X. Q. & Blunt, L. (2003). Wavelet Bayesian Method for Denoising of Nano-scalar Surface Measurement. *The International Journal for Manufacturing Science & Production*, Vol. 5, No. 1-2., pp. 95 – 98, ISSN 0793-6648

14

Compression of Spectral Images

Arto Kaarna
Lappeenranta University of Technology
Finland

1. Introduction

In this chapter we describe methods how to compress spectral imaging data. Normally the spectral data is presented as spectral images which can be considered as generalizations of colour images. Rapid technological development in spectral imaging devices has initiated the need for the compression of raw data. Spectral imaging has been central to many remote sensing applications like geology and environment monitoring. Nowadays, new application areas have arisen in industry, for example in the quality control of assembly line products and in applications, where the traditional three-chromaticity colour measurements are not accurate enough. Spectral imaging produces large amounts of raw data which will be processed later in various applications. Image compression provides a possibility to reduce the amount of raw data for storing and transmission purposes. The image compression can be either lossless or lossy. In the lossy compression the quality of the reconstructed data should be estimated to evaluate the usefullness of the reconstructed data. The lossy compression is justified in the sense that the compression ratios are much higher than in the lossless case where the reconstructed data is identical to the raw data.

Spectral images are now available for different applications due to the development in the spectral imaging systems (Hauta-Kasari et al., 1999; Hyvärinen et al., 1998). Geoscience and remote sensing have been the main application areas of spectral images but nowadays several new application areas have emerged in industry. For example, applications in quality control, exact colour measurement, and colour reproduction use spectral information, since RGB colour information only is not sufficient.

Image compression has been one of the main research topics in image processing. The compression methods are usually developed for images visible to humans, i.e. for grey-scale or RGB colour images. Applications in the field of remote sensing and recent advances in industrial applications, however require the compression of spectral images (Vaughn & Wilkinson, 1995). Some compression methods are lossless (Memon et al., 1994; Roger & Cavenor, 1996), but most of the methods are lossy (Abousleman et al., 1997; Gelli & Poggi, 1999). Some applications can accept data which is compressed by a lossy scheme, but naturally the important features in the data must be present. If the lossy compression method cancels out any of the important features for the applications, then the lossless compression is the only possibility to decrease the amount of the raw data.

Compression is required due to the large amounts of data captured in the images. Regular digital cameras in everyday use apply JPEG or TIFF-compression. Images displayed in web-

pages are compressed with the same methods. Compression in these applications is accepted as a normal procedure as long as the visual quality is not reduced.
With spectral images the memory or the transmission requirements are very high. Observations of Earth in spatial, spectral, temporal and radiometric methods produce data volume which is growing faster than the transmission bandwidth (Abousleman et al., 2002; Aiazzi et al., 2001). This means, that for long term storing or transmission, these databases should be compressed. The compression should be such that the spatial and spectral quality of the reconstructed image is high enough for the application. Table 1 shows examples of spectral imaging systems developed for remote sensing (Kerekes &Baum, 2002; Lillesand & Kiefer, 2000; AVIRIS, 2006; HyMap, 2006; HYDICE, 2006; Landsat, 2006; Hyperion 2006; Ikonos, 2006; OrbView, 2006; Aisa Eagle, 2006).

Name	# of channels	Spatial resolution, m	Radiometric resolution, bits	Raw data: kB/km²
Airborne				
M7	12	10	8	120
AVIRIS	224	20	12	840
HYDICE	210	3	12	35000
HyMap	200	2	16	100000
Aisa Eagle	244	0.5	12	1400000
Spaceborne				
ERTS/MSS	4	80	8	0.6
Landsat/TM	7	30	8	7.8
Hyperion/EO-1	220	30	12	366.7
IKONOS	4/1	4/1	11	1719
OrbView-5	4/1	1.64/0.41	11	2045
OrbView-4	200	8	8	31250

Table 1. Examples of remote sensing systems. The spatial resolution for airborne sensors depends on the flight altitude. kB means kilobytes.

As an example, one spectral Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) (AVIRIS, 2006) tape, taken in one day, can have up to 16 GB of raw data. Large amounts of data are also recorded in an application for the quality control of ceramic tiles (Kälviäinen et al., 1998): imaging of 25 ceramic tiles made up a spectral database of size 312 megabytes. Nowadays, there are several conferences where new spectral imaging systems for industrial applications are presented (MCS, 2006; EI, 2006; IGARSS, 2006).
When the client's application is known in advance the data for it can be extracted from the original database. For example, in mineral mapping the spectral range from around 2.0μm to 2.5μm is sufficient. Infrared systems utilize also a narrow band above the visual range for example in night time vision systems. If the colour features are enough, one can extract 30 bands out of 224 from the AVIRIS images for that specific application. In all previous cases and for various client requirements, the high quality database or even the original database must be present for the data extraction.

As the imaging systems have developed, at the same time the resources for storing the images are advanced due to the technological changes. In Table 2 we show some development features in hard drive technologies and properties (Thompson & Best, 2000; Hughes, 2002; Grochowski & Halem, 2003; Moreiro, 2006).

Feature	1970	1980	1990	2000	2009
Density, Mb/cm^2	$1*10^0$	$5*10^1$	$3*10^2$	$4*10^4$	$5*10^5$
Internal data rate, Mb/s	0.8	2	4	50	200
Capacity, GB	0.03	0.3	1	100	1600
Price, $/MB	NA	200	8	0.05	<0.002

Table 2. Advances in hard drive features. NA stands for information Not Available.

A similar growth pace as for the hard disk drives is experienced also in digital transmission both in wired and wireless cases: in average, every fifteen years the capacities have become thousand-fold.

The spectral imaging systems produce a vector of values for each pixel of the image. The values depend on the resolution of the imaging system and they are normally presented as 8 bit, 12 bit or 16 bit values. Thus, a spectral image can be considered as a set of two-dimensional, equal size images. Now, compression methods can be similar to the methods applied to greyscale images or to RGB-colour images. For lossless compression also regular text compression methods can be applied. This simple approach may be usable, when a) the original image should be perfectly reconstructed, b) the compression method should be widely available, and c) high compression ratios are not required. These methods include entropy modelling followed by Huffman coding, arithmetic coding or Burrows-Wheeler transform. The standard Unix tool, gzip, is based on Lempel-Ziv coding (Ziv & Lempel, 1977). It gives the average lossless compression ratio 1.41 for a set of four Moffet Field scenes and 1.39 for a set of five Jasper Ridge scenes from the AVIRIS dataset (AVIRIS, 2006). Much better lossless compression ratios are received if the composition of the spectral images is observed. Best lossless compression methods are most often based on predictive coding combined with entropy modeling (Aiazzi et al., 2002; Aiazzi et al., 2001; Aiazzi et al., 1999; Benazza-Benyahia et al., 2001; Mielikäinen & Toivanen, 2003; Mielikäinen, 2006). Also integer transforms (Kaarna, 2001) or lossless vector quantization (Ryan & Arnold, 1997-1) is possible for the perfect reconstruction.

A lossy compression procedure for spectral images consists of three phases. The first phase decorrelates the raw data in spatial and spectral dimensions, the second phase quantizes the coefficients from the first phase. The third phase utilizes some lossless scheme to encode the quantized coefficients. This procedure is depicted in Fig. 1.

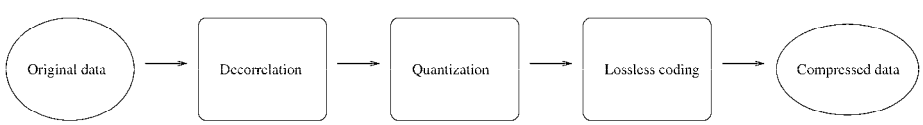


Fig. 1. The three phases in the lossy compression.

A compression procedure is of any practical interest only if it has an inverse procedure which reconstructs the original data or image. An inverse procedure includes the same phases as the compression procedure in Fig. 1, but they are processed in reverse order. First, the compressed data is decoded resulting in the quantized coefficients. Then the quantized coefficients are restored to their original values and these values compose the original data. In Fig. 2 the decompression, the inverse procedure for compression, is depicted.

Fig. 2. Decompression, the inverse procedure for compression.

Different methods and their parameter values are possible in the compression procedure. Depending on the selections the reconstructed data can be equal to the original data or some information can be lost. The methods and parameters are selected such that the important features of the image are present and the information lost can be regarded as an observation noise or otherwise irrelevant for the application. The evaluation of the quality of the reconstructed data is necessary when using lossy compression. The quality measurements are most often based on the pixelwise or bandwise difference between the original image and the reconstructed image resulting to logarithmic signal-to-noise ratio (Rabbani & Jones, 1991). Specific measures for spectral images include both the percentage maximum absolute distortion measure (PMAD) (Ryan & Arnold, 1997-2; Ryan & Arnold, 1998) and the blockwise distortion measure for multispectral images BDMM (Kaarna & Parkkinen, 2002). PMAD guarantees that every value in the reconstructed image is within a maximum distance from the original value. The maximum distance is relative to the original value. BDMM correlates blockwise filtering of the original and the reconstructed image to the visual quality of the distorted images.

In the following sections we introduce methods for lossy and lossless compression of spectral images. Then we describe how to evaluate the quality of reconstructed data in the lossy case. Finally, we show experimental results and evaluate different compression methods.

2. Lossy compression of spectral images

A comprehensive study on theoretic aspects of lossy source coding can be found from (Berger & Gibson, 1998). Data compression is thoroughly considered in (Donoho et al., 1998). Both scalar and vector quantization is widely surveyed in (Gray & Neuhoff, 1998). The wavelet transform is described in detail in (Daubechies, 1992; Taubman & Marcellin, 2002).

Lossy compression methods achieve remarkably higher compression ratios than lossless compression by neglecting some unessential data in the compression phase. Several lossy compression methods have been developed for the compression of spectral images. Some of them are two-dimensional methods applied separately to each band of the spectral image

(Abousleman et al., 1994), and some methods have been further enhanced from the two-dimensional methods to be three-dimensional (Abousleman et al., 1997; Kaarna & Parkkinen 1999). Most of the recent methods apply separate subtasks to the spectral and spatial dimensions due to their dissimilar characteristics (JPEG2000, 2006; Aware, 2006; Kaarna et al., 2000; Kaarna et al., 2006).
A rough classification of the compression methods for the spectral images include the principal component analysis (PCA) for the decorrelation of the spectral data, the wavelet transform for the spatial compression of images, predictive methods applied simultaneously to the spectral and spatial dimensions of the image, and finally the vector quantization of the spectra in the image. Each of these methods can alone compress the image, but in practise, best compression results are obtained through combining these methods.

2.1 Vector quantization

Clustering is an unsupervised method to classify patterns in an image. Patterns within a cluster are more similar to eachother than they are to a pattern belonging to another cluster. Thus, a lossy compression method can be established on that notation: each member of a cluster are represented by the cluster center. The compressed data consists of cluster centers and an index image.
Vector quantization utilizies the previous idea (Ryan & Arnold, 1997-1; Ryan & Arnold, 1997-2). First, a decomposition of the image into a set of vectors is performed. With spectral images the decomposition naturally consists of the spectral vectors. Then a codebook is generated from a training set of vectors using an iterative algorithm. Finally, each spectral vector of the image is quantized to the closest vector in the codebook according to the selected distortion measure. The compressed data consists of a codebook and a set of indices to the codebook. One index is required for each spectrum of the image.
The generalized Lloyd algorithm (GLA) tries to optimize the codebook C. The algoritmic presentation of the GLA is :

Algorithm 1:

Step 1: Select the initial codebook C_1, set $m=1$.
Step 2: With the given codebook C_m perform one iteration to generate an improved codebook C_{m+1}.
Step 3: Compute the average distortions for C_{m+1}. If the change from the previous iteration is small enough, then stop.
Otherwise set $m = m+1$ and continue from Step 2.

The Step 2 of Algorithm 1 is generally implemented using a Nearest Neighbor condition:
Algorithm 2:
Step 1: Using the codebook $C_m = y_i$ partition the training set T into clusters R_i with the NN condition: $R_i = \{ x \in T : d(x,y_i) \leq d(x,y_j);\ \ all\ j \neq i \}$.
Step 2: Compute the centroids for the clusters $\{cent(R_i)\}$ to obtain an improved codebook $C_{m+1} = \{cent(R_i)\}$.

In vector quantization each vector is represented by the centroid of a cluster it belongs to. The resulted data from the VQ consists of the cluster centroids and of an index image, which

describes the inclusion of each vector into one cluster. In Fig. 3. we illustrate the vector quantization for the compression of spectral images.

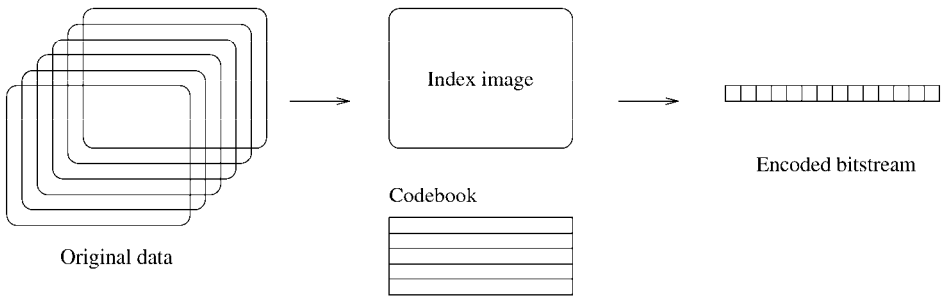


Fig. 3. Vector quantization in lossy compression of spectral images.

Similar approach to vector quantization was defined in (Toivanen et al., 1999). The procedure produced an index image and a codebook which was generated with a Self-Organizing Map (SOM). The results were good compared to a clustering method in (Kaarna et al., 1998).

Even though vector quantization is theoretically optimal in lossy coding, the implementation issues constrain the performance (Poggi & Ragozini, 2002). Computational complexity and memory requirements have been the main drawbacks that have been attacked (Poggi & Ragozini, 2002; Kaarna et al., 2000; Ryan & Arnold, 1997-2; Kamano et al., 2001).

A tree-structured product-codebook was designed to support progressive transmission (Poggi & Ragozini, 2002). In product-codebook VQ, all codewords are of type $x_{ij} = u_i * v_j$, where * is the decomposition rule. The component codebooks were organized in a tree-structure in order to speed up the codeword selection. The optimal design of the components codebooks is a complex task, but a suboptimal solution clearly lowered the computational requirements. A tree-structure was also applied in (Kaarna et al., 2000) to accelarate the look-up functions in clustering. The leaves of the tree consisted of short linear lists and the search operation combined both the tree-structured and linear look-up functions.

An important feature in vector quantization is how to define an appropriate distortion measure for two vectors (Ryan & Arnold, 1997-2). The Euclidian distance between the two vectors X and Y is defined as

$$E = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{1}$$

where x_i and y_i are components of vectors X and Y, respectively. A drawback for the Euclidian distance is that it doesn't account for the various shapes of the vectors. The PMAD distortion measure was developed to guarantee that every pixel $B'(s_1,s_2,\lambda)$ of the compressed and reconstructed image is within a maximum distance of $p*100\%$ from its original value $B(s_1,s_2,\lambda)$, i.e. $(1-p)B(s_1,s_2,\lambda) < B'(s_1,s_2,\lambda) < (1+p)B(s_1,s_2,\lambda)$. Using this distortion measure, lossy compression ratios cr up to $cr=17$ were received with airborne multispectral images.

One large codebook can be replaced with two codebooks (Kamano et al., 2001). The first one, a relative small codebook, with few training sets was generated. The second codebook was generated from the residual data between the original image and the first codebook output. The proposed scheme improved the coding efficiency and reduced the transmission rate according to the numerical experiments.

2.2 Spectral decorrelation with PCA

In image compression, the principal component analysis (PCA) produces optimal results in the sense of the mean-square error reconstruction (Karhunen & Joutsensalo, 1995).
The principal component analysis is based on the covariance matrix $C= E[(x-\mu)(x-\mu)^T]$, $\mu= E[x]$ of the original data. In practical calculations the matrix C is replaced by an estimated $\hat{C}$

$$\hat{C} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu^*)(x_i - \mu^*)^T \tag{2}$$

where x_i is a sample vector and μ^* is the estimated mean vector of the sample set. The sum is over all the n samples of the set. From the estimated $\hat{C}$ the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ and the respective eigenvectors $u_1, u_2, ..., u_n$ are calculated. Due to the properties of the autocorrelation matrix, the eigenvalues λ_i , $i=1,n$ are all real and nonnegative.
As soon as the eigenvalues λ_i are known and without loss of generality the indexing is such that $\lambda_1 > \lambda_2 > ... > \lambda_n$, the reconstruction x^* of x is obtained as

$$x^* = \sum_{i=1}^{p}(x^T u_i)u_i \tag{3}$$

where p, $p < n$ is selected such that the required quality in reconstruction will be achieved. In Fig. 4 the principle of the PCA compression of spectral images is shown.

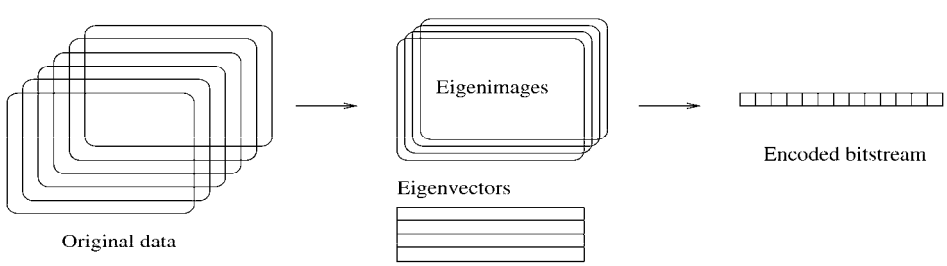


Fig. 4. The PCA in lossy compression of spectral images. In this example, the number p of eigenimages/eigenvectors is $p=4$.

2.3 Transform coding

Function transforms have been used for centuries to solve problems i.e. in mathematics, physics and engineering (Zayed, 1996). For example, in audio signal processing it would be interesting to know what frequencies are included in the measured signal. This problem can be solved using the Fourier transform.

In general, a transform is a mathematical operation where a function or data f in domain u is transformed into another function or data F in domain U: $f \rightarrow F$. The purpose of the transform is normally one of the following:

- After the transform it would be easier to solve the original problem.
- The transformed data gives a new insight to the problem at hand.
- The data in the domain $F(U)$ is measured experimentally and the function f needs to be constructed from this data.

The transforms $f \rightarrow F$ of any practical value has an inverse transform, where the original function f is completely constructed from F, i.e. $F \rightarrow f$. Thus, the transform pair is used to solve the original problem using data F in domain U, and then the solution is transformed back to the data f in domain u.

A popular transform in engineering is the Fourier transform. The transform was developed by Joseph Fourier in 1822 as he demonstrated, that most signals of practical interest can be expanded into a series of sinusoidal functions. Later on, this continuos transform has been developed to be applicable in discrete computations (Proakis & Manolakis, 1994).

The wavelet transform f^w of a function $f(t)$ also provides a time-frequency localization (Chui, 1992; Daubechies, 1988; Daubechies, 1992; Mallat, 1998; Vetterli & Kovačević, 1995) as

$$f^{w}(a,b) = |a|^{-1/2} \int f(t)\psi\left(\frac{t-b}{a}\right)dt \tag{4}$$

where ψ is called a mother wavelet with zero average, $\int\psi(t)dt = 0$. The mother wavelet $\psi(t)$ is defined as a double-indexed function as

$$\psi^{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{5}$$

Practical applications, like the signal compression, require fast implementations. In signal processing community, the wavelet transform is implemented with convolution as an filtering operation and the conjugate mirror filters are used as filter banks. The orthogonal wavelet transform is implemented by the cascading conjugate mirror filters. The perfect reconstruction is achieved with this implementation. The orthonormal bases of wavelets can be constructed using multiresolution analysis.

The fast discrete wavelet transform is computed using perfect reconstruction filter banks. Vetterli showed (Vetterli, 1986), that perfect reconstruction was always possible using FIR-filters. The multiresolution approximation lead to two discrete, finite length filters, and, thus, a filter bank was a solution to a fast implementation.

Using the definition

$$f(t) = \sum_{n=-\infty}^{\infty} a_0[n]\phi(t-n) \in V_0 \tag{6}$$

where $\phi(t)$ a is the scaling function, and due to the properties of multiresolution, $\{\phi(t-n)\}_{n \in Z}$ is orthonormal, then

$$a_0[n] = \langle f(t), \phi(t-n) \rangle \tag{7}$$

The approximation a_{j+1} in the next coarser level of the multiresolution is obtained by

$$a_{j+1}[p] = \sum_{n=-\infty}^{\infty} h[n-2p]a_j[n] \tag{8}$$

and the difference d_{j+1} between the two levels by

$$d_{j+1}[p] = \sum_{n=-\infty}^{\infty} g[n-2p]a_j[n] \tag{9}$$

where $g[n]$ is defined using the discrete filter $h[n]$ as

$$g[n] = (-1)^{1-n} h[1-n] \tag{10}$$

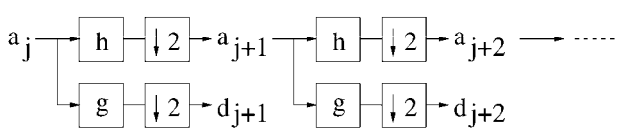
At the reconstruction of the data the coefficients are obtained as

$$a_j[p] = \sum_{n=-\infty}^{\infty} h[p-2n]a_{j+1}[n] + \sum_{n=-\infty}^{\infty} g[p-2n]d_{j+1}[n] \tag{11}$$

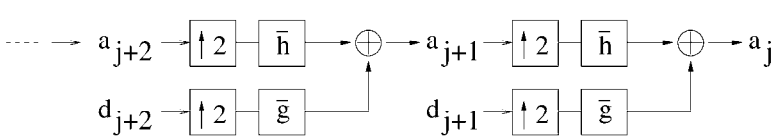
and finally, the discrete values f_d of the original function are recovered from

$$f_d[p] = \sum_{n=-\infty}^{\infty} a_0[n]\phi_d[p-n][n] \tag{12}$$

Since the scaling and the wavelet filters h ang g are finite, the infinite sums in Eqs. 8-12 are computed using the convolution. The discrete wavelet transform is illustrated in Fig. 5: part a) shows the transform and part b) the inverse transform.



a)



b)

Fig. 5. The discrete wavelet transform: a) the transform, b) the inverse transform.

The transform coefficients are the values of a_{j+N} and d_{j+i}, $i=1,...,N$. Downsampling by two (↓2) is performed in the transform and upsampling by two (↑2) in the inverse transform. In practice, Eq. 12 is not used and the values for a_0 are obtained directly as discretized values $f[n]$ of $f(t)$. Due to the perfect reconstruction property, the inverse transform returns the discretized values $f[n]$ directly as coefficients a_0.

The orthogonal, compactly supported wavelets described above has several enhancements. They include biorthogonal wavelets (Cohen et al., 1992), which allow symmetric wavelets.

This was a result of a modification in the multiresolution approximation. Another modification in the multiresolution lead to the wavelet packet analysis (Coifman & Wickerhauser, 1992). Wavelet packets are described as a full transform of the coefficients. Since in the original transform only the coefficients a_j are transformed, see Fig. 5, then in the wavelet packet analysis also the coefficients d_j are transformed in a similar way.

2.4 Linear and non-linear compression

The principal component analysis (PCA) is a widely used statistical technique in pattern recognition, image processing, and signal processing. PCA is an optimal solution in minimization of the mean-square representation error $E\{||x - x^*||^2\}$ where the data x is approximated using a lower dimensional linear subspace x^* (Karhunen & Joutsensalo, 1995). The principal component analysis provides orthonormal basis functions that optimally decorrelate the data. Other methods, like DCT or wavelets, approximate this optimal decorrelation. The justification for the wavelet transform in signal compression comes from the nonlinear approximation (Daubechies, 1998; Donoho et al., 1998; Devore et al., 1992), where the linear combination of N basis functions is used instead of the first N basis functions. In the linear approximation, the space S_n spanned by the first N basis functions $\boldsymbol{\Phi}_n$ is

$$S_n = \left\{ \sum_{n=1}^{N} c_n \Phi_n ; c_n \in C \right\} \tag{13}$$

and in the nonlinear approximation the space S_n is

$$S_n = \left\{ \sum_{n} c_n \Phi_n ; c_n \in C, \#\{n, c_n \neq 0\} \leq N \right\} \tag{14}$$

In nonlinear approximation the wavelet coefficients are ordered according to their significance and the most significant coefficients and their addressing are included in the bit stream.

2.5 Nonlinear compression through the three-dimensional wavelet transform

In two-dimensional case the construction of the wavelet transform starts from a tensor product of two one-dimensional multiresolution analyses (Daubechies, 1992; Mallat, 1989), $\boldsymbol{V_0} = V_0 \otimes V_0$, where V_j, $j \in Z$ is a multiresolution of $L^2(R)$. The multiresolution ladder is similar to that of one-dimensional case,, and now the multiresolution is

$$\begin{aligned} &(1) \quad \ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots \\ &(2) \quad V_0 = V_0 \otimes V_0 \\ &(3) \quad F \in V_j \Leftrightarrow F(2^j \cdot, 2^j \cdot) \in V_0, F(x_1, x_2) = f(x_1) f(x_2), f, g \in V_0 \end{aligned} \tag{15}$$

and the product

$$\Phi_{0,n,m}(x_1, x_2) = \phi_{0,n}(x_1)\phi_{o,m}(x_2) = \phi(x_1 - n)\phi_{0,m}(x_2 - m), n, m \in Z \tag{16}$$

is an orthonormal basis for $\boldsymbol{V_0}$. The basis for $\boldsymbol{V_j}$ is obtained (Mallat, 1998) as

$$\Phi_{j,n,m}(x_1,x_2)=\phi_{j,n}(x_1)\phi_{j,m}(x_2)=\frac{1}{2^j}\Phi(2^{-j}x_1-n)\Phi(2^{-j}x_2-m) \tag{17}$$

The orthogonal complement in V_{j-1} for V_j is W_j

$$\begin{aligned} V_{j-1} &= V_{j-1}\otimes V_{j-1}=\left(V_j\oplus W_j\right)\otimes V_j\oplus W_j \\ &= V_j\otimes V_j\oplus\left[\left(V_j\otimes W_j\right)\oplus\left(W_j\otimes V_j\right)\oplus\left(W_j\otimes W_j\right)\right] \\ &= V_j\oplus W_j \end{aligned} \tag{18}$$

and, thus, W_j consists of three parts, whose bases Ψ are combinations of one-dimensional scaling function ϕ and wavelet function ψ:

$$\begin{aligned} \Psi^h(x_1,x_2)&=\phi(x_1)\psi(x_2) \\ \Psi^v(x_1,x_2)&=\psi(x_1)\phi(x_2) \\ \Psi^d(x_1,x_2)&=\psi(x_1)\psi(x_2) \end{aligned} \tag{19}$$

The set $\{ \Psi^{\lambda}_{j,n};\ j \in Z,\ n \in Z^2,\ \lambda = h,v,d \}$ is an orthonormal basis for $L^2(R^2)$ (Daubechies, 1992). In this construction the sampling is done separately in vertical and horizontal directions, but the wavelet bases are nonseparable.

The fast two-dimensional wavelet transform is performed using filtering operations on vertical and horizontal dimensions of the image. The original image is filtered into quadrants and then the approximation quadrant is filtered further on. If the size of the original image in $N * N$ then each quadrant is of size $N/2 * N/2$. The transform has the perfect reconstruction property.

Similar approach as in the two-dimensional case gives the three-dimensional wavelets that are applied to the three-dimensional data like spectral images. If the separation of the spectral dimension is not applied, then the multiresolution analysis gives the configuration for the transform as

$$\begin{aligned} V_{j-1} &= V_{j-1}\otimes V_{j-1}\otimes V_{j-1} \\ &= \left(V_j\oplus W_j\right)\otimes\left(V_j\oplus W_j\right)\otimes\left(V_j\oplus W_j\right) \\ &= \left(V_j\oplus W_j\right)\otimes\left\{\left(V_j\otimes V_j\right)\oplus\left(V_j\otimes W_j\right)\oplus\left(W_j\otimes V_j\right)\oplus\left(W_j\otimes W_j\right)\right\} \\ &= \left(V_j\otimes V_j\otimes V_j\right)\oplus \\ &\quad \left\{\left(V_j\otimes V_j\otimes W_j\right)\oplus\left(V_j\otimes W_j\otimes V_j\right)\oplus\left(V_j\otimes W_j\otimes W_j\right)\oplus\right. \\ &\quad \left.\left(W_j\otimes V_j\otimes V_j\right)\oplus\left(W_j\otimes V_j\otimes W_j\right)\oplus\left(W_j\otimes W_j\otimes V_j\right)\oplus\left(W_j\otimes W_j\otimes W_j\right)\right\} \end{aligned} \tag{20}$$

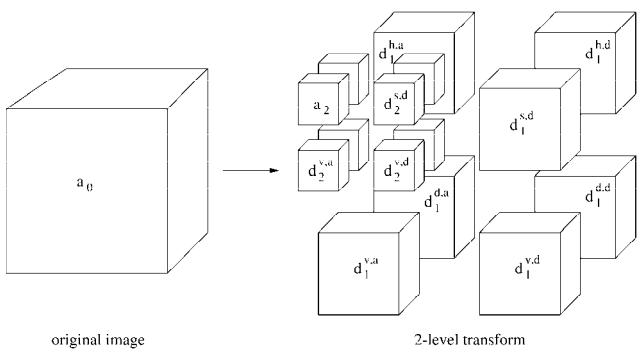
The scaling function for the basis V_0 is

$$\Phi_{0,n_1,n_2,n_3}(x_1,x_2,x_3)=\phi(x_1-n_1)\phi(x_2-n_2)\phi(x_3-n_3),\quad n_1,n_2,n_3\in Z \tag{21}$$

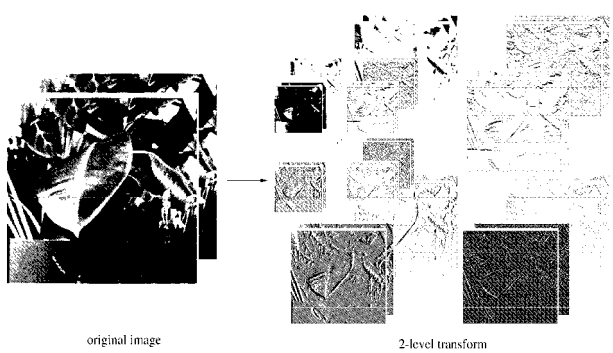
and the filtering of the spectral image is done using one scaling function and seven wavelets, which are defined as

$$
\begin{aligned}
\Phi^{s,a}(x_1,x_2,x_3) &= \phi(x_1)\phi(x_2)\phi(x_3)\\
\Psi^{h,a}(x_1,x_2,x_3) &= \phi(x_1)\phi(x_2)\psi(x_3)\\
\Psi^{v,a}(x_1,x_2,x_3) &= \phi(x_1)\psi(x_2)\phi(x_3)\\
\Psi^{d,a}(x_1,x_2,x_3) &= \phi(x_1)\psi(x_2)\psi(x_3)\\
\Psi^{s,d}(x_1,x_2,x_3) &= \psi(x_1)\phi(x_2)\phi(x_3)\\
\Psi^{h,d}(x_1,x_2,x_3) &= \psi(x_1)\phi(x_2)\psi(x_3)\\
\Psi^{v,d}(x_1,x_2,x_3) &= \psi(x_1)\psi(x_2)\phi(x_3)\\
\Psi^{d,d}(x_1,x_2,x_3) &= \psi(x_1)\psi(x_2)\psi(x_3)
\end{aligned}
\tag{22}
$$

where all dimensions are dilated similarly and the sampling is done separately along each dimension of the three-dimensional image. The original spectral image of size $N * N * N$ is filtered into octants of size N/2 * N/2 * N/2 as illustrated in Fig. 6.



a)



b)

Fig. 6. Three-dimensional wavelet transform applied twice. a) The principle of the transform, the coefficients *a* come from the low-pass filtering and the coefficients *d* from the high-pass filtering. b) Three-dimensional transform applied to one Bristol-image (Parraga et al., 1998).

Similar procedure will produce wavelets in higher dimensions than three. A theorem says (Mallat, 1998) that the family obtained by dilating and translating the 2^p-1 wavelets for $\alpha \neq 0$

$$\left\{ 2^{-\frac{pj}{2}} \Psi^{\alpha}\left(\frac{x_1 - 2^j n_1}{2^j}, \ldots, \frac{x_p - 2^j n_p}{2^j} \right) \right\}_{1 \le \alpha < 2^p, (j, n_1, \ldots, n_p) \in Z^{p+1}} \tag{23}$$

is an orthonormal basis for $L^2(R^p)$. The configuration of the three-dimensional transform in Eqs. 20, 21, and 22 is compatible with this theorem.
The multiwavelet based transform is slightly more complicated due to preprocessing, computations, and housekeeping (Kaarna & Parkkinen, 1999). This transform has similar variants as the scalar case above. The multiwavelet transform with two scaling functions compatible with Eq. 22, Fig. 6 would contain the coefficients of the first scaling function in the front part of each cubic block and the coefficients from the second scaling function in the back part of each cubic block. The similar division applies to the coefficients from the two wavelet functions.

2.6 Linear compression through spectral decorrelation and spatial compression

A reference method for image compression is based on the principal component analysis (PCA) as a spectral decorrelation method combined with a two-dimensional transform as the spatial compression method. Both the discrete cosine transform (Rabbani & Jones, 1991) and the wavelet transform are used as the spatial compression methods (Kaarna & Parkkinen, 2001). Also the discrete cosine transform has been enhanced to the hyperspectral images (Abousleman et al., 1995).
In Fig. 7 we illustrate the compression method, where the spectral decorrelation by PCA is followed the spatial wavelet transform.

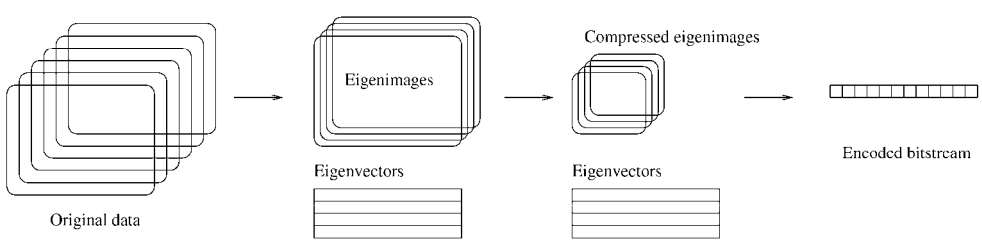


Fig. 7. PCA and the 2D wavelet transform in the lossy compression of spectral images.

The new image compression standard, JPEG2000 was basically defined only for colour images, but the Part 2, extensions, includes also transforms for multiple component imagery (Taubman & Marcellin, 2002). The linear block transform can be considered as a matrix multiplication as described in Section 2.2 as the PCA transform. The multiplication has an inverse operation and, thus, the data can be reconstructed. Also the wavelet transform is defined suitable for a point transform in JPEG2000. The work is currently underway, so the exact definitions are open (Taubman & Marcellin, 2002; JPEG2000, 2006).

2.7 Encoding in the Lossy Compression

The encoding phase of the compression is accomplished in a lossless manner, see Fig. 1. In encoding the quantized coefficients are coded to minimize the amount of data to be stored or transmitted. In addition, the new representation should contain the same information as the original data. In decoding the new representation is coded back to the original data. The encoding methods are divided into statistical methods and dictionary methods. In statistical methods, like static Huffman coding and arithmetic coding, the probabilities of the source symbols are required and two passes of the source are needed for encoding. In dictionary based methods, adaptive methods like the Ziv-Lempel-algorithms LZ77 and LZ78, only one pass is sufficient for encoding (Lelewer & Hirschberg, 1987; Ziv & Lempel, 1977; Ziv & Lempel, 1978).

In LZ77 (Ziv & Lempel, 1977), the source characters were encoded using a window of length *N*. The first *N-F* source symbols were already encoded and the last *F* source symbols constituted a lookahead buffer. The next source symbols in the lookahead buffer *F* were encoded by searching the longest match from the *N-F* source symbols in the window *N*. The match was coded using a pointer and the length of the match. In decoding, no search was needed, since the data was copied from the pointer position (Lelewer & Hirschberg, 1987; Ziv & Lempel, 1977). A modification to the previous method is the LZ78-encoding (Ziv & Lempel, 1978; Bell et al., 1989). Now the source symbols seen so far are split into phrases, where each phrase is the longest matching phrase seen so far plus one source symbol. Each phrase is coded as an index to its prefix plus the extra symbol. The new phrase is also added to the list of phrases that may be referenced. After the introduction of the original LZ-methods, there have appeared several enhancements and modifications to these methods, see e.g. (Bell et al., 1989; Lelewer & Hirschberg, 1987).

In arithmetic coding the source symbols are coded to a magnitude in range *[0,1)* (Langdon, 1984; Rissanen & Langdon, 1979). Initially, the range was split by the probabilities of single source symbols. New source symbols split the existing subranges in a similar way. Finally, all source symbols were manipulated and the subranges gave the codes. The encoding carried the prefix property. In decoding, the model of the source used by the encoder must be known. The encoded value was compared to the known probabilities in range *[0, 1)*, then in the subranges, and finally the decoder output the original source symbols.

SPIHT (Said & Pearlman, 1995) is an effective wavelet-based compression method for two-dimensional images. Color images are compressed through applying the method in each R, G, and B-band separately. For spectral images this approach has been extended to simultaneously manipulate all the bands of the spectral image (Dragotti et al., 2000). The approach combines the wavelet transform with the coding of the selected wavelet coefficients. The wavelet coefficients are coded using a hierarchical tree structure. In Fig. 8, the two-dimensional tree structure is depicted. In the three-dimensional case the structure is extended to include also the spectral dimension. Then the two-dimensional "squares" of coefficients become three-dimensional "cubics". The extension is analogious to that depicted in Fig. 6 for the wavelet transform.

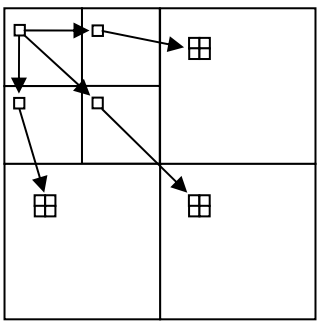


Fig. 8. The tree structure in 2D SPIHT.

2.8 Results from lossy compression

In the experiments we applied the integer PCA and wavelet transform to four AVIRIS images: Jasper Ridge, Moffet Field, Lunar Lake, and Cuprite (AVIRIS, 2006). The spatial size of each image was 608*512 and the number of bands was 224. The resolution of the original images was 16 bits. Thus, each image occupied 139,460,608 bytes of disk space in the raw form. Band 200 from Moffet Field image is displayed in Fig. 9.

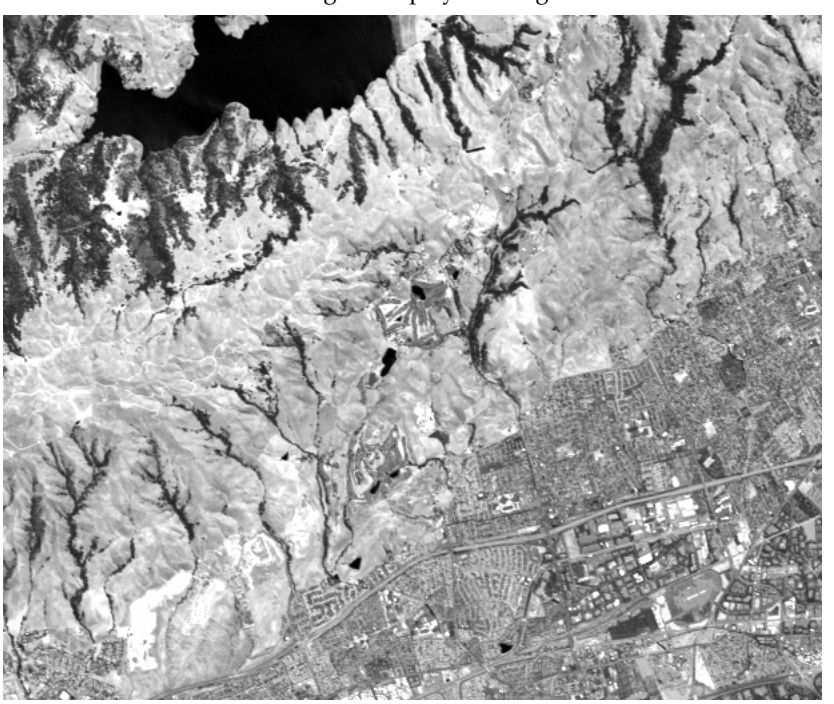


Fig. 9. Spectral band 200 from Moffet Field AVIRIS image.

In Fig. 10 the results from the PCA decorrelation with the 2D wavelet tranform are shown, see Fig 7. With PCA, the variable-bit-rate approach was also applied: the bit-allocation

between the eigenimages was entropy-based (Kaarna et al., 2006). The results from the 3D wavelet transform are also included, see Fig. 6. The horizontal axis is the compression ratio (CR) and the reconstruction quality as PSNR in *dB* (Eq. 28) is in the vertical axis.

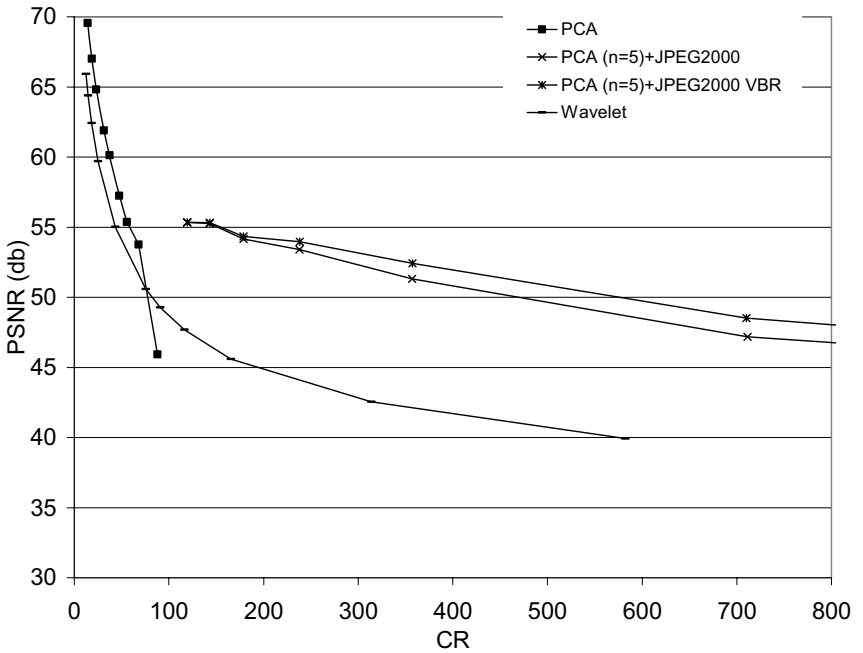


Fig. 10. Results from the lossy compression of AVIRIS spectral images. In PCA, *p=5* (Eq. 3).

3. Lossless compression of spectral images

The lossless compression is also called image coding when all the information in the image is included in the bit stream and the representation of the information is changed into a more compact form. In lossless compression of the spectral images the methods described in Section 2.7 can be used, since they are universal coding approaches for any digital information. These entropy-based textual coding approaches produced compression ratios from 1.5 to 2.0 for a set of AVIRIS-images (Kaarna & Parkkinen, 2001).

For spectral images, the lossless compression can be done by vector quantization and arithmetic coding (Ryan & Arnold, 1997-1). First, the vector quantization is applied to the spectra of the image; second, the residual image is created and then it is coded using arithmetic coding. The residual image contains only integer values obtained after rounding the difference between the closest vector center and the original spectral values. In addition, addresses from vector quantization and a set of other parameters are stored as side information. Also other lossless coding methods exist, e.g. they are based on band ordering (Tate, 1997; Toivanen et al., 2005) or spectral and spatial noncausal prediction (Memon et al., 1994). Thus, in the lossless compression of spectral images better results are achieved through the transform coding and especially, with the predictive coding.

3.1 Transform coding in lossless compression of spectral images

We applied the principal component analysis (PCA) to define the approximation image (Kaarna, 2001). PCA will produce a set of base vectors, which minimize the approximation error in the L^2-sense. The problem of heavy computations in PCA is solved by selecting only a small number of spectra from the image for the calculation of the base vectors. Also an integer version of PCA is needed. The calculation of the eigenvalues and eigenvectors is done with floating values, but the double precision results from the analysis are transformed to integer values such, that the required number of correct digits is maintained in the reverse transform. The PCA transform was described in Section 2.2.

Since the approximation image is available, then the residual image can be calculated. The residual image is then further compressed with the integer wavelet transform. Reversible integer-to-integer wavelet transforms have shown good performance in lossless colour and grey-scale image coding (Adams & Kossentini, 2000). The integer wavelet transform is based on the lifting scheme: different filters were derived by combining the prediction step with the update step (Calderbank et al., 1998). The integer wavelet transform is one-dimensional in nature. In the two-dimensional case, the one-dimensional transform is applied to the rows and columns of the image. In the three-dimensional case, the one-dimensional transform is applied to the spatial and spectral domains separately. The approach is the same as in the general case, see Fig. 8.

Similarly to the floating case, there exists different integer wavelet transforms (Adams & Kossentini, 2000; Calderbank et al., 1998; Daubechies, 1998). The oldest form of the integer wavelet transform subtracts the even samples from the odd samples to get the difference d_1 and the new approximation a_1 as

$$\begin{aligned} d_{1,l} &= a_{0,2l+1} - a_{0,2l} \\ a_{i,l} &= a_{0,2l} + \lfloor d_{1,l} / 2 \rfloor \end{aligned} \tag{24}$$

where the original data is stored in a_0. The second subscript refers to the index of the sample vector. The exact reconstruction comes from calculating the values in reverse order as

$$\begin{aligned} a_{0,2l} &= a_{1,l} - \lfloor d_{1,l} / 2 \rfloor \\ a_{0,2l+1} &= a_{0,2l} + d_{1,l} \end{aligned} \tag{25}$$

In general, the integer wavelet transform consists of the prediction and of the update based on the lifting where the number of vanishing moments is increased. In (Adams & Kossentini, 2000), the best lossless compression results for grey-scale images were obtained with the 5/3-transform, the forward 5/3-transform is defined as

$$\begin{aligned} d_{1,l} &= a_{0,2l+1} - \lfloor 1/2 \left(a_{0,2l+2} + a_{0,2l} \right) \rfloor \\ a_{i,l} &= a_{0,2l} + \lfloor 1/4 \left(d_{1,l} + d_{1,l-1} + 1/2 \right) \rfloor \end{aligned} \tag{26}$$

where a_0 refers to the even samples and d_0 to the odd samples of the original signal. We implemented also other integer wavelet transforms, but the final results were calculated with the 5/3-transform.

The zero order entropies of the AVIRIS test images, see section 2.8, are tabulated in Table 3, column ent_o. The test images were compressed with the lossless Burrows-Wheeler algorithm (Nelson, 1996) and the bitrates are in Table 3, the second column (Bit-rate). The compression

results with the integer PCA and wavelet approach are shown also in Table 3 (Kaarna, 2001). The compression ratio (CR) is a ratio between the size of the original file size and the size of file containing the encoded image. The bitrate is calculated as 16 bits/sample divided by the compression ratio (column Bit-rate). The zero order entropy of the residual image is tabulated before (ent_b) and after (ent_a) the integer wavelet transform. Also, the entropy of all encoded data is tabulated (ent_f). All entropies are expressed as bits/sample. The last column (ratio) contains the compression ratio as the ratio between the original entropy and the entropy of all encoded data (ent_f).

Image	ent_o	Bit-rate	ent_b	ent_a	ent_f	CR	Bit-rate	ratio
Jasper	11.19	7.94	6.14	5.24	5.62	2.83	5.65	1.99
Moffet	11.55	8.11	6.38	5.36	5.74	2.79	5.73	2.01
Lunar Lake	12.17	7.07	5.84	5.14	5.50	2.79	5.73	2.21
Cuprite	12.07	7.29	6.12	5.15	5.51	2.90	5.52	2.19

Table 3. Entropies and actual compression ratios for the four test images.

Our test images were from the AVIRIS free data set. They were measured in 1997 and most of the comparative results were calculated using older AVIRIS data sets. Thus, the comparisons will give only suggestions on the coding properties of our method. The results from (Ryan & Arnold, 1997-1) are collected in Table 4.

Image	Original entropy	Final entropy	ratio
Jasper	9.79	5.73	1.71
Moffet	9.64	5.63	1.71

Table 4. Results from (Ryan & Arnold, 1997-1).

In (Tate, 1997) the actual compression ratio for the data from a single AVIRIS image was 3.53. In (Memon, et al., 1994) the entropy of the residual of Cuprite image ranged from 5.48 to 5.61 bits/sample.

3.2 Predictive coding in lossless compression of spectral images

In (Mielikäinen et al., 2002; Mielikäinen et al., 2003) an interband version of predictive coding is presented. Linear prediction is one of the best performing image coding techniques. The least squares estimation approach defines the prediction coefficients from the causal set. An estimate $p'_{x,y,z}$ for the current pixel $p_{x,y,z}$ at location x,y,z is calculated as

$$p'_{x,y,z} = \sum_{k=0}^{O} \sum_{j=0}^{T} \sum_{i=L_{j,k}}^{R_{j,k}} a_{i,j,k}\, p_{x-i,y-j,z-k} \tag{27}$$

where $a_{i,j,k}$ is a prediction coefficient of the pixel at location i,j,k. O is the number of bands and T is the number of rows in the causal set. The $L_{j,k}$ and $R_{j,k}$ are the delimiters in spatial

and spectral dimensions for the causal set. These definitions lead to matrix operations and finally to the causal estimates of the pixel. Since the coefficients $a_{i,j,k}$ are known the estimate of the pixel $p_{x,y,z}$ is defined. The causal set was structured from the spatial and spectral dimensions. In the experiments a small causal set with prediction only from the previous band proved to give the best results. Also the heuristic for prediction was considered, some bands were not predicted but entropy coded without prediction. This enhancement further added the coding performance. The results presented outperformed the results found from the literature.

In Table 5 the average results for AVIRIS images are collected (Mielikäinen et al., 2003). They used the same image data as shown in Table 3. The table contains also reference results from vector quantization (VQ), enhanced principal component analysis with integer wavelets (PCA, see Table 3), the discrete cosine transform (DCT) and finally the results from the methods presented in (Mielikäinen et al., 2003), the prediction and the adaptive prediction (Pred/1, A&P/3).

Image	VQ	PCA	DCT	Pred/1	A&P/3
4 test images	3.06	3.03	2.72	3.14	3.23

Table 5. Results from (Mielikäinen et al., 2003).

A general conclusion from the previous is that the prediction methods work best in spectral image coding. The transform methods are more suitable for lossy compression.

4. Quality in lossy compression

The quality of the lossy compressed/reconstructed image is hard to evaluate. The error measures used in the lossy compression of the spectral images are similar to those used in the compression of the grey-scale or RGB-colour images: the error is evaluated using mean-square-error based quantitative measures like root-mean-square error, signal-to-noise ratio (SNR) or peak-signal-to-noise ratio (PSNR). All of these measures are computed pixelwise and thus, they show limited correlation with the human visual system. For example, the PSNR error remains the same, even though the relative error becomes large. This relative error is important in perceptual measures, since the human visual system notices the intensity variation in grey area better than in dark or bright area (Li et al., 1999).

Qualitative measures are becoming more important, web-based applications like e-commerce will require images with high visual quality.

4.1 Energy-based quality measures

For grey-level images the signal-to-noise ratio (SNR) and the peak-signal-to-noise ratio (PSNR) can be defined as

$$SNR = 10\log_{10}\frac{E^o}{\left|E^o - E^{cr}\right|}, \quad PSNR = 10\log_{10}\frac{N^2 s^2}{\left|E^o - E^{cr}\right|} \tag{28}$$

where s is the peak value of the image, normally $s = 2^8-1 = 255$, E^o is the energy of the original image, E^{cr} is the energy of the compressed/reconstructed image, and N^2 is the number of pixels in the image (Rabbani & Jones, 1991).
These measures have some advantages and some drawbacks. Both of the measures are computed pixel-wise and thus, they show poor correlation with human visual perception. One of the advantages is that these measures can be computed fast: the amount of computations needed is linearly dependent on the size of the spectral image, i.e. $O(n)$, where $n = N*N*M$, and N is the number of pixels in each spatial dimension and M is the number of the spectral bands. Also, any type of images can be used with this measure: greyscale, colour, or spectral images. The PSNR has a constant energy E^o for the images of equal size and thus it provides an absolute measure for the error. For example, if in one pixel image with the resolution of 8 bits, the pixel values are $x_i^o=5$, and $x_i^{cr}=3$, then $PSNR=42.1dB$. If a similar error, two units, is in the range closer to the peak value, like $x_i^o=251$ and $x_i^{cr}=249$, the PSNR remains the same, $PSNR=42.1dB$. Thus the PSNR does not notice the locations of the equal size errors in the intensities of the samples. These locations are important in perceptual measures, since the eyes notice the intensity variation in the grey area better than in the dark or in the bright areas (Li et al., 1999). Despite of this, the PSNR is widely used measure due to the absolute nature. The SNR measure uses energy E^o which is dependent on the values of the image, so the measure is a proportional measure. In the similar case as previously, the SNR measure has different values depending on the range of the pixel values. For example, if $x_i^o=5$, and $x_i^{cr}=3$ in a 8 bit resolution image, then $SNR=8.0dB$. If $x_i^o=251$ and $x_i^{cr}=249$, then $SNR=42.0dB$. For this reason, the SNR values cannot be compared between different sets of images without normalization.

4.2 Content difference-based quality measures

For the spectral image compression a quantitative measure based on the percentage maximum absolute distortion (PMAD) is developed (Ryan & Arnold, 1997-2). The PMAD is measured as a distance between each pixel from the original image and the reconstructed image and it guarantees, that every distance is below $p*100\%$ of the original pixel value. The measure showed predictable behaviour as the compression ratio increased and vice versa. The quality of the compression/reconstruction can be predicted, if the compression ratio is known in advance. The details of PMAD were already described in Section 2.1.

Similarly to PMAD, quality controlled compression methods were developed for near-lossless compression (Aiazzi et al., 2001). The data used was optical data, either pancromatic 2D data or hyperspectral 3D data. Also, psychophysically derived quantization in wavelet based compression has been considered (Ferguson & Allinson, 2002). Their method minimizes distortions and provides smooth perceived degradation for compressed images. Again, colour images were used in the experiments.
The pixel-wise error measures are good for random errors but not for structured or correlated errors (Franti, 1999; Miyahara et al., 1998; Nakauchi et al, 1998). Typical compression artefacts include blockiness, blurring, and jaggedness of the edges and they require spatial consideration, which is based on the original pixel values.
A sliding cube of size 3x3x3 was used to process the spectral image and three components were computed for each pixel: the contrast, the spatial and the spectral structure, and the number of different grey-levels (Kaarna & Parkkinen, 2002). The contrast measures how

each pixel differs from the background. The spatial structure, or the edges, of the image are blurred or jagged in the compression. The number of different grey-levels in a block measures blockiness (Eskicioglu & Fisher, 1995; Franti, 1999; Miyahara et al., 1998). The final measure, the Blockwise Distortion Measure for Multispectral images (BDMM), was calculated using these three components from the original image and from the compressed/reconstructed image. The matching of the BDMM to the visual tests was done with a neural network.

The contrast is a local change in brightness and it is computed using standard deviation (Sonka et al., 1993). The first error component, the contrast error e_c for a block was computed using the difference of the standard deviations for the blocks from the original and the compressed/reconstructed image (Kaarna & Parkkinen, 2002). The spatial and spectral structure is the response to edge detection operations in a block (Sonka et al., 1993). For a three-dimensional 3x3x3 block the three edge detectors G_i, $i=x,y,z$ were filtering operations adopted from Laplacian edge-detectors by modifying the two-dimensional detectors to three-dimensional ones. The second error component, the error e_s in the spatial structure was a sum of all the edge-detection operations normalized with the contrast value of the block (Kaarna & Parkkinen, 2002). The third error component, the quantization error e_q was based on the number of different grey-levels in a block, both from the original image and from the compressed/reconstructed image. The total errors E_c , E_s , and E_q between the original and reconstructed images were received by computing the average values of the blockwise errors e_c , e_s , and e_q over the entire images (Kaarna & Parkkinen, 2002).

The matching between the computed values of the distortion measure BDMM and the visual tests was obtained using a multi-layer perceptron with back-propagation (Kaarna & Parkkinen, 2002). The problem is a curve-fitting problem with three input variables E_c , E_s , and E_q and the goal received from the visual tests V, $BDMM = f(E_c , E_s , E_q , V)$. The function f was modelled by a three-layer neural network with 6 neurons in the hidden layer and one neuron in the output layer. The function f was obtained as the network weights and biases, which are then used to compute the BDMM for all images used in the experiments.

In the experiments three spectral images were used, AISA, AVIRIS, and BRISTOL. Each image was of size 256x256 pixels and they had 32 spectral bands. The spectral range of the AISA (Airborne Imaging Spectrometer for Applications) was from 649 to 747 nm (Aisa, 2006). Our test image contains mainly vegetation. For AVIRIS-image, 32 channels from 1 to 218 by step 7 were selected. Thus, the spectral range is from 370nm to 2450nm. The image is part of one of the Moffet field-images (AVIRIS, 2006). The BRISTOL-image is taken in laboratory conditions from a flower leaf using the visible spectral range from 400 nm to 700 nm (Parraga et al., 1998).

In the experiments three methods were applied to compress the three test images. The three-dimensional wavelet transform with multi-wavelet kernel, the PCA spectral compression and the two-dimensional DCT/JPEG compression were applied. The multispectral images were compressed with several compression ratios and their visual quality was assessed by 18 subjects. After the matching the correlation coefficients for the three different images were computed: for AISA, 0.9911; for AVIRIS, 0.9836; and for BRISTOL, 0.9943. In Fig. 10 we visualize the correlation of the visual grading and the results from the filtering operations after matching.

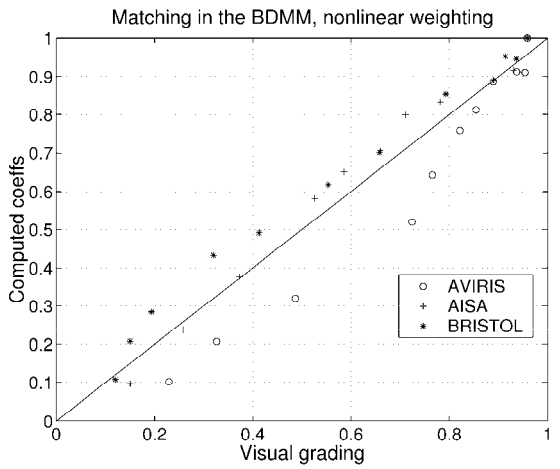


Fig. 10. Visual grading versus filtering operations after matching.

4.3 Spectral–based quality measures

In classification problems, for example in detection of minerals, classification of fields or in environmental monitoring, it is important to find exact matches between two spectra. Thus, comparison of data in vector format is required. Normally, the difference between two vectors is defined using the Euclidean distance (Kaarna et al., 2006).

The Euclidean distance d_e, defined in Eq. 1, measures only the difference in magnitudes between the two spectra and it doesn't observe the shape of spectra. The Euclidean distance is zero for two equal spectra and large values mean large differences in magnitudes of spectra.

Similar vectors have identical magnitudes and directions. The Spectral Similarity Value (SSV) includes these two metrics (Granahan & Sweet, 2001). SSV was defined as

$$SSV = \sqrt{d_e^2 + r_1^2} \tag{29}$$

For Eq. 29, the modified Euclidean distance was defined and the factor $1/n$ was inserted under the square root. n is the number of spectral bands in the hyperspectral image. Because a metric, whose large values meant dissimilar vectors, was needed, the coefficient r_1^2 was defined as $r_1 = 1 - r^2$, where r is the correlation between the vectors x and y

$$r^2 = \left(\frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y} \right) \tag{30}$$

where μ_x and μ_y are the mean values for vectors x and y. Respectively, σ_x and σ_x are standard deviations for vectors x and y. The range of r is between zero and one. SSV is zero for identical vectors and larger values mean more dissimilar spectra.

Spectral Angle Mapper (SAM) (Chang, 2000) calculates the angle between two spectra. The SAM only measures the shape of two spectra and it doesn't observe the difference in

magnitudes. The SAM value is zero for similar spectra and larger for more dissimilar spectra. The Spectral Angle Mapper value was defined as

$$SAM = \arccos\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}\right) \tag{31}$$

In the experiments the Euclidean distance, SSV and SAM were used for spectral matching after the original image was clustered into groups. The vectors of the original image were matched to the cluster centres and the spectral matching was applied also to the vectors of a compressed image. After that the results were compared. In optimum all pixels should have gone to the same clusters in both cases. However, the information loss in the lossy compression normally results in classification inaccuracies, which are lower than 100%. Classification accuracy was used to measure the image quality of a compressed image.
The Euclidian distance performed similarly with the two test images. SSV was more vulnerable than SAM in compression with the image with higher standard deviation. When the image contained larger spatial equi-value areas, the situation was vice versa. The approach of defining the spectral differences was reasonable but it still requires more research.

5. Conclusions

In this section we have collected experiences when different spectral images were compressed in a lossy manner with various methods described in the previous sections.
The following abbreviations are used:

- CL-W: wavelet transform in the spectral reduction followed by clustering,
- CL-P: PCA in the spectral reduction followed by clustering,
- WT-3M: the three-dimensional wavelet transform, Chui-Lian multiwavelets (Chui & Lian, 1996),
- WT-3H: the three-dimensional wavelet transform, Haar wavelet,
- SP-P: PCA in the spectral reduction and SPIHT (Said & Pearlman, 1996) in the spatial dimensions,
- JP2K-P: PCA in the spectral reduction and JPEG2000 (Taubman & Marcellin, 2002) in the spatial dimensions,
- JPG-P: PCA in the spectral reduction and DCT/JPEG (Rabbani & Jones, 1991) in the spatial dimensions,
- SP-O: SPIHT in the spatial dimensions, no spectral reduction,
- JPG-O: DCT/JPEG in the spatial dimensions, no spectral reduction,
- JP2K-O: JPEG2000 in the spatial dimensions, no spectral reduction.

In earlier experiments (Kaarna et al., 2000), it was found, that PCA and wavelets performed best with clustering, and, thus, from the comparisons we leave out the other possible variations.
The last three methods provided trivial solutions to the compression of spectral images. These methods applied SPIHT, DCT/JPEG, or JPEG2000 to the spatial dimensions of the images without any compression in the spectral dimension. Thus, we could get some

indication of the spectral redundancy, and we could also compare the two-dimensional compression methods to each other. The wavelet based compression methods, like the set partitioning in hierarchical trees (SPIHT) and JPEG2000 are effective definitions and implementations of a two-dimensional wavelet compression technique. DCT/JPEG is not of as high quality, but there has been progress also with the discrete cosine transform (Ponomarenko et al., 2005). They have extended the original 8*8 block size to 32*32 block size and carefully considered the quantization of the DCT coefficients.

In spectral image compression one has to consider the noise from the imaging system. The apparent noise is most often modelled as additive noise and there are automatic methods for removing this kind of noise resulting to high quality, noise free images (Ponomarenko et al., 2006). In this case, if the image compression is lossless, one can consider the approach as near-lossless, the loss comes from removing the noise, not any part of the information. If the noise removal should be automatic, then the noise model should well match to the imaging system and the algorithms should be carefully designed and implemented.

In the experiments we had totally 65 images from three data-sets, AISA, AVIRIS and BRISTOL, see section 4.2. Every image was compressed with the methods described above. All the experiments were performed in Matlab-environment.

Comparison to the references in the literature is not straight-forward, since various images and quality measures have been used in different studies. Thus, similar exact CR/PSNR comparison results cannot be presented as are presented for the standard RGB colour images like Lena-image.

In Tables 6 and 7 we give general comments on the compression methods described. The tables contain the summary of our experiments. In the tables a minus sign means that the approach has a bad property, doubled or tripled minus signs mean even worse property value. A plus sign means a positive property value. In Table 7, the compression quality is a combination of the compression ratio and the respective reconstruction quality. This general evaluation is based on the detailed errors included also in the table.

Method	Spectral compression	complexity	Independency of the eimage
SP-O	No	-	+
JPG-O	No	-	+
JP2K-O	No	-	+
CL-P	Yes	- -	-
CL-W	Yes	-	-
SP-P	Yes	- -	-
JPG-P	Yes	- -	-
JP2K-P	Yes	- -	-
WT-3M	Yes	- - -	+
WT-3H	Yes	- -	+

Table 6. Summary of the experiments with different compression methods for spectral images, compression features. - means poor, + means good value of property, multiple symbols mean stronger emphasis.

Method	Compression ratio	Blocking artefacts	Ringing artefacts	Structural errors	Compression quality
SP-O	- -	+	- -	+ + +	+
JPG-O	- - -	- - -	+	- -	-
JP2K-O	- -	+	- -	+ + +	+
CL-P	+ +	- -	+	+ +	+
CL-W	+	- -	+	+ +	-
SP-P	+ + +	+ +	-	+ + +	+ + +
JPG-P	+ +	- -	+	- -	+
JP2K-P	+ + +	+ +	-	+ + +	+ + +
WT-3M	+ +	+	-	+	+ +
WT-3H	+ +	+	- -	+	+

Table 7. Summary from the experiments with different compression methods for spectral images, compression ratio and the quality of compression. - means poor, + means good value of property, multiple symbols mean stronger emphasis.

Currently, the up-to-date system for the lossy compression of the spectral images contains the principal component analysis for the decorrelation of the spectral domain. Then this is followed by a two-dimensional transform for compression the eigenimages. Currently, the wavelet transform is the up-to-date choice for the two-dimensional compression. In the lossless case, the prediction based approaches produce the best coding results.

6. References

Abousleman, G. P.; Gifford, E. & Hunt, B. R. (1994). Enhancement and Compression Techniques for Hyperspectral Data, *Optical Engineering*, Vol. 33, No. 8 (Aug. 1994), pp. 2562—2571.

Abousleman, G. P.; Lam, T. T. & Karam L. J. (2002). Robust Hyperspectral Image Coding with Channel-Optimized Trellis-Coded Quantization, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40, No. 4 (Apr. 2002) pp. 820—830.

Abousleman, G. P.; Marcellin, M. W. & B.R. Hunt, B. R. (1995). Compression of Hyperspectral Imagery Using the 3-DCT and Hybrid DPCM/DCT, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 33, No. 1, (Jan. 1995), pp. 26—34.

Abousleman, G. P.; Marcellin, M. W. & Hunt, B. R (1997). Hyperspectral Image Compression Using Entropy-Constrained Predictive Trellis Coded Quantization, *IEEE Transactions on Image Processing*, Vol. 6, No. 4, (1997) , pp. 566—573.

Adams, M.D. & Kossentini, F. (2000) Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis, *IEEE Transactions on Image Processing*, Vol. 9, No. 6 (June 2000), pp. 1010—1024.

Aiazzi, B.; Alba, P.; Alparone, L. & Baronti, S. (1999). Lossless Compression of Multi/Hyper-Spectral Imagery Based on a 3-D Fuzzy Prediction, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37, No. 5 (Sep. 1999), pp. 2287—2294.

Aiazzi, B.; Alparone, L. & Baronti, S. (2001). Near-Lossless Compression of 3-D Optical Data, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 11 (Nov. 2001), pp. 2547—2557.

Aiazzi, B.; Alparone. L. & Baronti, S. (2002). Context Modeling for Near-Lossless Image Coding, *IEEE Signal processing Letters*, Vol. 9, No. 3 (March 2002), pp. 77--80.

Aisa Eagle (2006). AISA Eagle Data Sheet. Specim. http://www.specim.fi. Accessed 27.12.2006.

AVIRIS (2006). AVIRIS Home Page, National Aeronautics and Space Administration, Jet Propulsion Laboratory. http://makalu.jpl.nasa.gov/ aviris.html, accessed Dec. 27, 2006.

Aware (2006). Aware JPEG2000-3D: Compression of volumetric medical image data using JPEG2000-Part 2. http://www.aware.com/products/compression/J2K3D.html. Accessed Dec. 28, 2006.

Bell, T.; Witten, I. H. & Cleary, J. G. (1989). Modeling for data compression, *ACM Computing Surveys*, Vol. 21, No. 4 (Dec. 1989), pp. 557—591.

Berger, T. & Gibson, J. D. (1998). Lossy Source Coding, *IEEE Transactions on Information Theory*, Vol. 44, No. 6 (Oct. 1998), pp. 2693—2723.

Benazza-Benyahia, A.; Pesquet, J.-C. & Hamdi, M. (2001). Lossless Coding for Progressive Archival of Multispectral Images, *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2001), pp. 1817—1820.

Calderbank, A.R.; Daubechies, I.; Sweldens, W. & Yeo, B-L. (1998) Wavelet transforms that map integers to integers, *Applied and Computational Harmonic Analysis*, No. 5 (1998), pp. 332—369.

Chang, C.-I. (2000). An Information-Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image Analysis, *IEEE Transactions on Information Theory*, Vol. 46, 2000, pp. 1927—1932.

Chui, C. K. (1992). *An Introduction to Wavelets,* Academic Press, USA, 1992.

Chui, C. K. & Lian, J. (1996) A Study of Orthonormal Multi-wavelets, *Journal of Applied Numerical Mathematics*, Vol. 20 (1996), pp. 273—298.

Cohen, A.; Daubechies, I. & Feauveau, J.-C. (1992). Biorthogonal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, Vol. 45, 1992, pp. 485—560.

Coifman, R. R. & Wickerhauser, M. W. (1992). Entropy-based algorithms for best basis selection, *IEEE Transactions on Information Theory*, Vol. 38, No. 2 (March 1992), pp. 713--718.

Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, Vol. 41, No. 7 (Oct. 1988), pp. 909—996.

Daubechies, I. (1992) *Ten Lectures on Wavelet*s, CBMS-NSF, Regional Conference Series in Applied Mathematics, 61, SIAM, USA (1992).

Daubechies, I. (1998) Recent results in wavelet applications, *Journal of Electronic Imaging,* Vol. 7, No. 4 (Oct 1998), pp. 719—724.

DeVore, R. A.; Jawert, B.; Lucier, B. L. (1992). Image compression through wavelet transform coding, *IEEE Transactions on Information Theory*, Vol. 38, No. 2 (Oct. 1992), pp. 719—746.

Donoho, D. L.; Vetterli, M.; DeVore, R. A. & Daubechies I.(1998). Data Compression and Harmonic Analysis, *IEEE Transactions on Information Theory*, Vol. 44, No. 6 (Oct. 1998), pp. 2435—2476.

Dragotti, P. L.; Poggi, G. & Ragozini, R. P. (2000) Compression of hyperspectral images by three-dimensional SPIHT algorithm, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 38 (2000), pp. 416—428.

EI (2006). IS&T/SPIE Electronic Imaging. http://electronicimaging.org. Acccessed Dec 27, 2006.

Eskicioglu, A. M. & Fisher, P. S. (1995) Image Quality Measures and Their Performance, *IEEE Transactions on Communications*, Vol. 43, No. 12 (1995), pp. 2959—2965.

Ferguson, K. L. & Allinson, M. N. (2002) Psychophysically derived quantization model for efficient DWT coding, *IEE Proceedings -- Visual, Image and Signal processing*, Vol. 149, No. 1 (2002), pp. 51—56.

Fränti, P. (1999) Blockwise distortion measure for statistical and structural errors in digital images, *Signal Processing: Image Communication*, Vol. 13 (1998), pp. 89—98.

Gelli, G. & Poggi, G. (1999). Compression of Multispectral Images by Spectral Classification and Transform Coding, *IEEE Transactions on Image Processing*, Vol. 8, No. 4, (1999) pp. 476—489.

Granahan, J. C. & Sweet, J. N. (2001). An Evaluation of Atmospheric Correction Techniques Using the Spectral Similarity Scale, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS'01*, Sydney, Australia, 2001, vol. 5, pp. 2022—2024.

Gray, R. M. & Neuhoff, O. L. (1998). Quantization, *IEEE Transactions on Information Theory*, Vol. 44, No. 6 (Oct. 1998), pp. 2325—2383.

Grochowski, E. & Halem, R. D. (2003). Technological impact of magnetic hard disk drives on storage systems. *IBM Systems Journal,* Vol. 42, No. 2 (2003), pp. 338—346.

Hauta-Kasari, M.; Miyazawa, K.; Toyooka, S. & Parkkinen, J. (1999). Spectral Vision System for Measuring Color Images. *The Journal of the Optical Society of America A,* Vol. 16, No. 10, (1999), 2352—2362.

Hughes, G. F. (2002). Computers: wise drives, *IEEE Spectrum,* Vol. 39, No. 8 (Aug. 2002), pp. 37—41.

HYDICE (Hyperspectral Digital Imagery Collection Experiment) (2006). Goodrich Corporation. http://www.oss.goodrich.com. Accessed Dec. 27, 2006.

HyMap Airborne Scanners (2006). Integrated Spectronics. http://www.intspec.com. Accessed Dec. 27, 2006.

Hyvärinen, T.; Herrala, E. & Dall'Ava, A. (1998). Direct Sight Imaging Spectrograph: a Unique Add-on Component Brings Spectral Imaging to Industrial Applications, *Proc. of SPIE Conference on Digital Solid State Cameras: Designs and Applications,* SPIE Vol. 3302 (Apr. 1998), 165—175.

Hyperion (2006). National Aeronautics and Space Administration, Earth Observing-1, Hyperion Instrument. http://eo1.gsfc.nasa.gov. Accessed Dec. 27, 2006.

IGARSS (2006). IEEE International Geoscience and Remote Sensing Symposium. http:// www. grss-ieee.org. Accessed Dec 27, 2006.

Ikonos (2006). Space Imaging. IKONOS earth imaging satellite. http://www.spaceimaging. com. Accessed Dec. 27, 2006.

JPEG2000 (2006). JPEG 2000 3D (Part 10 –JP3D). http://www.jpeg.org/jpeg2000/j2part10.html. Accessed Dec 28., 2006.

Kaarna, A. (2001). Integer PCA and wavelet transforms for multispectral image compression *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'01,* Vol. 4 (July, 2001), pp. 1853—1855.

Kaarna, A. & Parkkinen, J. (1999). Multiwavelets in Spectral Image Compression, *Proc. 11'th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 7--11, 1999, pp. 327—334.

Kaarna A., Parkkinen, J. (2001) Transform Based Lossy Compression of Multispectral Images, *Pattern Analysis & Applications*, Vol. 4, No. 1, 2001, pp. 39—50.

Kaarna, A. & Parkkinen, J. (2002). Quality metric for multispectral image compression, *Journal of the Imaging Society of Japan*, Vol. 41, No. 4 (2002), pp. 379—391.

Kaarna, A.; Toivanen, P. & Keränen, P. (2006). Compression and Classification Methods for Hyperspectral Images, *Pattern Recognition and Image Analysis*, Vol. 16, No. 3 (Sept. 2006), pp. 413—424.

Kaarna, A.; Zemcik, P.; Kälviäinen, H. & Parkkinen, J. (1998). Multispectral image compression, *14th International Conference on Pattern Recognition,* Vol. 2, 16-20 Aug. 1998, Brisbane, Australia, pp. 1264—1267.

Kaarna, A.; Zemcik, P.; Kälviäinen, H. & Parkkinen, J. (2000). Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 38, No. 2 (March 2000) pp. 1073—1082.

Kamano, A.; Morimoto, M. & Nagura, R. (2001). Multispectral Image Compression using Hierarchical Vector Quantization, *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'2001*, Vol. 4, 9-13, July, Sydney, Australia, pp. 1856—1858.

Karhunen, J. Joutsensalo, J. (1995). Generalizations of Principal Component Analysis, Optimization Problems, and Neural Networks, *Neural Networks*, Vol. 8, N0. 4, 1995, pp. 549—563.

Kerekes, J. P. & Baum, J. E. (2002). Spectral Imaging System Analytical Model for Subpixel Object Detection, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40, No. 5 (May 2002), pp. 1088—1101.

Kälviäinen, H.; Kukkonen, S.; Hyvärinen, T. & Parkkinen, J. (1998). Quality Control in Tile Production, *Proc. the SPIE Conference on Intelligent Robots and Computer Vision* XVII, SPIE Vol. 3522 (1998), pp. 355—365.

Landsat (2006). National Aeronautics and Space Administration, Goddard and U.S. Geological Survey. http://landsat.gsfc.nasa.gov. Accessed Dec 27, 2006.

Langdon Jr., G. G. (1984) An introduction to arithmetic coding, *IBM Journal of Research and Developmen*t, Vol. 28, No. 2 (March 1984), pp. 135—149.

Lelewer, D. A. & Hirschberg, D. S. (1987). Data compression, *ACM Computing Surveys*, Vol. 19, No. 3 (Sept. 1987), pp. 261—296.

Lillesand, T. M. & Kiefer, R. W. (2000). *Remote Sensing and Image Interpretation*, John Wiley & Sons, New York, USA.

Li, J.; Chaddha, N. & R.M. Gray, R. M. (1999) Asymptotic Performance of Vector Quantizers with a Perceptual Distortion Measure, *IEEE Transactions on Information Theory*, Vol. 45, No. 4 (1999), pp. 1082—1091.

Mallat, S. (1989) Multiresolution approximation and wavelet orthonormal bases of $L^2(R)$, Transactions of American Mathematical Society, Vol. 315 (Sep. 1989), pp. 69—87.

Mallat, S. (1998) *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, USA, 1998.

MCS (2006). International Symposium on Multispectral Color Science. http://www.multispectral.org. Accessed Dec. 27, 2006.

Memon, N.D.; Sayood, K. & Magliveras, S. S. (1994). Lossless Compression of Multispectral Image Data, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 32, No. 2, (1994) pp. 282—289.

Mielikäinen, J. (2006). Lossless compression of hyperspectral images using lookup tables, *IEEE Signal Processing Letters,* Vol. 13, No. 3 (March 2006), pp. 157—160.

Mielikäinen, J.; Kaarna, A. & Toivanen, P. (2002) Lossless hyperspectral image compression via linear prediction, *Proceedings of Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII*, SPIE 4725, Orlando, USA, April 1-5, 2002, pp. 600—608.

Mielikäinen, J.; Toivanen, P. & Kaarna, A. (2003) Linear prediction in lossless compression of hyperspectral images, *Optical Engineering,* Vol. 42, No. 4 (April 2004), pp. 1013—1017.

Mielikäinen, J. & Toivanen P. (2003). Clustered DPCM for the lossless compression of hyperspectral images, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 41, No. 2 (Dec. 2003), pp. 2943—2946.

Miyahara, M.; Kotani, K. & Algazi, V. R. (1998) Objective Picture Quality Scale (PQS) for Image Coding, *IEEE Transactions on Communications*, Vol. 46, No. 9 (1998), pp. 1215—1226.

Moreiro, F. (2006) STORAGE: Ten-year forecast of storage evolution, Deliverable D12.5, Presto-Space, 2006.

Nakauchi, S.; Hatanaka, S. & Usui, S. (1998) Color gamut mapping by minimizing perceptual differences between images, *Systems and Computers in Japan*, Vol. 29, Issue 10 (1998), pp. 46—56.

Nelson, M. (1996). Data compression with the Burrows-Wheeler transform, *Dr. Dobbs Journal.* http://dogma.net/markn/articles/bwt/bwt.htm, accessed 1.1.2007.

OrbView (2006). Orbimage, OrbView-3 satellite. http://www.orbimage.com. Accessed Dec. 27, 2006.

Parraga, C. A.; Brelstaff, G.; Troscianko, T. & Moorehead, I. R. (1998) . Color and luminance information in natural scenes, *Journal of the Optical Society of America*, JOSA A, Vol. 15, Issue 3 (1998), pp. 563—569.

Poggi, G. & Ragozini, A. R. P. (2002). Tree-structured product-codebook vector quantization, *Signal Processing: Image Communication*, Vol. 16, 2002, pp. 421—430.

Ponomarenko, N.N.; Lukin, V. V.; Egiazarian, K. O. & Astola, J. T. (2005) DCT Based High Quality Image Compression, *Proc. Scandinavian Conference on Image Analysis, Springer Series: Lecture notes in computer science*, Vol. 3540, Joensuu, Finland, June 2005, pp. 1177—1185.

Ponomarenko, N.; Lukin, V.; Zriakhov, M. & A. Kaarna (2006). Preliminary Automatic Analysis of Characteristics of Hypespectral Aviris Images, *International Conference on Mathematical Methods in Electromagnetic Theory*, 26-29 June, 2006, pp. 158—160.

Proakis, J. G. & Manolakis, D. G. (1994). Digital Signal Processing: Principles, Algorithms, and Applications, Macmillan, New York, 1994.

Rabbani, M. & P. W. Jones, P. W. (1991). *Digital Image Compression Techniques*, SPIE - Tutorial Text Series, Volume TT 7, Bellingham, WA, USA.

Rissanen, J. & Langdon Jr., G. G. (1979) Arithmetic coding, *IBM Journal of Research and Development*, Vol. 23, No. 2 (March 1979), pp. 149—162.

Roger, R. E. & Cavenor, M. C. (1996). Lossless compression of AVIRIS images, *IEEE Transactions on Image Processing*, vol. 5, no. 5, (1996) pp. 713—719.

Ryan, M. & Arnold, J. (1997-1) The lossless compression of AVIRIS images by vector quantization, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 35, No. 3 (May 1997), pp. 546—550.

Ryan, M. J. & Arnold, J. F. (1997-2). Lossy Compression of Hyperspectral Data Using Vector Quantization, *Remote Sensing of Environment*, Vol. 61 (Mar. 1997), pp. 419—436.

Ryan, M. J. & Arnold, J. F. (1998). A Suitable Distortion Measure for the Lossy Compression of Hyperspectral Data, *Proc. the IEEE International Geoscience and Remote Sensing Symposium IGARSS'98*, Vol. 4 (July 1998), pp. 2056—2058.

Said, A. & Perlman, W. A. (1996) A new, fast, and efficient image codec based on set partioning in hierarchical trees, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 3 (June 1996), pp. 243—250.

Sonka, M.; Hlavac, V. & Boyle, R. (1993) *Image Processing, Analysis and Machine Vision*, Chapman & Hall Computing, Cambridge University Press, England.

Tate, S.R. (1997) Band ordering in lossless compression of multispectral images, *IEEE Transactions on Computers*, Vol. 46, No. 4 (1997), pp. 477—483.

Taubman, D. S. & Marcellin, M. W. (2002). *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, USA.

Thompson, D. A. & Best, J. S. (2000). The future of magnetic data storage technology, *IBM Journal of Research and Development*, Vol. 44, No. 3 (2000), pp. 311—322.

Toivanen, P.; Lehtinen, A.; Ansamäki, J. and Kälviäinen, H. (1999). Two-Stage Multispectral Image Compression Using the Self-Organizing Map, *Proceedings of the 11th Scandinavian Conference on Image Analysis (SCIA'99)*, Kangerlussuaq, Greenland, June 7-11, 1999, pp. 903—909.

Toivanen P.; Kubasova, O. & Mielikäinen, J. (2005). Correlation-based band ordering heuristics for lossless compression of hyperspectral sounder data. *IEEE Transactions on Geoscience and Remote Sensing Letters*, Vol. 2, No. 1 (Jan. 2005), pp. 50—54.

Vaughn, V.D. & T.S. Wilkinson, T. S (1995). System Considerations for Multispectral Image Compression Designs, *IEEE Signal Processing Magazine*, (Jan 1995), pp. 19—31.

Vetterli, M. & Kovačević, J. (1995) *Wavelets and Subband Coding*, Prentice Hall, USA, 1995.

Vetterli, M., Filter banks allowing perfect reconstruction, *Signal Processing*, Vol. 10, No. 3 (Apr. 1986), pp. 219—244.

Zayed, A.I. (1996). Handbook of function and generalized function transformations, CRC Press, USA, 1996.

Ziv, J. & Lempel, A. (1977). A universal algorithm for sequential data compression, *IEEE Transactions on Information Theory*, Vol. IT-23, No. 3 (May 1977), pp. 337—343.

Ziv, J., Lempel, A. (1978). Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory*, Vol. IT-24, No. 5 (Sept. 1978), pp. 530—536.

15

Data Fusion in a Hierarchical Segmentation Context: The Case of Building Roof Description

Frédéric Bretar
Institut Géographique National – MATIS laboratory
France

1. Introduction

Automatic mapping of urban areas from aerial images is a challenging task for scientists and surveyors because of the complexity of urban scenes. The 2D image information can be converted into 3D points provided that aerial images have been acquired in a (multi-) stereoscopic context (Kasser & Egels, 2002). Altitudes are then processed using automatic correlation algorithms to generate Digital Surface Models (DSM) (Pierrot-Deseilligny & Paparoditis, 2005), (Baillard & Dissard, 2000). The DSM helps in the understanding of an urban scene, especially for the 3D building reconstruction problem. There are two main approaches to take to this problem:

1. detecting 3D primitives (segments or planes) before making polyhedric building models (Jibrini & al, 2000),
2. using a parametric model-based approach (Lafarge et al, 2006).

This study aims to present a methodology for detecting building roof facets. These facets are meant to be integrated into an algorithm for building reconstruction. Many researches have been performed on this topic using DSMs as altimetric data. Nevertheless, in the last past years, airborne lidar systems (ALS) have become an alternative source for acquiring altimetric data (Baltsavias, 1999). These systems are based on the recording of the time-of-flight distance between an emitted laser pulse and its response after a reflection on the ground. They provide sets of 3D irregularly distributed points, georeferenced with an integrated GPS/INS device. The accuracy (< 0,15 m in altimetry) and the robustness of such systems are better than photogrammetric derived DSMs. However, ALS do not provide textural information that can be exploited, as they are with optical aerial images.

We therefore propose in this paper to use jointly calibrated aerial images and 3D lidar data to extract 3D roof facets. We built a joint image segmentation paradigm that includes radiometric, geometric and semantic properties of each data set. Very few researches have been performed on the fusion of lidar and aerial images and are mainly focused on image classification (Rottensteiner et al, 2004), (Haala & Walter, 1999).

We will present in the first part the theoretical background of our methodology, especially the hierarchical segmentation framework. We will then show some results of 3D roof facet extraction.

2. Methodology

2.1 Background

A region is defined as a set of pixels sharing the same properties. Segmenting an image *I* in *n* regions consists in determining a partition $\Delta_n I$ of *I* satisfying:

$$\Delta_n I = \bigcup_{i\in[1,n]} R_i, \quad R_i \bigcap_{i\neq j} R_j = \phi, \forall i, R_i \; is\, connected$$

The segmentation problem may be considered under various points of views seeing that a unique and reliable partition does not exist. Beyond classical region growing algorithms, approaches based on a hierarchical representation of the scene retained our attention. These methodologies open the field of multi-scale descriptions of images (Guigues et al, 2003). Here, we are interested in obtaining an image partition whereon roof facets are clearly delineated and understandable as unique entities.

A hierarchy is defined as a tree structure. It is a graph where nodes are related to image regions and edges (father-child relationships) to region subset inclusions. The root of the tree corresponds to the whole image and the leaves to the initial partition (over-segmentation) of the image. An **eligible partition** onto a hierarchy (or a *cut*) is therefore a set of nodes which related leaf region sets are disjoint. Figure 1 sketches a *cut* in a hierarchy represented as a dendrogram as well as the corresponding partition.

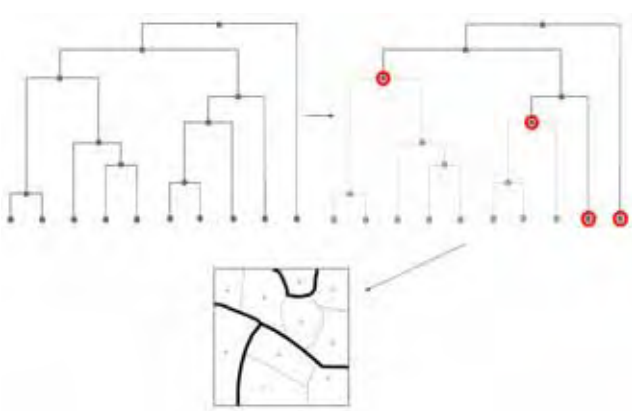


Fig. 1. Sketch of a *cut in* a hierarchy (dendrogram). Red circles are the selected cut nodes and correspond to the presented image partition.

A data structure for representing an image partition is the Region Adjacency Graph (RAG). The RAG is defined as an undirected graph $G(E,V)$ where V is the set of nodes related to an image region and E the set of edges related to adjacency relationships between two neighbouring regions. Each edge E is weighted by a **cost function** (or energy) that scores the dissimilarity between two adjacent regions. The general idea of a hierarchical ascendant segmentation is to merge sequentially the most "similar" pair of regions (or the one that

minimises the cost function) until a single region remains. The fusion of these two regions (or the contraction of the RAG minimal edge) creates a node in the hierarchy and two father-child relationships in case of a binary tree. Figure 2 sketches the generation process of the hierarchy from an initial partition of the image.

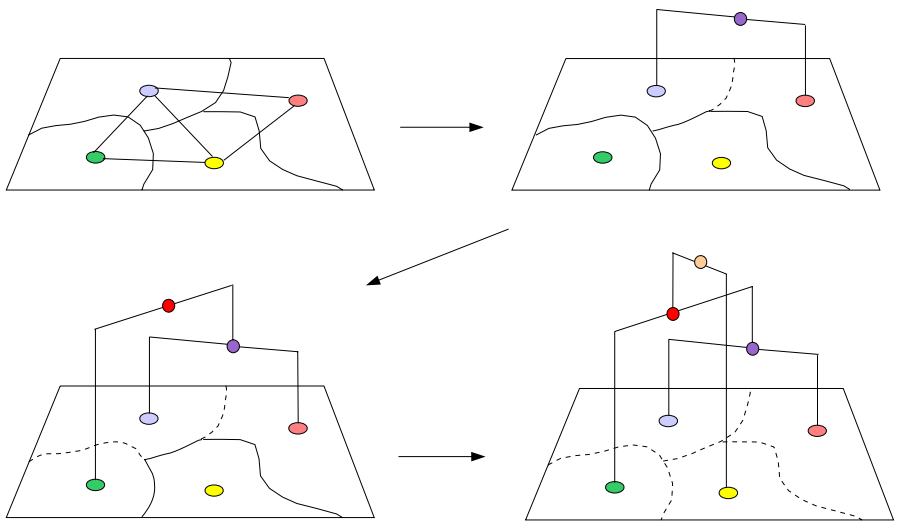


Fig. 2. Construction of a hierarchy based on a RAG (left).

2.2 Theory

The shape of the hierarchy (therefore the region merging order) constraints the existence of an eligible partition. In other words, initial regions that theoretically belong to a roof facet must be mutually merged until a node in the hierarchy appears as a roof facet entity. If it appears that sub-regions of a facet merge with adjacent regions that do no belong to their supporting facet, the embedded geometry is broken.

2.2.1 The cost function

The region merging order depends on the definition of the energy E associated to each edge of the RAG. We can write E as a sum of three terms E_r, E_l and E_s respectively related to the image radiometry, to the lidar geometry and to the semantic extracted from lidar data.

E_r is defined to minimise the loss of information when describing the image from *n* to *n-1* regions. We retained the cost function given by *Haris* (Haris et al, 1998) for merging two neighbouring regions R_i and R_j :

$$\mathrm{E}_r(R_i, R_j) = \frac{\|R_i\|_r \|R_j\|_r}{\|R_i\|_r + \|R_j\|_r} (\mu_i - \mu_j)^2$$

Where $\|.\|_r$ is the number of pixels in each region and $\mu = \frac{1}{\|R\|_r}\sum_k I(k)$ the average value of the radiometries at image sites *k* of the region.

In our context, E_l is defined to take advantage of both the **accuracy** and the **regularity** of lidar measurements onto roof surfaces to make appear in the hierarchy nodes corresponding to roof facet entities. It is therefore expected that image regions merge independently over each roof facet of the focused building. Higher levels of the hierarchy are not of interest in this study. The adequation of lidar points to lie on a roof facet is measured by estimating a plane onto those included in $R_i \bigcup R_j$. A non robust least square estimator is applied specifically for neighbouring regions not to merge when the estimated plane is corrupted by non coplanar points. Such is the case when attempting to merge two regions apart from the roof top before other couples of regions belonging to the same roof facet with possible significant radiometric dissimilarities. If $\|N_i\|_l$ (resp. $\|N_j\|_l$) is the number of lidar points in region R_i (resp. R_j) and r_p the residuals of a laser point to the estimated plane, ρ_l^2 is the average square distance of laser points to the estimated plane with

$$\rho_l^2 = \frac{1}{\|N_i\|_l + \|N_j\|_l} \sum_{p=1}^{\|N_i\|_l + \|N_j\|_l} r_p^2$$

If we consider a similar weighting factor as for E_r depending on the number of lidar points $\|.\|_l$ in image regions, E_l is expressed as:

$$\mathrm{E}_l(R_i, R_j) = \frac{\|N_i\|_l \|N_j\|_l}{\|N_i\|_l + \|N_j\|_l} \rho_l^2$$

3D lidar points have been previously processed to extract a binary semantics: **ground** and **off-ground** points. Theoretically, the off-ground class includes building and vegetation. However, we will only consider the segmentation algorithm to be focused on buildings. The process is performed with a high level of relevancy over urban areas owing to the sharp slope breaking onto building edges (Bretar et al, 2004). An image region will be classified as **ground** if it contains at least one projected lidar **ground** point. Otherwise, the region is considered to be a built up area. This binary semantics provides a reliable ground mask that can be integrated into the initial segmentation. Two regions of different classes are kept disjoint until the highest levels of the hierarchy. Finally we can write

$$\mathrm{E}_s(R_i, R_j) = \begin{cases} \infty \; if \; R_i \; or \; R_j \; is \; a \; ground \; region \\ 0 \; if \; not \end{cases}$$

2.2.2 The optimal eligible partition

A roof facet is defined as a 3D planar polygon which average square distance to lidar support points (ρ_l^2) is less than a threshold *s*. The final partition is obtained by recursively exploring the binary tree structure from its root comparing ρ_l^2 of each node to *s*.

3. Results

The test area is part of the inner city of Amiens, France. Aerial images (resolution 0,2m) are firstly re-projected into ortho-rectified geometry in order to avoid the segmentation of building facades. Using the original geometry of a set of calibrated aerial images is thought as future work. In order to enforce the region borders to lie on real discontinuities, we applied a contour detection algorithm (hysteresis thresholding) on the gradient image. The gradient was computed with a Canny-Deriche operator ($\alpha = 1$) (Deriche, 1987). The watershed algorithm is finally applied on a combination of both images (maximum of gradient and contour images). Figure 3 sketches the flowchart of the entire methodology.

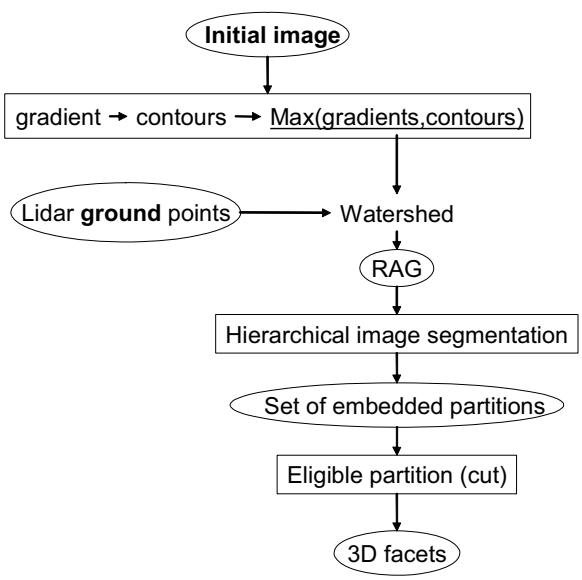


Fig. 3. Flow chart of the algorithm

We present in table 1 a set of embedded image partitions. Region contours are back-projected onto the ortho-rectified image. Parameter *s* describes a partition set. Following *s*, one can notice that structures progressively appear as unique entities until adjacent facets merge together. At the time of the study, *s* is tuned after a visual evaluation of each partition. Indeed, this threshold is highly related to the roof shape and is therefore different from one building to the other. We clearly see on these examples the delineation of the buildings with regard to ground regions as well as to courtyards. Isolated elementary regions remain within the large ground region due to the lack of lidar points inside them.

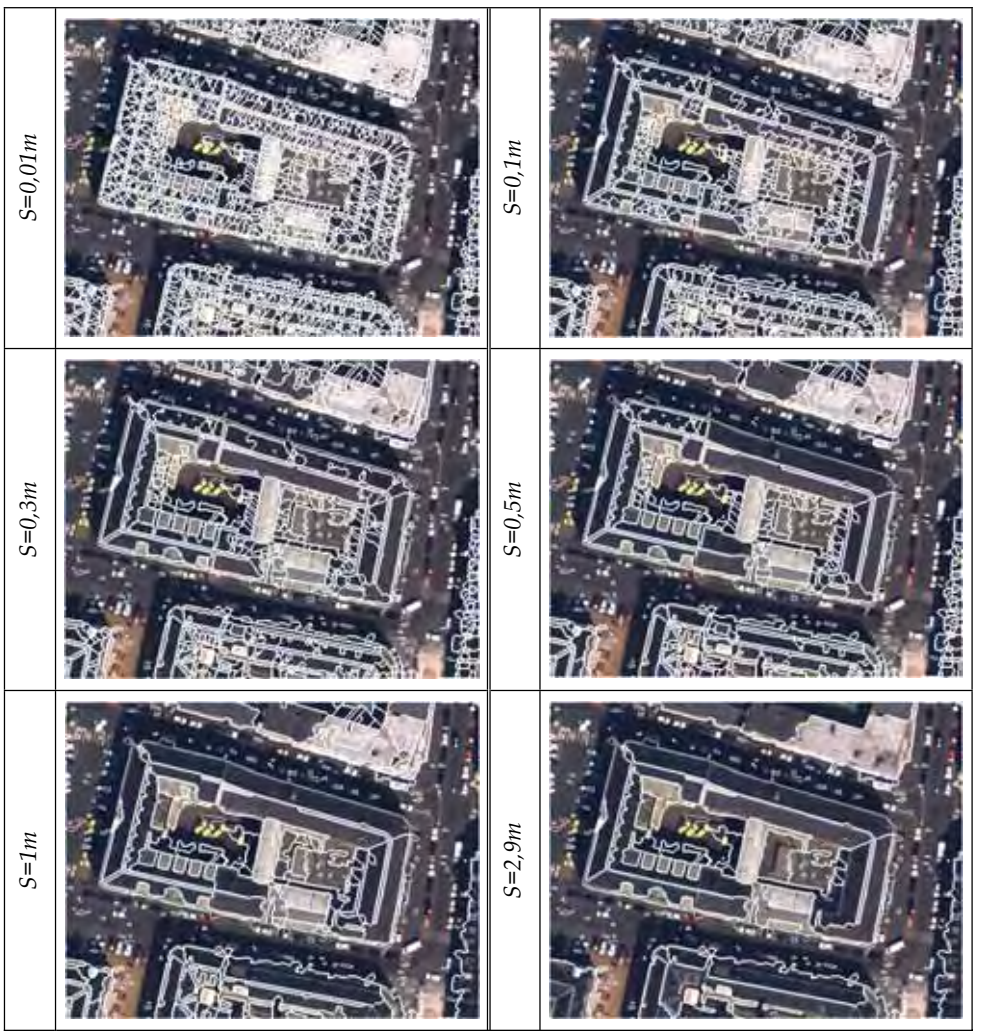


Table 1. Examples of partitions at different scales .White segments are the region borders.

As for the building presented in table 1, we consider that a final eligible partitions is achieved for *s=0,5m*. Figure 4 shows the reconstructed 3D facets of this building. This reconstruction considers lidar points belonging to an image region larger than 30 pixels and which orientation is greater than 30° from vertical. The presented 3D scenes give a realistic representation of the buildings whereon hyper-structures such as dormer windows are particularly visible. There delineation could not have been obtained considering only lidar data due to their low spatial density. The high radiometric contrast of the aerial image over some of these structures is then real complementary information. The accuracy of lidar points gives also the opportunity to detect two neighbouring regions with a low orientation difference as two different facets.

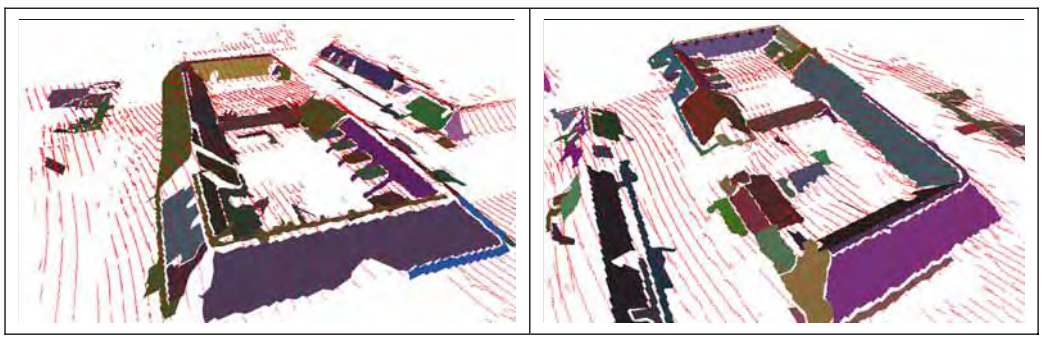


Fig. 4. 3D views of facets estimated from lidar (red) points (same building as in table 1).

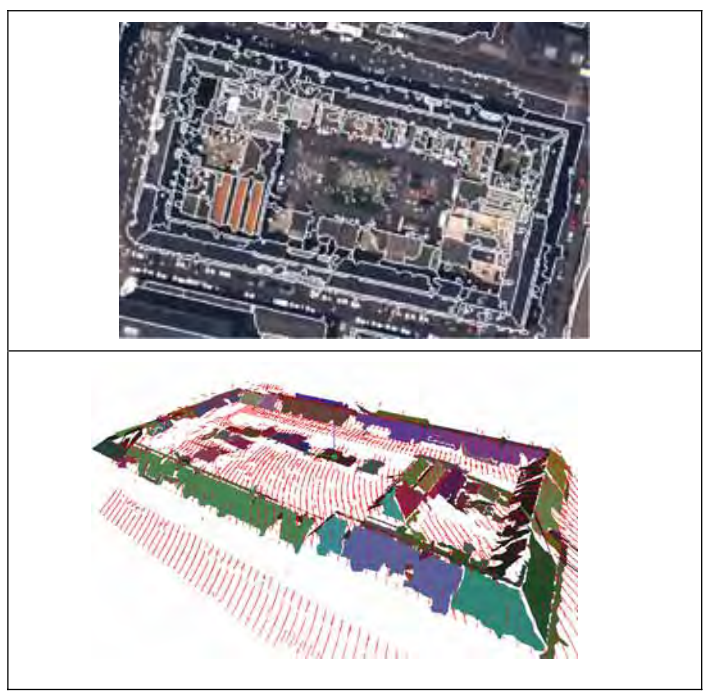


Fig. 5. Result of the algorithm on a complex building.

The 3D region contours are calculated from 2D region contours. In case of overlapping regions, the smallest one is extruded from the largest, which explains the general shape of the presented facets.

4. Conclusion

We have presented a methodology for extracting roof facets over buildings by merging aerial images and 3D lidar data in a hierarchical segmentation framework. Building roof facets are detected using radiometric, semantic and geometric information of images and of

lidar data. We have shown that integrating lidar points in an image segmentation process has enhanced the potentialities of using only 3D lidar points for extracting planar surfaces.
The 3D facet contours are not accurate or realistic even if they are based on the image contours. This is mainly due to remaining small regions located at the region borders. Seeing that images have been resampled to fit the ortho-rectified geometry, facet contours do not take benefit of the original image geometry. The future work consists at first in using aerial images in their original geometry to avoid the resampling artefacts onto building borders. In a second step, we would like to derive global criteria to provide admissible range values of parameter *s*.

5. References

Baillard, C. & Dissard, O. (2000). A stereo matching algorithm for urban digital elevation models. *Photogrammetric Engineering and Remote Sensing*, 66(9), Sept. 2000

Baltsavias, E. P (1999). Airborne laser scanning: Basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:1999-214, 1999

Deriche, R. (1987). Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, volume 1(2), pages 167–187, May 1987

Bretar, F., Chesnier, M., Pierrot-Desseilligny, M. & Roux, M. (2004). Terrain modeling and airborne laser data classification using multiple pass filtering. *Proceedings of the XXth ISPRS Symposium* , pages 314–319, Istanbul, Turkey, 2004.

Guigues, L. Lemen, H. & Cocquerez, J-P. (2003). Scale-sets image analysis. *ICIP'03*, pages 299-306, Barcelone, Spain, Oct. 2003

Jibrini, H., Pierrot-Deseilligny, M. & Maître, H. (2000). Automatic building reconstruction from very high resolution aerial stereopairs using cadastral ground plans. *Proceedings of the XIXth ISPRS Symposium*, Amsterdam , The Netherlands, 2000.

Haala, K. & Walter, V. (1999), Automatic classification of urban environment for database revision using lidar and color aerial imagery. *IASPRS,* Valladolid, Spain, 1999

Haris, K., Efstratiadis, S., Maglaveras, N. & Katsaggelos, A. K. (1998), Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12):1684–1689, Dec. 1998.

Kasser, M. & Egels, Y. (2002). *Digital photogrammetry*. Hermes – Lavoisier, Paris

Lafarge, F., Descombes, X., Zerubia J. & Deseilligny, M. P. (2006). An automatic 3D city model : a bayesian approach using satellite images. *Proc. IEEE InternationalConference on Acoustics, Speech and Signal Processing (ICASSP),* Toulouse, France, 2006

Pierrot-Deseilligny, M. & Paparoditis, N. (2005). A Multiresolution and Optimization-Based Image Matching Approach : An Application to Surface Reconstruction from Spot5-HRS Stereo Imagery. WG I/5 & I/6 Workshop on Topographic Mapping from Space (with Special Emphasis on Small Satellites), Ankara, 2005

Rottensteiner, F., Trinder, J., Clode, S., Kubik, K. & Lovell, B. C. (2004). Building detection by Dempster-Shafer fusion of Lidar data and Multisepctral aerial imagery. *ICPR*, pages 339-342, 2004

16

Natural Scene Text Understanding

Céline Mancas-Thillou, Bernard Gosselin
Faculté Polytechnique de Mons
Belgium

1. Introduction

In a society driven by visual information and with the drastic expansion of low-priced cameras, vision techniques are more and more considered and text recognition is nowadays a fast changing field, which is included in a large spectrum, named text understanding.
Previously, text recognition was dealing with documents only; those which were acquired with flatbed, sheet-fed or mounted imaging devices. Recently, handheld scanners such as pen-scanners appeared to acquire small parts of text on a fairly planar surface such as that of a business card. Issues having an impact on image processing are limited to sensor noise, skewed documents and inherent degradations to the document itself. Based on this classical acquisition method, optical character recognition (OCR) systems have been designed for many years to reach a high level of recognition with constrained documents, meaning those falling into traditional layout, with relatively clean backgrounds such as regular letters, forms, faxes, checks and so on and with a sufficient resolution (at least 300 dots per inch (dpi)). With the recent explosion of handheld imaging devices (HIDs), i.e. digital cameras, standalone or embedded in cellular phones or personal digital assistants (PDAs), research on document image analysis entered a new era where breakthroughs are required: traditional document analysis systems fail against this new and promising acquisition mode and main differences and reasons of failures will be detailed in this section. Small, light, and handy, these devices enable the removal of all constraints and all objects, such as natural scenes (NS) in different situations in streets, at home or in planes may be now acquired! Moreover, recent studies [Kim, 2005] announced a decline in scanner sales while projecting that sales of HIDs will keep increasing over the next 10 years.

1.1. Challenge of natural scene text understanding

First of all, in order to understand challenges of this field, new imaging conditions and newly considered scenes need to be detailed. The new imaging conditions deal with:

- **Raw sensor image and sensor noise**: in low-priced HIDs, pixels of a raw sensor are interpolated to produce real colours, which can induce degradations. Demosaicing techniques, viewed more as complex interpolation techniques, are sometimes required. Moreover, sensor noise of an HID is usually higher than that of a scanner.
- **Viewing angle**: scene text and HIDs are not necessarily parallel creating perspective to correct.

- **Blur**: during acquisition, some motion blur can appear or be created by a moving object. All other kinds of blur, such as wrong focus, may also degrade even more image quality.
- **Lighting**: in real images, real (uneven) lighting, shadowing, reflections onto objects, inter-reflections between objects may make colours vary drastically and decrease analysis performance.
- **Resolution and Aliasing**: from webcam to professional cameras, resolution range is large and images with low resolution must also be taken into account. Resolution may be below 50 dpi which causes commercial OCR to fail. It may lead to aliasing creating fringed artefacts in the image.

The newly considered scenes represent targets such as:

- **Outdoor/non-paper objects**: different materials cause different surface reflections leading to various degradations and creating inter-reflections between objects.
- **Scene text**: backgrounds are not necessarily clean and white, and more complex ones make text extraction from background difficult. Moreover scene text such as that seen in advertisements may include artistic fonts.
- **Non-planar objects**: text embedded in bottles or cans suffer from deformation.
- **Unknown layout**: there is no a priori information on structure of text to detect it efficiently.
- **Objects in distance**: distance between text and HIDs can vary, and character sizes may vary in a wide range, leading to a wide range of character sizes in a same scene.

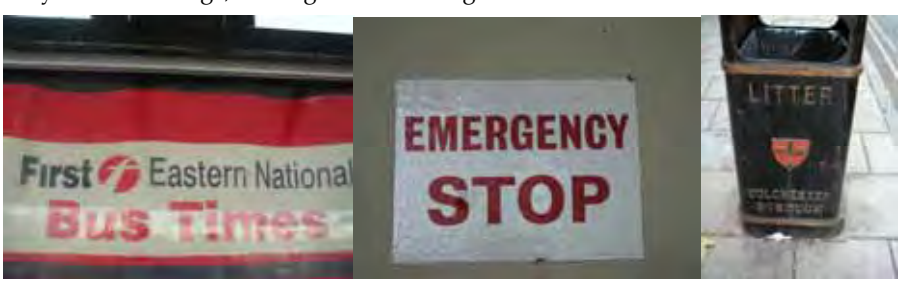


Fig. 1. Samples of natural scene images.

The main challenge is to design a system as versatile as possible to handle all variability in daily life, meaning variable targets with unknown layout, scene text, several character fonts and sizes and variability in imaging conditions with uneven lighting, shadowing and aliasing. Our proposed solutions for each text understanding step must be context independent, meaning independent of scenes, colours, lighting and all various conditions. Hence we focus on methods which work reliably across the broadest possible range of NS images, such as displayed in Figure 1.

1.2. Numerous applications

As HIDs become more and more powerful, on-the-fly image processing becomes possible, opening up a new range of applications. Nevertheless, today's HIDs are easily connected to various networks and supplementary computing resources. Starting from sign recognition for foreigners for the 2008 Olympic Games in Beijing, automatic license plate recognition to driver assisted systems with text projection on windshields, various situations could be

handled. Interesting applications such as mobile phones operating as fax machines even led to strict sanctions in Japanese bookstores!

Visually impaired people are directly affected by such research [Thillou et al, 2005]. With an HID and sufficient resources, scene in daily life may be analyzed to give them access to text and, coupled with a text-to-speech algorithm, make them "read" book covers, banknotes, labels on office doors, medicine labels and so on. For the blind community, such devices are really expected.

Another promising application is the one of visual landmark-based robot navigation. Several kinds of robot navigation may be listed such as dead-reckoning, map-based navigation, positioning sensor-based navigation or landmark-based navigation, which can be divided into natural and artificial landmarks. Natural landmarks may be designed on purpose for indoor robot navigation, such as room numbers [Mata et al., 2001], displayed in Figure 2, but may also be part of real life such as natural scenes. Even if conditions of navigation are still constrained, natural landmark-based one is very promising and satisfying results already appeared. Hence either nameplates, information signs or any text embedded in images contain large quantities of useful semantic information. Text understanding may be useful in high level robot navigation, such as path planning or goal-driven navigation. Applications are very numerous and currently only limited by imagination. Scene text is an important feature to understand for all these applications.

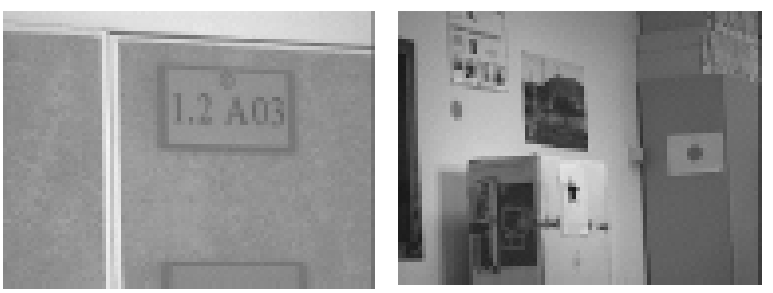


Fig. 2. Natural and artificial landmarks used in [Mata et al., 2001].

1.3. Overview of the chapter

How does one achieve the pre-cited applications? By using a text understanding system, which encompasses three main steps: text detection and localisation, text extraction from background, and text recognition.

Text detection and localisation find answers to the question: "Is there any text and where is it?". This part has been extensively studied during previous years. Text extraction from background is the field dealing mainly with uneven lighting and complex backgrounds. It is a paramount step to prepare data for OCR. Classical image segmentation such as separating sky from mountains does not need as much accuracy as text extraction, which is considered more as object-driven segmentation. Actually, text is a meaningful object which has to be extracted properly to be better recognised afterwards. Text recognition is the final step to convert character images into ASCII values to understand text and use it for particular applications.

Other NS text analysis steps such as warping, mosaicing or text tracking are also part of text understanding systems for different applications and for more details, the reader may refer to the overall state-of-the-art of Liang et al [Liang et al., 2005].
Particular focus is cast on the text extraction step: it is declared as the "most important factor for high performance" by In-Jung Kim [Kim, 2005]. Slightly studied since the inception of camera-based text analysis, text extraction suffers from imaging conditions. On the other hand, the text detection step will be only briefly mentioned in this chapter. S. Lucas, after the ICDAR (International Conference on Document Analysis and Recognition) 2005 text locating competition [ICDAR Competition, 2003], was able to conclude that "in text locating, [...] there has been a significant advance in performance [and] most easy-to-read (for humans) text is now well detected". He also mentioned that variations in illumination such as reflections cause significant problems for text understanding. Hence, considerations on uneven lighting and how to circumvent it for efficient text extraction are particularly highlighted as well.
Section 2 will describe background on text extraction and additional steps to achieve an efficient text understanding system such as character segmentation. Literature survey is also browsed along these lines. Section 3 will form the main body of the chapter with our selective metric clustering (SMC) algorithm for text extraction. The proposed solution is detailed with justifications of each step and several experiments including comparisons with other recent techniques to highlight the performance of the whole method. Section 4 will be devoted to segmentation of extracted text into individual units such as characters to improve recognition afterwards. Log-Gabor filters, well designed for NS images, are used here for the first time for character segmentation into individual components. Section 5 will describe home-made recognition used for natural scene characters with details to build an efficient training database. Finally, Section 6 will end this chapter with conclusions about text understanding for NS images and remaining issues.

2. State-of-the-Art of Natural Scene Text Understanding

Text understanding systems include three main topics: text detection, text extraction and text recognition. We assume images input into our system have previously detected text if there is any in the image. A text extraction system usually assumes that text is the major input contributor, but also has to be robust against variations in the detected text's bounding box size. For a detailed survey on text localisation methods, usually grouped into region-based, edge-based, connected components-based and texture based, the reader may refer to the survey of Jung et al. [Jung et al., 2004]. Hence, this section first details state-of-the-art methods of text extraction and then discusses character segmentation to improve text extraction and consequently, text recognition.
Text extraction is a critical and essential step as it sets up the quality of the final recognition result. It aims at segmenting text from background, meaning isolated text pixels from those of background. A very efficient text extraction method could enable the use of commercial OCR without any other modifications. Due to the recent launch of the NS text understanding field, initial works focused on text detection and localisation and the first NS text extraction algorithms were computed on clean backgrounds in the gray-scale domain. Following that, more complex backgrounds were handled using colour information. Identical binarisation methods were at first used on each colour channel of a predefined colour space without real efficiency for complex backgrounds, and then more sophisticated

approaches using 3D colour information, such as clustering, were considered. The classification of text extraction methods is displayed in Figure 3 and will be detailed further.

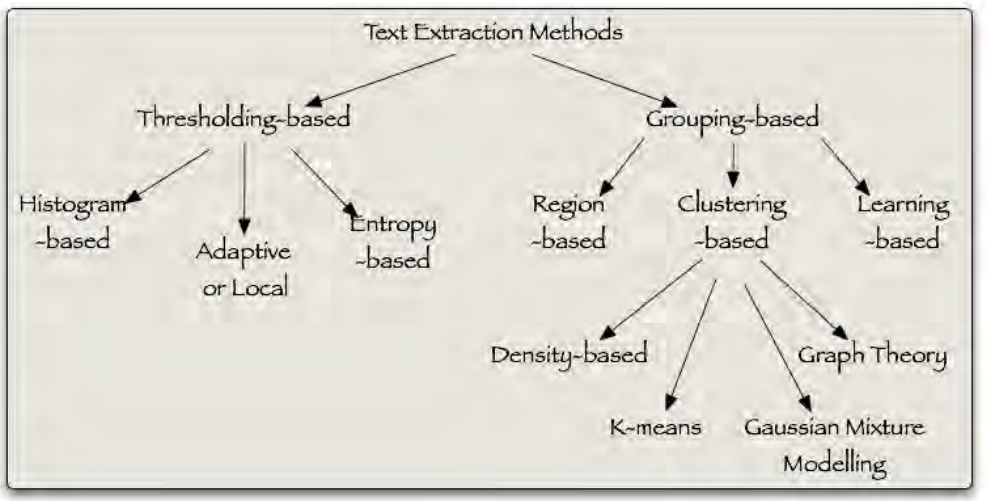


Fig. 3. Classification of text extraction methods.

- **Thresholding-based methods**

Thresholding-based methods, as the name implies, define a threshold globally (for the whole image) or locally (for some given regions) to separate text from background. **Histogram-based thresholding** is one of the most widely used techniques for monochrome image segmentation. Images are composed of several homogeneous regions with different pixel values; text is one of these regions. A histogram counts the number of each pixel value. Peaks (or modes) in histogram (meaning that several pixels have this same value) are considered as regions to segment. The threshold is chosen as the value corresponding to the valley between two peaks. The most referenced method is the one described by Otsu [Otsu, 1979], which minimises the weighted sum of within-class variances of the foreground and background pixels to get an optimum threshold as in [Thillou et al., 2005] for a visually impaired-driven application. Messelodi and Modena [Messelodi & Modena, 1992] chose two thresholds to strictly isolate the peak corresponding to text. These methods work well with low computational resources but are applied mostly on gray-scale images or colour channels independently. Moreover, they fail for images without any obvious peaks or with broad valleys which appear with complex backgrounds and slightly varying colours. **Adaptive or local binarisation techniques** define several thresholds for different image parts depending upon the local image characteristics. Several papers [Li & Doermann, 1999; Zandifar et al., 2005] for video text extraction used the Niblack's method [Niblack, 1986] where the threshold depends on local mean and standard deviation over a square window of size to define. An extension is the method of Sauvola and Pietikäinen [Sauvola & Pietikäinen, 2000] where the threshold is defined according to two parameters to define. Gllavata et al. [Gllavata et al., 2003] created their own local thresholding based on beginning and end of text lines. They assumed fairly horizontal text lines which is not necessarily the case for NS

images. Adaptive binarisations may handle more degradation (uneven lighting, varying colours) than global ones but suffer to be too parametric which is not versatile. Moreover, these techniques still consider gray-scale images only and were mainly used for video caption text or documents with clean backgrounds. **Entropy-based methods**, appropriately named, use the entropy of the gray levels distribution in a scene. Li and Doermann [Li & Doermann, 1999] minimised the cross-entropy between the input video gray-scale frame and the output binary image. The maximisation of the entropy in the thresholded image means that a maximum of information was transferred. Du et al. [Du et al., 2004] compared Otsu's binarisation and different entropy-based methods to assess that the joint relative entropy performs best on RGB channels independently for video caption text. Entropy-based techniques have been little referenced in NS context and applied only on gray-scale images or separate channels of a particular colour space.

Thresholding-based methods are lightweight enough to fit low-computational resources; that is why they are preferred for particular applications with clean backgrounds for their satisfying results on gray-scale images. Nevertheless, they are not the most suitable to handle complex backgrounds, varying colours, uneven lighting and so on.

- **Grouping-based methods**

The following methods group text pixels together according to certain criteria to extract text from background. Most popular techniques are clustering-based and are detailed further below. **Region-based approaches** include spatial-domain region growing, splitting and merging, and have been extensively used in general colour image segmentation with unknown content. These methods may be classified into two groups: top-down and bottom-up. The first one has been experienced in Kim et al. [Kim et al., 2005] by starting with the entire image and going towards smaller parts with differences between gray values exceeding a certain value. A merging process followed to refine results. In video captions, a bottom-up approach has been used by Lienhart and Wernicke [Lienhart & Wernicke, 2002]. Based on the assumption that the text contrasts well with its background, a seed around borders of text bounding box was chosen to be sure it belonged to background. With the Euclidean distance between RGB colours in a 4-neighborhood, background was extended if the distance remained below a particular value. In these two methods, a value was pre-defined and as all parametric methods, it is not versatile and cannot handle all degradations of NS images. Moreover region-based approaches are computationally quite expensive. However, they use spatial information which groups text pixels efficiently. **Learning-based approaches** have initially been designed to mimic humans by learning a training database to further recognise similar patterns. Text has interesting spatial properties and may be considered as a particular texture. Several classifiers are widely applied for pattern recognition and multi-layer perceptrons (MLP) and self-organising maps (SOM) are the most studied in text extraction. Neural networks, MLP or SOM, composed of linked neurons such as human brains, may model very general functions with any degree of non-linearity to separate pixels of text and non-text into two classes. In Hamza et al. [Hamza et al., 2005], a cascaded approach for colour historical documents with a SOM followed by an MLP was used in the training part while the trained MLP was used for testing alone. It overcame results of thresholding-based methods. Nevertheless, a training database is needed and with the wide range of NS images, this task is difficult to realise. Moreover it implies storage problems and labelling of the whole training database before being effective. **Clustering-based approaches** group colour pixels into several classes assuming that colours tend to

form clusters in the chosen colour space. They belong to unsupervised segmentation while learning-based approaches belong to supervised segmentation. Clustering-based algorithms are the most renowned and efficient methods for NS images. They are often considered as the multidimensional extension of thresholding methods. The most popular method is k-means but its generalisation, Gaussian Mixture Modelling (GMM), is more and more exploited.

1. **From density-based clustering to Mean-Shift:** Extension of histogram-based thresholding, density-based clustering is applied on colour images and needs the computation of a 3D histogram to handle colour dimensions. Adjacent colours are then merged towards the nearest highest peak. The algorithm terminates when the number of desired colours is obtained. It was used on coloured books and journal covers with relatively clean background and video scene text in Sobottka et al. [Sobottka et al., 1999]. Perroud et al. [Perroud et al., 2001] used a 4D-histogram with the RGB colour space and the channel of luminance. The Mean-Shift algorithm, first created by Fukunaga in 1975 and extended by Comaniciu [Comaniciu, 2000], seeks the "mode", point of highest density, of the 3D colour histogram. First it defines a window centered randomly at a point. The mean over the window is computed and the Mean-Shift is expressed according to the density estimate. This successful technique has not been tested on NS text, but more generally on colour segmentation.
2. **From graph theory to spectral clustering:** In graph theory concept, colour pixels are merged based on the minimum Euclidean distance (or another one) in a connected neighbourhood to form regions in the colour space. These merged pixels are represented by vertices in the graph and links between geometrically adjacent regions have weights that are proportional to the colour distance between the regions they connect. They describe a hierarchy to solve by graph theory such as in [Lopresti & Zhou, 2000; Wang et al., 2004]. It may be solved by finding a minimum of normalised cuts or more generally by spectral clustering. This latter method computes eigenvectors of the Laplacian matrix to have representation in the spectral space. The Laplacian matrix L is equal to $L = I - D^{-1/2} A D^{-1/2}$ where I is the identity matrix, D is the diagonal matrix whose diagonal elements are the sum of corresponding row of A, the affinity matrix, by then stacking the k eigenvectors in columns in a matrix which will be normalised, and fed to the k-means algorithm. k is the desired number of clusters. The main advantage of this technique is the invariance against varying colours.
3. **From k-means to GMM**: k-means is considered the most used technique in clustering. The procedure follows a simple approach to classify colour pixels in a defined colour space through a certain number of clusters (k) fixed a priori. The main idea is to define k centroids, one for each cluster and compute a defined distance between points and centroids. Iteratively, all pixels belong to a cluster whose centroid is the nearest one. Another way to deal with clustering issues is to use a model-based approach, also called probabilistic clustering. In practice, each cluster can be mathematically represented by a parametric distribution (assumed to be Gaussian). All colour pixels are therefore modelled by a finite mixture of these distributions and parameters are automatically computed with the Expectation-Maximisation (EM) algorithm or one of its variants.

There has been little experimentation done on text extraction using other clustering methods such as fuzzy c-means, which is the extension of k-means with a degree of belonging to a

cluster. As all methods can obviously not be cited in this thesis, the reader may refer to the survey of Berkhin [Berkhin, 2002].

Faced with multiple degradations and diversity of situations, text extraction alone is not sufficient to produce recognisable text for off-the-shelf OCR. Work on OCR itself may be done to improve results such as recognition of much degraded characters [Ojima et al., 2005] without any pre-processing. Nevertheless, since the main aim is to provide a solution having satisfying performance for several kinds of NS images, it is better to improve text quality beforehand, and only if necessary. Typical OCR fails against medium-quality extracted text having background portions, misalignment, too many adjoining characters such as text on a wavy tee-shirt where some characters are closer than others or totally connected. Hence to provide a very high quality extracted text, some post-processing is sometimes required and literature mainly counts rule-based methods and segmentation algorithms of characters into individual components.

- **Rule-based methods** are useful to remove spurious parts of non-textual extracted parts. Gatos et al. [Gatos et al., 2005] defined several thresholds and global variables such as the maximum and minimum number of expected characters in a text line along with the maximum and minimum number of lines in a paragraph, while Esaki et al. [Esaki et al., 2004] defined a number of rules about character sizes to remove certain parts after a global binarisation method. Text properties, such as geometry, alignment, colour, differentiating text from other objects may be used to improve text extraction algorithms. Nevertheless, strict rules with thresholds are not exploitable at all for NS images.
- Classical **character segmentation** for traditional typewritten characters fails for NS images as it assumes clean conditions and particular kinds of connectedness between characters such as the projection profile method implying vertical break lines [Luo et al., 2004]. An exhaustive survey on classical character segmentation into individual components may be found in [Casey & Lecolinet, 1996]. With the recent emergence of NS image analysis, most papers focus on text detection and localisation. When text extraction is considered, main tested images include either clean or complex backgrounds but almost without joined characters. Text on NS images such as road signs, advertisements, has to be large and easy to view with well-spaced characters. Nevertheless, more complex images may be considered with all text present in daily life such as labels on logos, brand names on clothes and so on. As previously mentioned, few papers proposed solutions. Among them, Karatzas and Antanacopoulos [Karatzas & Antanacopoulos, 2004] worked on WWW images with difficult text and suggested a region-based method to extract text followed by a fuzzy proximity measure to add topological properties of character strokes. Chen [Chen, 2003] obtained more individual components by considering text extraction with spatial information by using MRF-based text extraction. Thillou and Gosselin [Thillou & Gosselin, 2004] extracted text with a k-means clustering method and combined textual clusters by paying attention to pixels which connected individual components.

Part of our motivation is to build an efficient text understanding system with lightweight algorithms to fit within mobile devices' resources (such as PDAs) as they will be intensive future users of these systems.

3. Natural Scene Text Extraction

Text extraction is a challenging issue, made even more difficult in a NS context. Classical binarisation algorithms on gray-scale images showed their limitations to handle NS degradations. Colours have to be taken into account and we propose an algorithm that we call *Selective Metric Clustering (SMC)*. We perform a 3-means clustering algorithm using two metrics, the Euclidean distance D_{eucl} and an angle-based similarity S_{cos}, in order to mainly circumvent effects of varying colours, complex backgrounds and uneven lighting.

Several metrics, either distances or similarities, have been designed to be used in k-means in different fields requiring unsupervised classification, such as the Minkowski metric, generalisation of the traditional Euclidean distance, the Canberra distance or the normalised correlation for example. Several other measures exist and the reader is referred to [Plataniotis & Venetsanopoulos, 2000]. Angle-based similarities have been previously used for edge detection or colour segmentation by Wesolkowski [Wesolkowski, 1999] by exploiting the sine of the angle between colour vectors, for colour classification by Hild [Hild, 2004], and for vector directional filtering by Lukac et al. [Lukac et al., 2005].

To include hue information inside the RGB colour space, angle-based similarities may be considered as:

$$Hue = \begin{cases} \theta \text{ if } B < G \\ 2\pi - \theta \text{ otherwise} \end{cases} \qquad \theta = \arccos\left(\frac{1}{2} \frac{(R - G) + (R - B)}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \tag{1}$$

Hence, by keeping the same colour space and preventing computationally expensive conversions, hue information may be included with the use of angle-based similarities. Moreover, similar colours have parallel orientations even when degraded with uneven lighting or by shiny material. In natural scene images, (slight) variations are a frequent occurrence within the same object of same colour due to all sources of variations and angle-based similarity may deal with metamers to properly extract text. Finally, an angle-based similarity represents chromaticity difference information whereas the Euclidean distance computes the intensity difference information. Their combination enables one to perform intensity-dependent segmentation directly from the RGB image in areas of different colours, and the other to perform intensity-invariant segmentation in regions of similar but not identical colours.

Based on intensive tests [Mancas-Thillou, 2006], we chose an angle-based similarity S_{cos} equal to Equation 2.

$$Scos = 1 - \left(\frac{xy}{\|x\|.\|y\|} \right) \left(1 - \frac{\|x\| - \|y\|}{\max(\|x\|, \|y\|)} \right) \tag{2}$$

Additionally, intensity is paramount information to distinguish similar pixels of the same colour but different intensities and SMC includes a gray-scale image, thresholded with a traditional global binarisation to build a multi-hypothesis text extraction. Finally, as text is a meaningful object and as the chosen k-means clustering does not integrate spatial information, SMC opts for the proper text extraction by using clues of spatiality.

Figure 4 details steps of the SMC algorithm for text extraction and the following subsections detail each of these steps.

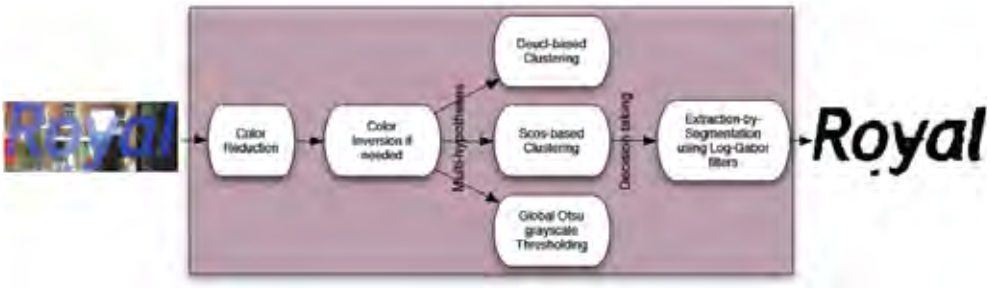


Fig. 4. Steps of the SMC algorithm.

3.1. Utilisation of a multi-hypothesis text extraction

After a colour reduction and colour inversion to always get dark text on bright background, SMC performs two clustering algorithms on the initial image with both metrics, D_{eucl} and S_{cos}. Moreover, to alleviate effects of achromatic images and improve results of text extraction, we add intensity information with the thresholded gray-scale image. For pure achromatic images (meaning R=G=B), S_{cos} cannot build 3 clusters efficiently as all pixels are on the same diagonal in the RGB cube. The same phenomenon appears for non-pure achromatic images where it is rather difficult to separate colours efficiently. This drawback is also true in hue-based colour spaces where hue is even not defined! We obtain also three possible text extraction results of both metrics and the binarised gray-scale image.

K-means clustering applied on NS colour images with two metrics forms 3 clusters for each one and one cluster is obviously a part of the background, another one is a part of the text and the third one is either text or background. For sharper results and hence better character recognition, it may be interesting to combine both textual clusters. First of all, the background colour is selected very easily and efficiently as being the colour with the biggest rate of occurrences on the image edges. Next, we propose a new text validation measure R to find the most textual foreground cluster over the two remaining clusters. Based on properties of connected components of clusters, spatial information is already added at this point to find the main textual cluster. R is based on the largest regularity of connected components of text compared to those of noise and background and is defined in Equation3.

$$R = \sum_{i}^{N} \left| \text{area}(i) - \frac{1}{N} \left(\sum_{i}^{N} \text{area}(i) \right) \right| \tag{3}$$

where N is the number of connected components and area(i) refers to the area of component i. This measure enables the computation of the variation in candidate areas. The main textual cluster is identified as the one having the smallest R. If the third unknown cluster belongs to text, both textual clusters need to be merged. A new computation of R is performed considering the merging of both clusters. If R decreases, the fusion is processed. This method enables the merging of text of different colours in the same word for instance as regularity becomes better.

With this multi-hypothesis text extraction, we may handle a very large range of NS images. The use of S_{cos} is preponderant, as illustrated in Figure 5 with some complex NS images which can not be better handled in a k-means framework. Some comparisons were done

with the Euclidean distance and by increasing the number of clusters or with other colour spaces [Mancas-Thillou, 2006]. Angle-based similarities can extract text of very challenging NS images without additional effort and by keeping versatility for other NS images.

Fig. 5. Extraction results using SMC in a RGB-based k-means framework.

3.2. Extraction-by-segmentation

After computation of k-means with two different metrics, the choice between the three text extraction methods has to be done. A multi-hypothesis method has been shown by Chen [Chen, 2003] by varying the number of clusters in a GMM-based clustering and choosing the right segmentation with the final step of recognition. One drawback to this method is to keep several segmentations to process during subsequent steps and to increase the number of text areas to recognise. Moreover, recognition is logically an efficient step to choose the right segmentation, but in complex NS images, character segmentation or even denoising steps must be added, and no decision could be done before the final step of recognition; otherwise, recognition results may be erroneously considered bad. In SMC, we choose to intermingle consecutive steps to avoid this disadvantage and to add as much information as possible.

Colour information is a very consistent clue for NS images. However the segmentation process, previously described in this section, does not make use of spatial information, which is quite necessary for object-driven segmentation and specifically text extraction. In order to extract characters properly, we exploit the same tool for character segmentation, detailed in depth in Section 4. We need to have spatial information to locate characters in the image, as well as needing the frequency information to use illumination variation to detect character edges. Hence, log-Gabor filters proposed by Field [Field, 1987] are chosen for decision making, because they particularly fit well to NS images.

One important parameter for log-Gabor filters is the filter frequency. As we used them to enhance characters in a gray-scale image, we choose a frequency equal to the inverse of the rough thickness of characters, determined by the number of pixels of the extracted result and its skeleton. A simple ratio between these two latter values is computed and the inverse is the frequency of log-Gabor filters. Results of log-Gabor filters present globally high responses to characters with this set frequency. Hence in order to efficiently choose the best extracted text result, we perform an average of pixel values. The segmentation having the highest average is chosen as the final segmentation.

3.3. SMC evaluation and results

Table 1 details results for the three hypotheses (two clustering and global binarisation) on the public database ICDAR 2003 [ICDAR Competition, 2003], which includes 2268 natural scene words. Results are expressed in terms of Precision, Recall and F-scores defined in Equation 4. F-score is the weighted harmonic average of Precision and Recall in order to more easily compare results.

$$\text{Precision} = \frac{\text{Correctly extracted characters}}{\text{Total extracted characters}}, \ \text{Recall} = \frac{\text{Correctly extracted characters}}{\text{Total number of characters}} \quad (4)$$
$$\text{F-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}}$$

Extraction	Precision	Recall	F-score
D_{eucl}	0.90	0.88	0.89
S_{cos}	0.93	0.36	0.52
Binarised gray-scale image	0.88	0.76	0.82

Table 1. Precision, Recall and F-score measures of text extraction performed by the three extraction hypotheses.

To add more arguments to complementarities between these three extracted results, D_{eucl} performs better in 5% images, while S_{cos} in 12% and the global thresholding in 9%. There is a larger overlap between D_{eucl} and the global thresholding which performs quite equally in 69% images.

To choose the right text extraction, we opt for log-Gabor filters by adding spatial information. In [Mancas-Thillou & Gosselin, 2006], we compared the performance of this method with the Silhouette technique, a measure of how well clusters are separated, to choose between the two metrics only. It can be logical to think that best text extraction results present the best separation between clusters. However, it is not always true because Silhouette performs well in 77.7% images and our proposed method using spatial information performs well in 93.2%, yielding an improvement of 19.9%.

A few works deal with NS text extraction and we compare SMC, firstly, with solutions of Wolf et al. [Wolf et al., 2002] which designed an extended method of Sauvola and Pietikäinen [Sauvola & Pietikäinen, 2000] to extract text from NS images or videos, and then, with solutions of Garcia and Apostolidis [Garcia & Apostolidis, 2000] which used a k-means clustering in the HSV space with the Euclidean distance only. Combination of clusters in this last method has not been implemented and a perfect combination is assumed while our method is tested including our combination method. Results are presented in terms of Precision, Recall and F-score in Table 2.

Methods	Precision	Recall	F-score
Wolf et al.	0.35	0.19	0.25
Garcia and Apostolidis	0.66	0.57	0.61
SMC	0.95	0.91	0.93

Table 2. Comparison of Precision, Recall and F-score measures between Wolf's method, Garcia and Apostolidis's method and our SMC method.

The combination of two metrics in a clustering framework and a global thresholding has proven its efficiency compared to two recent and competing algorithms. Finally, due to the explosion of use of camera phones or digital cameras and huge amount of images to process for text extraction, the algorithm needs to be relatively fast in order to provide satisfying results for frequent use. Our text extraction algorithm runs in 0.61 seconds on average for our databases on a PC with a Pentium M-1.7 GHz micro-processor. The source code for text extraction was developed in C language but could be optimised further.

4. Unit-based Segmentation

This section deals with segmentation of text areas into specific units, such as lines, words and characters. In commercial OCR systems, this process is usually included and is quite successful except for severely degraded characters, strongly broken or tightly connected ones where recognition rates drastically drop. Incorrect segmentations due to perspective, for example, may even lead to no recognition at all. Usually, NS text, handled in literature, is well separated due to their reading goal. However, complex NS images with low resolution, perspective or wavy surfaces present challenges and unit-based segmentation has recently become a point-of-interest to circumvent recognition errors. Hence, we describe a fast and simple line and word segmentation method and an innovative and robust character segmentation method using log-Gabor filters.

4.1. Line segmentation

NS images may present several words but usually only a few lines if we cite street names or book titles. Nevertheless, colourful magazine headlines or abstracts on book covers or even camera-based documents such as restaurant menus may have several lines. Line segmentation are usually not considered as difficult for NS images but present interesting challenges for skewed text areas; as such we present very fast and intuitive algorithms.

Segmentation into lines is an old topic and the two main and successful methods are either the vertical projection profile or the Hough transform. The first one is a histogram of the number of text pixels accumulated along text lines and projected vertically. The projection profile has maximum-height peaks for text and valleys for inter-line spacing. It is quite sensitive to noise and skewed lines. The second method maps each point in the original (x,y) plane to all points in the (r, θ) Hough plane of possible lines through (x,y) plane with slope θ and distance from origin r. This method performs well on skewed text and may also simultaneously deskew it with the knowledge of θ value but it is on the other side computationally quite expensive.

Connected components coming from our text extraction step to perform the deviation measure R are already computed with general properties, such as height of characters h_{char}. On the bounding box of the text area, we define the approximate number of lines N_l by:

$$N_l = \text{floor}(1 + \frac{h_{text} - \mu(h_{char})/2}{\mu(h_{char}) * 3/2}) \quad (5)$$

where h_{text} is the height of the text area, $\mu(h_{char})$ is the average of h_{char} on all characters and floor(x) is the largest integral value less or equal to x. All y-coordinates of character centroids are then clustered with the k-means algorithm, k being equal to N_l segmentation.
For strongly skewed lines, a fast deskewing is required based on the height of the text bounding box. The first text pixel of the first row of the tightest bounding box is detected and if its position is before the middle of the image width, the skew angle is negative; otherwise it is positive. A first rotation of 1° is computed in the determined direction. If the bounding box is shorter in height than the previous one, successive rotations are performed until the bounding box becomes higher meaning that the skew angle was larger than 1°.

4.2. Word segmentation

Word segmentation, contrarily to line segmentation useful for better character recognition, is a crucial step for text understanding after recognition, such as by speech synthesis. A natural linguistic parser is always part of a text-to-speech algorithm and it is important to identify words for a proper pronunciation as explained in the example:
Ex: in French, the phonetic transcription can be different, depending on word segmentations:

« les tas » → [l e t a] and « lestas » → [l ε s t a]

In Latin alphabets, the inter-words distance D_{IW} is larger than the one of inter-characters D_{IC}. We compute word segmentation by identifying word separations by all distances superior to std(D_{IC}) + mean(D_{IC}) with std(.) and mean(.), respectively standard deviation and mean of inter-character distances in a given line. This step occurs after the refined character segmentation in order to have more correct calculations based on characters and spaces between characters.
For this step, we use a simple statistic method. Some errors may occur when a few words are present with distances between words varying due to different fonts or perspective. Nevertheless, this algorithm is robust when run against text areas presenting only one word, which is quite frequent in NS images or after text detection algorithms, which usually oversegment lines. Finally, this rule basically bends to oversegmentation more than subsegmentation, which may be more easily handled by our recognition and correction we proposed in [Mancas-Thillou, 2006].

4.3. Character segmentation using log-Gabor Filters

The first character segmentation algorithms, developed for typewritten characters, appeared more than forty years ago to separate each character individually, in order to subsequently feed into OCR. Later, these techniques have been extended to segmentation of cursive writing for handwritten text. Main techniques for typewritten characters are categorised into three groups. *Image-based methods* are mainly issued from projection analysis or the

"Caliper" distance, which is the distance between the uppermost and bottommost pixels in each column meaning that smallest distances are tentative segmentation places, as experienced in camera-based document processing [Thillou et al, 2005]. These methods imply vertical separation only, which is not convenient at all for strongly joined characters or skewed and italic ones where parts of a character infringe on the space occupied by the next one. *Recognition-based methods* use a sliding window of variable width to provide sequences of hypothetical segmentation locations which are confirmed or refuted by character recognition. These techniques also give only vertical separations and need robust OCR to reject or accept all possible segmentations, which are quite numerous, even for a single word! *Hybrid methods* mainly encompass oversegmentation methods. A word is dissected into its smallest possible components and recognition is based on these units to individually recompose the characters one at a time. They are particularly well suited for joined and broken characters and segmentation results are not only vertical as based on small components. Nevertheless, oversegmentation techniques need a dedicated recogniser based on unit features.

NS images need robust character segmentation since not all aforementioned methods are suitable, and off-the-shelf OCR using them lead to too many recognition errors. A gap between complex NS images and character recognition has to be filled to extend applications and use of NS images in daily life. A NS character segmenter is needed to increase NS character recognition and has to be robust against already individual characters, broken and joined ones and against unknown fonts, italic characters or with perspective. A very innovative solution, using log-Gabor filters and the recognition step that follows in a hybrid method, is fundamentally different from existing ones, and is presented after focusing on properties of these filters

- **Why are log-Gabor filters appropriate for NS character segmentation?**

Character segmentation in NS images obviously needs text properties and gray-level information to complement the colour information exploited in text extraction. Hence simultaneous spatial and directional information (for character separation location) and frequency information (gray-level variation to detect cuts) are required. Gabor filters are a traditional choice to address this issue: they are cosine-like filters having a given direction and modulated by a Gaussian window. They have been extensively used to characterise texture, and more specifically in our context, to detect and localise text into an image. In this aim, Gabor filters are quite time consuming because several directions and frequencies must be used to handle the variability in character sizes and orientations. Moreover, Gabor filters present limitations: large bandwidth filters induce a significant continuous component and only a maximum bandwidth of 1 octave could be designed. Field [Field, 1987] proposed an alternative function called log-Gabor which lets us choose a larger bandwidth without producing a continuous component. Moreover, he suggested that natural images are better coded by filters that have a Gaussian transfer function on a logarithmic frequency scale, by showing that their spectrum statistically falls off at approximately 1/f, which corresponds well to where the log-Gabor filter spectrum falls off on a linear scale. Figure 6 displays the shape of log-Gabor functions at the same frequency but with bandwidth varying from 2 to 8 octaves. Log-Gabor functions have the same appearance as Gabor functions for bandwidths less than one octave. The possibility of sharpening the filters is highlighted.

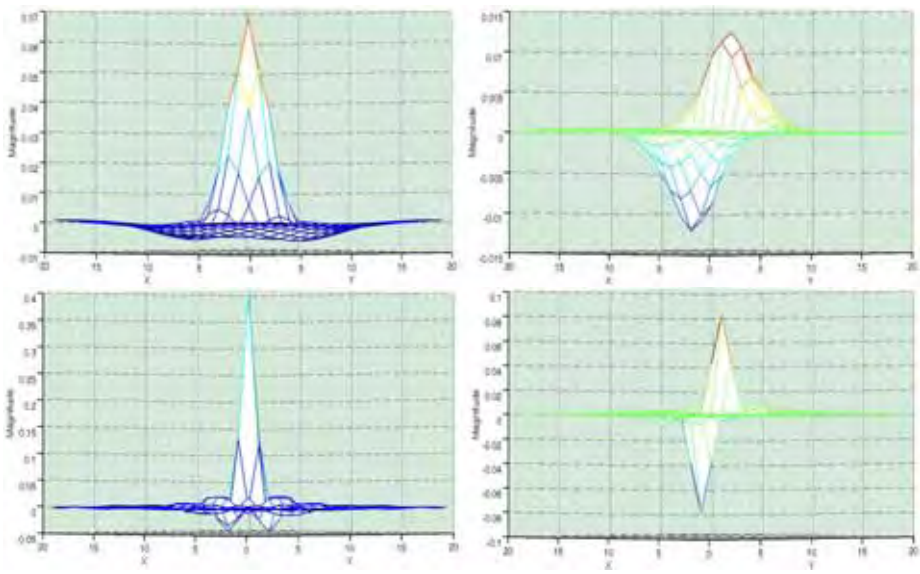


Fig. 6. From top to bottom: even (left) and odd (right) log-Gabor filters with a bandwidth of 2 octaves and even (left) and odd (right) log-Gabor filters with a bandwidth of 8 octaves. In the spatial domain, the possibility of sharpening the filters is highlighted.

Log-Gabor filters in the frequency domain can be defined in polar coordinates by $H(f, \theta) = H_f * H_\theta$ where H_f is the radial component and H_θ, the angular one:

$$H(f,\theta) = \exp\left\{\frac{-[\ln(f/f_0)]^2}{2[\ln(\sigma_f/f_0)]^2}\right\} * \exp\left\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\right\} \tag{6}$$

where f_0 is the central frequency, θ_0 is the filter direction, σ_f is the standard deviation of the radial components of the Gaussian describing the filter and is used to define the radial bandwidth and σ_θ is the standard deviation of the angular part of the Gaussian and enables the definition of the angular bandwidth. As we are looking for vertical separation between characters, we only use two directions for the filter: the horizontal and the vertical ones. Hence, for each directional filter, we have a fixed angular bandwidth of $\pi/2$, which determines σ_θ. Log-Gabor filters are not really strict with directions and defining only two directions enables the handling of italic and/or misaligned characters. For highly misaligned characters, the number of directions could be increased to handle this additional degradation, but it is important to mention that the angular bandwidth will become narrower and hence more selective.

Only two parameters remain to be defined, f_0 and σ_f, which are used to compute the radial bandwidth. The central frequency f_0 is used to handle gray level variations to detect separation between characters. The spatial extent of characters is their thickness that we consider as the wavelength of "characters", hence it is logical to get a central frequency close to the inverse of the thickness of characters to get those variations. However, the measurement of character thickness may not be very accurate depending on the presence of

degradations. In order to handle all kinds of degradations, we compensate for inaccurate thickness estimation with the second parameter σ_f. If the thickness of characters is not consistent inside a character, some character parts can be removed permanently. In this case, by increasing the bandwidth, we can support the variability in the thickness of characters with a "larger" filter. Moreover, sometimes with very degraded or close characters, the thickness is very difficult to estimate and the filter must be very sharp to get each small variation in the gray level values such as in Figure 7, with a complex NS image.

Fig. 7. Impact of varying log-Gabor bandwidth for character segmentation. Original image (top left), binary version (top right), segmentation with large bandwidth (bottom left), segmentation with narrow bandwidth (bottom right).

As degradations and conditions of frequency estimation are quite unexpected, we chose the bandwidth filter in a dynamic way using recognition results. In the following part, we detail our method and how each parameter is estimated.

- **Character segmentation-by-recognition**

Based on the binarisation of the detected area, which is available with the proposed SMC algorithm, the character segmentation may now be performed on gray-level images. To define frequency, a classical way is to use a "wavelet-like" method. This means trying out several frequencies to get a good result for one of them. This method is time consuming due to several convolutions with multiple frequency filters and the number of computations rose to the power of two with the second parameter. Text embedded in natural scene images presents a quite consistent wavelength, which is very different from the background. For our filter, we decided to use a wavelength related to the average of the character thicknesses. This is computed by using the ratio between the number of pixels of the first mask obtained by the SMC method and its skeleton.

Due to the large variation in NS character fonts and sizes, the bandwidth has to be chosen dynamically. As objects to be segmented are text, we can use segmentation-by-recognition to choose the convenient bandwidth. We fix the initial and final values for the bandwidth estimation. From approximately 2 octaves to approximately 8 octaves, which makes σ_f/f_0 vary with a step of 0.1 (from 0.1 to 0.6), we process six filters and provide the result to an OCR engine.

The result is composed of the vertical filter only as the character separation is mainly vertical. Moreover, in the output, only the phase of the filter will be exploited. As the text and background information have different wavelengths, the phase contains much more information than magnitude, as displayed in Figure 8. Moreover, local variation issued from

the initial separation between characters induces a phase difference. The latter one contains the gray-level information while the phase shows a local map which makes a good separation between the background and the textual information; this intermediate result is then multiplied by the first mask from text extraction to remove possible noise around characters as displayed in Figure 9.

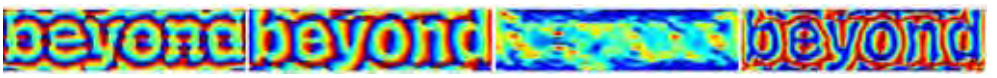


Fig. 8. Log-Gabor filtering results for each filter property. From left to right: phase of the horizontal filter, phase of the vertical filter, magnitude of the vertical filter and absolute phase of the vertical filter.

As shown in Figure 9 after filter convolution, characters have mainly low intensities and higher background intensities. In order to remove spurious parts between characters and to remain parameter-free, we use a global Otsu thresholding [Otsu, 1979], which automatically chooses the threshold to minimise the intra-class variance of the thresholded black and white pixels. With the use of the absolute phase of the vertical filter, only one threshold needs to be determined. After this step, we get a result, such as the one shown at the bottom of Figure 9, to choose the bandwidth for filters.

Fig. 9. Phase of the vertical filter multiplied by the mask issued from the text extraction (left) and result after global thresholding (middle). Improvement is obvious from the binary version (right).

We use a home-made OCR algorithm composed of a multi-layer perceptron with geometrical features to recognise characters, which is trained by a separate data set and is used to assess how well characters are segmented. Detailed explanations about this in-house OCR are provided in Section 5. After applying log-Gabor filters, connected components (mostly characters) are given as inputs to OCR. Recognition rates for each character or assumed character are averaged and the maximum score enables the choice of the bandwidth. This estimation needs six straightforward filters with only one frequency which enables the use of log-Gabor filters for character segmentation in a low-resource context.

Some examples are given in Figure 10 to appreciate performance of this proposed character segmentation based on log-Gabor filters. From top, the third example is composed of severely joined characters and the result after segmentation is very satisfying. Between 'i' and 'n' of the word 'smokin', the connection is still present but the recognition is now successful even with off-the-shelf OCR including traditional segmentation. The last example illustrates an original image with characters of two different major colours (yellow and white) and a yellow and blue background. Based on our combination of clusters, the 'M' of the word 'Memorex' has been reconstituted but simultaneously with some parts of background. Nevertheless, the yellow background information has a different intensity and frequency than the 'x' character, leading to a successful segmentation.

Even if in NS images, broken characters are rare due to the relatively large thickness of characters whose aim is to be read, it may be useful to have solutions for handling them. To recompose parts of a single character, we proposed in [Mancas-Thillou et al., 2005] an

algorithm using log-Gabor filters as well. It enables the correction of already broken characters (particular fonts or text extraction errors) and new broken characters due to recognition failures. The bandwidth is fixed and the frequency estimation is refined by an iterative log-Gabor convolution.

Fig. 10. Some character segmentation examples. From left to right: original image, SMC-based binary version and result after character segmentation.

The convolution of text extraction results with log-Gabor filters has several goals: to choose the better extracted text, to segment characters into individual parts and also to fuse broken characters by validating or not previous outputs. Log-Gabor filters give a large set of applications in NS images with a large modularity and very satisfying results as detailed in the following subsection.

4.4. Character segmentation evaluation

In Table 3, comparisons are done between the behavior of an efficient commercial OCR (ABBYY FineReader 8.0 Professional Edition Try&Buy) against initial images without any processing, after the SMC-based text extraction without character segmentation, after a classical "Caliper" distance-based segmentation and after the log-Gabor-based segmentation -by-recognition to show the efficiency and necessity of this latter method to improve recognition results. Error rates are computed using the Levenshtein distance between the ground truth and the resulting text. The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. Equal weights for each operation are employed in our computation. Error rates are then computed by dividing with the number of characters. By using the Levenshtein distance, some error rates for a word may be superior to 1, but it is useful to penalise broken characters. Tests have been computed on 10% of the database due to the impossible automatic processing with a commercial OCR. To compute log-Gabor filtering, we use the Kovesi' toolbox [Kovesi, 2006] in Matlab. The home-made OCR, which is useful to choose the right bandwidth, has been extended in C language from a version of Gosselin [Gosselin, 1996]. The "Caliper" distance and evaluation measures have been developed in Matlab.

Error rates	Colour images	SMC-based images	"SMC-based+ Caliper" images	"SMC-based +Log-Gabor" images
ICDAR2003 database	71%	40%	43%	19%

Table 3. Usefulness of character segmentation in natural scene images stated from recognition error rates with a commercial OCR.

For the ICDAR2003 database, "Caliper"-based segmentation even gives worse results than without segmentation. It is mainly due to the number of broken characters which increases. Log-Gabor segmentation drastically decreases error rates.
In this proposed character segmentation, the bandwidth is estimated with the recognition step and we compute the efficiency rate of this decision. Some erroneous choices could be made due to our majority vote on the whole text and the decision is correctly taken in 98.1% of images. Errors are mostly avoided with this character segmentation-by-recognition as each decision is checked with other steps dynamically. Main errors are either due to the OCR engine with much degraded characters or to the presence of thin characters. As log-Gabor filters exploit intensity information to accurately segment characters into individual components, if characters are too thin, they will be easy to break in several pieces of characters, leading to erroneous recognition.
Some deeper comparisons [Mancas-Thillou, 2006] have been done with a recent method from Gatos et al. [Gatos et al., 2005], who used the same public database. Their text extraction is based on a gray-scale adaptive thresholding and they proposed to recombine characters components based on several rules to avoid too many joined characters. We use the same evaluation method being the Levenshtein distance. Improvement may be observed with an error rate decreasing of around 43%.

5. Natural Scene Character Recognition

From text extraction to unit-based character segmentation, the main goal was to improve extracted text in order to finally increase recognition rates. Hence, in this section, the objective is to provide high-quality extracted text in order to exploit off-the-shelf OCR. Nevertheless, NS character recognition, faced with the very large diversity of images without any a priori information, needs suitable conditions to work properly, such as a huge and representative training database.

5.1. Considerations on character recognition

To focus on NS character recognition, main recent papers deal with gray-level characters to handle degradations and low resolution of acquisition. The idea is therefore to build efficient recognisers against some issues without improving characters beforehand. For WWW images, Zhou et al. [Zhou et al., 1997], first extracted characters by colour clustering and then converted the characters' colours into gray-scale. The main colour receives the value of 255 and the other ones are set to differences from the representative colour. The character shape is then treated as a 3D surface and a polynomial surface fitting method (Legendre polynomial basis) is used as feature extractor and a basic character-to-class Euclidean distance is used to recognise characters. For NS text, Zhang et al. [Zhang et al., 2002] exploited also gray-scale images after intensity normalisation with Gabor-based

features in the context of Chinese sign recognition. They performed feature selection with a linear discriminate analysis to build a space as discriminate as possible. Finally the classification is solved with kNN.

To perform segmentation-by-recognition in Section 4, we use an extended version of classifier from Gosselin [Gosselin, 1996], based on geometrical features and a multi-layer perceptron (MLP). In order to recognise many variations of the same character, features need to be robust against noise, distortions, style variation, translation, rotation or shear. Invariants are features which have approximately the same value for samples of the same character, deformed or not. To be as invariant as possible, our input-characters are normalised into an N*N size with N=16. However, not all variations among characters such as noise or degradations can be modelled by invariants, and the database used to train the neural network must have different variations of a same character.

In our experiments, we use a feature extraction based on contour profiles. The feature vector is based on the edges of characters and a probe is sent in each direction (horizontal, vertical and diagonal) and to get the information of holes like in the 'B' character, some interior probes are sent from the center. Moreover, another feature is added: the ratio between original height and original width in order to very easily discriminate an 'i' from an 'm'.

Experimentally, in order to lead to high recognition rates, we complete this feature set with Tchebychev moments, which are orthogonal moments. Moment functions of a 2D image are used as descriptors of shape. They are invariant with respect to scale, translation and rotation. According to [Mukundan et al., 2001], we use Tchebychev moments of order 2 for their robustness to noise.

No feature selection is defined and the feature set is a vector of 63 values provided to an MLP with one hidden layer of 120 neurons and an output layer of size 36 for each Latin letter and digit. Due to few training samples for capital letters, uppercase and lowercase letters were initially grouped into the same class. Nevertheless, with the algorithm of increasing database described in the next paragraph, an output layer of 62 neurons may be considered efficiently. The total number of training samples is 40614 divided into 80% for training only and 20% for cross-validation purpose in order to avoid overtraining.

5.2. Zoom on training database: how to build a relevant and general one?

Traditional database increasers are based on geometrical deformations such as affine transformations or on the reproduction of a degradation model such as [Sun et al., 2004] to mimic low resolution. In NS images, the very large diversity must be handled and character extraction of a huge data set is awkward and difficult to achieve. Hence, we increase the NS database with the image analogies of Hertzmann et al. [Hertzmann et al., 2001], with the particular algorithm of texture-by-numbers. Given a pair of images A and A', with A' being the binarised version of A, the textured image in our algorithm, and B' the black and white image to transfer texture, the texture-by-numbers algorithm applies texture of A into B' to create B. Binary versions are composed of pixels having values of 0 or 1; texture of A corresponding to areas of 0 of A' will be transferred to areas of 0 of B' and similarly for 1. Multiscale representations through Gaussian pyramids are computed for A, A' and B' and at each level, statistics for every pixel in the target pair (B, B') are compared to every pixel in the source pair (A, A') and the best match is found.

One sample used to increase the training database is displayed in Figure 11, which also schematises the concept of image analogies.

Fig. 11. Principe of image analogies in the context of database increase: A represents the textured and segmented character, A' its binary version. From a binary version of an 'm' in B', the texture is transferred onto B, similar to the analogy between A and A'.

The entire process of increasing database is firstly based on character extraction from a given data set, using SMC algorithm of Section 3. Characters are hence binarised and normalised. Deformations on character thickness, slant, rotation, and perspective are then performed and the texture-by-numbers is applied on each binary image. A huge and new data set is hence built. To provide standardised characters, all newly-textured characters are then binarised always using our SMC algorithm, leading to realistic degradations of NS images, which enables to increase the database as naturally as possible. Based on the finite steps of variation for each of the pre-cited parameters, for one extracted character and one given texture, 33480 samples may be created. Hence, the power of increasing database of this method is very large (almost infinite depending on the parameter variation and the number of textures). Some tests have been done on recognition and rates are slightly increased. Extensive studies are needed to know if the increase is due to the enlarging database and/or the representativeness of the database with texture transfer. Nevertheless, this technique enables the growing of a database in a fast and reliable way.
Finally, character recognition alone is hardly error-free and linguistic information needs to be added to correct errors for which we build a light and modular solution. For this purpose, we intermingle steps of recognition and correction in order not to consider OCR as a "black box".

6. Conclusion and Future Works

This last section aims at concluding this chapter by summing up main steps in the first part to highlight important points according to us to realize an efficient and versatile NS text understanding and the second parts emphasizes interesting work prolongations in other image processing fields and the focus to give in next years.
Our SMC algorithm has been proposed based on a multi-hypothesis text extraction by selecting either the right clustering metric or the dual information between colour and illumination, using log-Gabor filters. Several points have been detailed such as the superiority of metrics over colour spaces in a clustering framework inside a general NS context. Angle-based similarities have overcome any other colour spaces to handle complex NS images, meaning mainly images with complex backgrounds and uneven lighting. Moreover, complementarities between the Euclidean distance and angle-based similarities in a k-means method to handle a very large set of NS images have also been described. Spatial and luminance information have been added to choose the best text extraction to provide to recognition. To circumvent NS challenges, text extraction was

intermingled with the subsequent step of character segmentation and very encouraging results have been shown in terms of Precision, Recall and F-score, comparison with other state-of-the-art algorithms, and while keeping a reasonable computation time.

Our selective metric-based clustering is aimed at being versatile and results we have provided show that it is. Nevertheless, SMC mainly uses colour information and one drawback of our system is for natural scene images having embossed characters. In this case, the foreground and background have the same colour imparting partial shadows around characters due to the relief but not enough to discriminately separate the textual foreground from the background as displayed in Figure 12. Gray-level information with the simultaneous use of a priori information on shadows and character properties could be a solution to handle these cases. Nevertheless, it may be relevant to note that a robust OCR may also give satisfying results without any modifications of our algorithm.

Fig. 12. Error example of our selective metric-based clustering: initial colour embossed image on left and the SMC result on right.

In a second step, we propose NS character segmentation-by-recognition based on log-Gabor filters whose some parameters are defined dynamically. This algorithm fulfils initial requirements and gives interesting results under various aspects:

- No assumption on characters fonts, sizes or skew is done
- Characters are segmented with not only vertical separations but cuts following the character profile, leading to increased recognition rates
- Touching and broken characters are handled
- The algorithm is made more robust by using additional information with the consecutive step of character recognition
- Satisfying results in terms of recognition rates and Levenshtein distance.

To conclude, log-Gabor filters are very modular and efficient tools to segment NS characters into individual and understandable components.

Among future works of each step detailed in the previous paragraphs, one of the main prolongation works will be to extend some of these solutions for extraction of other objects in natural scene images to show once again versatility of these methods. Obviously, character segmentation is a dedicated step of text analysis. Nevertheless, our combination of colour, intensity and spatial information or handling of low resolution frames may lead to interesting results for other applications.

About the global system and if resources are available, the small amount of errors at each step may be decreased by keeping information until recognition. These additional hypotheses will be handled through another step of information fusion.
Due to the great expansion of electronic goods and their ever increasing performance, readers may wonder if these chapter topics will not be obsolete in a few years. In some recently launched smartphones in Asia with 3.2 Megapixels cameras and rudimentary embedded OCR or with expansion to 8 Megapixels of consumer-grade digital cameras, text extraction part handling complex backgrounds and uneven lighting will be necessary for a long time: professional expensive cameras have still problems with illumination by nature and complex backgrounds, especially in advertisements. Such issues will not disappear anytime! Unit-based segmentation may be removed by other computationally very demanding methods but character recognition is mandatory to understand text. Hopefully, text understanding steps will be automatically embedded into handheld imaging devices soon for exciting and useful applications in daily life!

7. References

Berkhin, P. (2002). Survey of clustering data mining techniques, *Tech. report*, Accrue Software

Casey, R.G. & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, pp. 690-706

Chen, D. (2003). *Text detection and recognition in images and video sequences*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne

Comaniciu, D. (2000). *Nonparametric robust methods for computer vision,* PhD thesis, Rutgers University

Du, Y.; Chang, C-I. & Thouin, P.D. (2004). Unsupervised approach to colour video thresholding, *Optical Engineering*, Vol. 43, No. 2, pp. 282-289

Esaki, N.; Bulacu, M. & Shomaker, L. (2004). Text detection from natural scene images: towards a system for visually impaired persons, *Proceedings of Int. Conf. Pattern Recognition*, pp. 683-686

Field, D.J. (1987). Relations between the statistics of natural images and the response properties of cortical cells, *Jour. Opt. Soc. Amer. A*, Vol. 4, No. 12, pp. 2379-2394

Garcia, C. & Apostolidis, X. (2000). Text detection and segmentation in complex colour images, *Proceedings of Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 2326-2330

Gatos, B.; Pratikakis, I. & Perantonis, S.J. (2005). Towards text recognition in natural scene images, *Proceedings of Int. Conf. Automation and Technology*, pp. 354-359

Gllavata, J.; Ewerth, R. & Freisleben B. (2003). Finding text in images via local thresholding, *Proceedings of IEEE Symposium on Signal Processing and Information Technology*, pp. 539-542

Gosselin, B. (1996). Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits, PhD thesis, Faculté Polytechnique de Mons

Hamza, H.; Smigiel, E. & Belaid, A. (2005). Neural based binarisation techniques, *Proceedings of Int. Conf Document Analysis and Recognition*, pp. 317-321

Hertzmann, A.; Jacobs, C.E.; Oliver, N.; Curless, B. & Salesin, D.H. (2001). Image analogies, *Proceedings of ACM SIGGRAPH, Int. Conf. On Computer Graphics and Interactive Techniques*

Hild, M. (2004). Colour similarity measures for efficient colour classification, *Jour. of Imaging Science and Technology*, Vol. 15, No. 6, pp. 529-547

ICDAR Competition (2003). http://algoval.essex.ac.uk/icdar

Jung, K.; Kim, K.I. & Jain, A.K. (2004). Text information extraction in images and video: a survey, *Pattern Recognition*, Vol. 37, No. 5, pp. 977-997

Karatzas, D. & Antonacopoulos, A. (2004). Text extraction from web images based on a split-and-merge segmentation method using colour perception, *Proceedings of Int. Conf. Pattern Recognition*, Vol. 2, pp. 634-637

Kim, I.J. (2005). Keynote presentation of camera-based document analysis and recognition, http://www.m.cs.osakafu-u.ac.jp/cbdar

Kim, J.; Park, S. & Kim, S. (2005). Text locating from natural scene images using image intensities, *Proceedings of Int. Conf Document Analysis and Recognition*, pp. 655-659

Kovesi, P.D. (2006). MATLAB and Octave functions for computer vision and image processing, School of Computer Science & Software Engineering, The University of Western Australia, http://www.csse.uwa.edu.au/~pk/research/matlabfns/

Li, H. & Doermann D. (1999). Text enhancement in digital video using multiple frame integration, *Proceedings of ACM Int. Conf. on Multimedia*, pp. 19-22

Liang, J.; Doermann, D. & Li, H. (2003). Camera-based analysis of text and documents: a survey, *Int. Journal on Document Analysis and Recognition*, Vol. 7, No. 2-3, pp. 84-104

Lienhart, R. & Wernicke, A. (2002). Localising and segmenting text in images, videos and web Pages, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 12, No. 4, pp. 256-268

Lopresti, D. & Zhou, J. (2000). Locating and recognising text in WWW images, *Information Retrieval*, Vol. 2, pp. 177-206

Lukac, R.; Smolka, B.; Martin, K.; Plataniotis, K.N. & Venetsanopoulos, A.N. (2005). Vector filtering for color imaging, IEEE Signal Processing, Special Issue on Color Image Processing, Vol. 22, No. 1, pp. 74-86

Luo, X.-P.; Li, J. & Zhen, L.-X. (2004). Design and implementation of a card reader based on build-in camera, *Proceedings of Int. Conf. Pattern Recognition*, pp. 417-420

Mancas-Thillou, C. (2006). *Natural scene text understanding*, PhD thesis, Faculté Polytechnique de Mons, Belgium

Mancas-Thillou, C. & Gosselin, B. (2006). Spatial and color spaces combination for natural scene text extraction, *Proceedings of Int. Conf. Image Processing*

Mancas-Thillou, C.; Mancas, M. & Gosselin, B. (2005). Camera-based degraded character segmentation into individual components, *Proceedings of Int. Conf Document Analysis and Recognition,* pp. 755-759

Mata, M.; Armingol, J.M.; Escalera, A. & Salichs, M.A. (2001). A visual landmark recognition system for topologic navigation of mobile robots, *Proceedings of Int. Conf. on Robotics and Automation*, pp. 1124-1129

Messelodi, S. & Modena, C.M. (1992). Automatic identification and skew estimation of text lines in real scene images, *Pattern Recognition*, Vol. 32, No. 5, pp. 791-810

Mukundan, R.; Ong, S.H. & Lee, P.A. (2001). Discrete vs. continuous orthogonal moments in image analysis, *Proceedings of Int. Conf. On Imaging Systems, Science and Technology*, pp. 23-29

Niblack, W. (1986). *An introduction to image processing*, Prentice-Hall, pp. 115-116

Ojima, Y.; Kirigaya, S. & Wakahara, T. (2005). Determining optimal filters for binarisation of degraded gray-scale characters using genetic algorithms, *Proceedings of Int. Conf Document Analysis and Recognition*, pp. 555-559

Otsu, N. (1979). A threshold selection method from gray level histograms, *IEEE Trans. System, Man and Cybernetics*, Vol. 9, No. 1, pp. 62-66, 1979

Perroud, T.; Sobottka, K.; Bunke, H. & Hall, L. (2001). Text extraction from colour documents - clustering approaches in three and four dimensions -, *Proceedings of Int. Conf Document Analysis and Recognition*, pp. 937-941

Plataniotis, K.N. & Venetsanopoulos, A.N. (2000). *Colour image processing and applications*, Springer Verlag

Sauvola, J. & Pietikainen, M. (2000). Adaptive document image binarisation, *Pattern Recognition*, Vol. 33, pp. 225-236

Sobottka, K.; Bunke, H. & Kronenberg, H. (1999). Identification of text on coloured book and journal covers, *Proceedings of Int. Conf Document Analysis and Recognition*, pp. 57-62

Sun, J.; Hotta, Y. & Katsuyama, Y. (2004). Low resolution character recognition by dual eigenspace and synthetic degraded patterns, *Proceedings of ACM Hardcopy Document Processing Workshop*, pp. 15-22

Thillou, C. & Gosselin, B. (2004). Segmentation-based binarisation for color degraded images, *Proceedings of Int. Conf. on Computer Vision and Graphics*

Thillou, C.; Ferreira, S. & Gosselin, B. (2005). An embedded application for degraded text recognition, *Eurasip Jour. on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems: methods and applications*, Vol. 13, pp. 2127-2135

Wang, B.; Li, X.-F.; Liu, F. & Hu, F.-Q. (2004). Colour text image binarisation based on binary texture analysis, *Proceedings of Int. Conf. Acoustics, Speech and Signal Processing*, pp. 585-588

Wesolkowski, S. (1999). Colour Image Edge Detection and Segmentation: a Comparison of the Vector Angle and the Euclidean Distance Colour Similarity Measures, Master thesis, University of Waterloo

Wolf, C.; Jolion, J. & Chassaing, F. (2002). Text localisation, enhancement and binarisation in multimedia documents, *Proceedings of Int. Conf. on Pattern Recognition*, pp. 1040-1057

Zandifar, A.; Duraiswami, R. & Davis, L.S. (2005). A video-based framework for the analysis of presentations/posters, *Int. Journal on Document Analysis and Recognition*, Vol. 7, No. 2-3, pp. 178-187

Zhang, J.; Chen, X.; Hanneman, A.; Yang, J. & Waibel, A. (2002). A robust approach for recognition of text embedded in natural scenes, *Proceedings of Int. Conf. on Pattern Recognition*

Zhou, J.; Lopresti, D. & Lei, Z. (1997). OCR for world wide web images, *Proceedings of SPIE on Document Recognition V*, Vol. 3027, pp. 58-66

17

Image Similarity based on a Distributional "Metric" for Multivariate Data

Christos Theoharatos, Nikolaos A. Laskaris, George Economou & Spiros Fotopoulos
Electronics laboratory, Dept. of Physics, University of Patras
Greece

1. Introduction

The problem of image similarity has become a challenging task in the field of computer vision through the last two decades. The assessment of (dis)similarity between color (or multichannel, in general) images or parts of images has been studied on several image processing application domains such as image indexing and retrieval, classification and unsupervised segmentation (Rubner et al., 2001). The basic operations that need to be carried out in order to estimate the similarity between two color images are three-fold (Stricker & Orengo, 1995): first, choose an appropriate color space for image representation; then, extract a signature for each image (using, commonly, low-level features) to construct a theoretically valid distribution; finally, establish pairwise comparisons based on these signatures. Each signature constitutes the content description of a corresponding image. It is summarized based on pixel attributes and provides a representation of the image in a multidimensional feature space. There, a proper (dis)similarity measure is defined in order to act as a general rule for comparing any given pair of images.

In these directions, several (dis)similarity measures have been developed and used as empirical estimates of the distribution of image features, confirming that distribution-based measures exhibit excellent performance in all areas (Rubner et al., 2001). In the context of visual image similarity, we make use of a nonparametric test from the field of multivariate statistics that deals with the "Multivariate Two-Sample Problem", originally presented by Friedman and Rafsky (1979). The specific test is a multivariate extension of the classical Wald-Wolfowitz test (WW-test) and compares two different samples of vectorial observations (i.e. two sets of points in $\mathbf{R}^p$) by checking whether they form different branches in the overall minimal spanning tree (MST) (Zahn, 1971). It provides an aggregate gauge of the match between color images, taking into consideration all the selected characteristics, while alleviating correspondence issues. The output of this test can be expressed as the probability that the two point-samples are coming from the same distribution. We have proven that this is a powerful measure for image similarity, relying on the statistical comparison of content representations in a properly defined feature space (Theoharatos et al., 2005).

Here, the above distributional-'metric' is introduced in conjunction with a prototyping method that dramatically speeds up the execution of the involved computations and results

in an efficient overall methodology (e.g. so as to be used in highly demanding applications such as image retrieval tasks). The current proposal incorporates the use of a computational intelligent module for content representation based on self-organizing neural networks (SONNs), the Neural-Gas algorithm (Martinez et al., 1993), which is responsible for generating a parsimonious description of the color distribution of each image. The multivariate distributions representing the individual images are then compared via the standard WW-test, providing enhanced performance when evaluated via a query-by-example image retrieval scheme (Theoharatos et al., 2006a).
Finally, we are discussing the applicability of the same distributional distance in order to compare images following a standard JPEG-format (Wallace, 1991) and with the scope to emphasize texture characteristics during the visual search. Color and texture features are directly extracted from the DCT-compressed domain, in the form of an ensemble of feature vectors that are the inputs to a standard WW-test. The emerging indexing scheme is found to be robust, providing invariant similarity results when image rotation is considered (Theoharatos et al., 2006b).

2. Background and related work

Research on image similarity has expanded lately, mainly due to the increased interest of content-based image retrieval (CBIR), which constitutes a highly challenging research area with the emerging techniques sharing many advantages (Smeulders et al., 2000). Even though the focus of interest for image similarity and retrieval has recently shifted towards the identification of high-level semantics from the content of the images (Eakins, 2002), not much success has been achieved so far. This is mainly due to the great difficulties in the derivation of semantically meaningful information at a general level (Sheikholeslami et al., 2002). As a consequence, nowadays methods are still constrained to use low-level visual features such as color, shape and texture to represent the image content.
Considerable investigation has been carried out on the basis of color content (Schettini et al., 2001). Color information has been recognized as the most important indicator of the general 'mood' of an image and is considered to capture, to a certain extent, image semantics. In the existing literature, researchers have experimented with different color spaces such as RGB, CIE-Lab, etc. (Castelli & Bergman, 2002), various color descriptors such as color histogram (Swain, & Ballard, 1991), color moments (Stricker & Orengo, 1995) and chromaticity moments (Paschos et al., 2003), and also miscellaneous similarity measures such as histogram intersection (Swain & Ballard, 1991), quadratic form distance functions and statistical indices (Rubner et al., 2001). The most popular representation of color information is the global histogram, which statistically denotes the joint probability of intensities of the three-color channels, thus describing the global color distribution in an image. In general, the color histogram provides useful clues for the subsequent expression of similarity between images, due to its robustness to background complications and object distortion. Moreover, it possesses translation, scale and rotationally invariant characteristics. A profound number of (dis)similarity measures have been proposed for computing the distance between histograms from two different images. In their work, Rubner et al. (2001) distinguished these measures generally into four categories: heuristic histogram distances, nonparametric test statistics, information-theory divergences and ground distance ones. In the context of image indexing and retrieval, the different variants of the color histogram-related methodology have provided satisfactory results, especially in practical situations in

which the feature extraction step needs to be accomplished as simply and promptly as possible. Soon it became popular, since it was very simple to implement and exhibits fast retrieval response time, making it a good candidate for real-time applications. However, the performance of this technique was not found to be high enough, mainly due to the necessary trade-off during the binning procedure. An adequate compromise could be achieved via the use of an adaptive binning procedure (e.g. Leow & Li, 2004), in which the histogram bins would adapt to the actual distribution of colors in images. Apart from the facts that bin-adaptation can be a computational demanding task and, in general, is still considered an open issue in the field of image processing, existing systems adopt fixed-binning histograms since most dissimilarity measures are unable to cope with histograms build over different sets of bins (Rubner et al., 2001).

In order to overcome the above limitation, an attempt was made recently by Rubner et al., (2000) to combine the benefits from the use of a distribution distance with a flexible description of color-content that adapts its resolution to individual images. The innovative work mentioned above introduced the Earth Mover's Distance (EMD), a computational demanding task based on the solution of the well-known transportation problem. In summary, a representation scheme suitable for color distributions and based on Vector Quantization (VQ) preceded the computation of EMD between pairs of distributions. In this scheme, after the complicated k-d trees algorithmic procedure for cluster analysis, each distribution was represented by means of a number of cluster-centroids and the corresponding proportions of image pixels with colors within the identified groups. The EMD-related technique was shown to be more robust than histogram-matching techniques, since it could operate on variable-length representations of the distributions that were avoiding quantization problems related with the binning procedure. In short, higher performance was achieved at the expense of computational efficiency. However, the integrated representation design is not related directly to the reliability of color distribution. Although the efficiency of k-d tree algorithm is generally recognized, their effectiveness for clustering data of complex distributions or data with high correlations among variables is questionable. Moreover, there is lack of supporting evidence in the field of statistics that EMD is indeed an appropriate measure for comparing multivariate distributions, apart from the theoretical benefit that correlates with perception when applied in the CIE-Lab space.

In the context of textural features, these are also represented using histogram-based methodologies. Indexing, similarity and retrieval of compressed images have recently become a very active research area, since the great amount of digital images provided on the WEB are stored in JPEG format (Wallace, 1991). In particular, the JPEG compression standard applies DCT transform in order to achieve a large amount of compression, significantly reducing the image size. Such compression is suitable for Internet-based applications, reducing the storage space while increasing the downloading speed. Thus, measuring image similarity directly in the compressed domain becomes more and more beneficial, compared to the pixel-based one. To bridge the gap between compressed- and pixel-space, where the majority of image processing algorithms are developed, recent research is now starting apace to develop content feature extraction algorithms working directly in the compressed domain (e.g. Zhong & Jain, 2000; Ngo et al., 2001; Jiang et al., 2004). Since the inverse DCT (IDCT) is an embedded part of the JPEG decoder and the DCT itself is one of the best filters for feature extraction working directly on the DCT domain, it has proven to be a well-promising area for image similarity in the compressed domain. DCT

has, to a certain extent, unique scale invariance and zooming characteristics, which can provide insight into objects and texture identification (Ngo et al., 2001). In addition, it exhibits a set of good properties such as energy compaction and image data decorrelation and, therefore, is naturally considered to be a potential domain in mining visual information. Thus, direct feature extraction from DCT domain can provide better solutions in characterizing the image content, apart from its advantage of eliminating any necessity of decomposing an image and detecting its features in the pixel domain (Jiang et al., 2004).
The rest of the presentation is organized as follows. Section 3 provides an overview of the proposed distributional-'metric' for comparing multivariate data, including the graph-theoretic framework of MST and the multivariate WW-test. Color image similarity is presented in Section 4, using the Neural-Gas network for expressing the image content-signature. In Section 5, visual similarity in the compressed domain is analysed by extracting color and texture attributes directly from the DCT-space. Finally, conclusions are drawn in Section 6, along with an outline of our future research objectives in Section 7. Throughout our study, image similarity is evaluated via a query-by-example image retrieval scheme

3. The Distributional 'Metric' for Comparing Multivariate Data

A *nonparametric* test dealing with the *"Multivariate Two-Sample Problem"* (Friedman & Rafsky, 1979) is proposed for measuring image similarity in a reliable and more sophisticated way. The specific test is a multivariate extension of the classical statistical test of Wald and Wolfowitz and compares two different samples of vectorial observations (i.e. two sets of points in $\mathbf{R}^p$). The output of the test can be expressed as the probability that two point-samples are coming from the same distribution. Its great advantage is that no a-priori knowledge about the distribution of points in the two samples is a prerequisite (Theoharatos et al., 2005). This model-free assumption stems from the graph-theoretic origin of the WW-test, which is actually based on the concept of the MST-graph (Zahn, 1971). For this reason, a compact description of MST is preceded first.

3.1 MST-Graph Representation

Given the establishment of a systematic procedure for extracting low-level characteristics from a color (or multivariate, in general) image that are individually represented as vectors in a predetermined space, one can rely on graph theory to provide a collective perspective that captures the essence of the visual content of the image under study. Graph theory, by putting emphasis on the structural relationships between the extracted characteristics, provides robust descriptions against noise degradation widely and randomly spread over the field and simple transformations like image scaling. Specifically the MST-graph appears as an extremely useful condensation of the bulk of information conveyed within the ensemble of image characteristics. In addition, the MST provides a compact description of a point set. It contains the '*nearest neighbor*' information about each point and the '*shortest linkage*' information about subsets of points (Laskaris & Ioannides, 2001). In his study, Zahn (1971) established another advantage of MST, the *determinacy*, meaning that the results from the application of a method working with MST-graph do not depend on random choices or the order in which points are scrutinized, but are affected solely by the point set provided as input. Overall, the MST structure is unchanged under transformations like translation,

rotation and non-linear ones, preserving the ordering of edge lengths (Theoharatos et al., 2005).

Graph theory sketches the MST structure with the following definitions (Zahn, 1971). A *graph* $G(V,E)$ is a mathematical structure for representing pairwise relationships among data. It consists of a set of points called *nodes* $V=\{V_i\}_{i=1:N}$ (or *vertices*) and a set of links $E=\{E_{ij}\}_{i\neq j}$ between nodes called *edges* (or *lines*). An edge links two nodes defining it, when it is incident on both of them. The *degree* d_i of a node is the number of edges incident to it. When a weight e_{ij} is assigned to each link, a weighted-graph is formed and in the particular case that $e_{ij}=e_{ji}$ this graph is called *undirected weighted graph*. A *connected graph* has a path between any two distinct nodes and a *tree* is a connected graph with no cycles. A *subgraph* of a given graph is a graph with all of its nodes and edges in the given graph. A *spanning tree T* of a (connected) weighted graph $G(V,E)$ is a connected subgraph of $G(V,E)$ such that: (i) it contains every node of $G(V,E)$, and (ii) it does not contain any cycle. The *MST* is a spanning tree containing exactly $N-1$ edges, for which the sum of edge weights is minimum.

Suppose now that $N-$pixels are randomly selected from an image and the corresponding RGB vectors are represented as an ensemble of points in the feature space. The specific points are used as the nodes of the original (fully-connected) graph, while the interpoint Euclidean distances as the weights of the corresponding edges. Using a standard algorithm (Prim, 1957), the MST is evolved from the original graph, offering a parsimonious description of the low-level information in an image. Given a second image, the color content of which is to be compared with the content of the first one, we can proceed with the selection of pixels as previously and transform the comparison between feature-contents into a comparison between the corresponding MST-graphs (Theoharatos et al., 2005). To perform such a comparison, a well-defined statistical test is available in the literature of multivariate statistics.

3.2 The Multivariate WW-Test

Consider samples of size *m* and *n* respectively from distributions F_x and F_y, both defined in $\mathbf{R}^p$. The hypothesis H_0 to be tested is whether they are coming from the same distribution, thus $F_x=F_y$. We are interested in the rejection of the original hypothesis, which is the alternative hypothesis $F_x\neq F_y$. In the univariate case ($p=1$), the WW-test begins by sorting the $N=m+n$ univariate observations in ascending order. Friedman and Rafsky (1979) proposed the use of MST as a multivariate generalization of the univariate sorted list, introducing in this way a methodology to define the two-sample test statistics based on the MST in analogy with those based on the sorted list.

In the multivariate case, the hypothesis H_0 to be tested is whether two multidimensional point samples $\{X_i\}_{i=1:m}$ and $\{Y_i\}_{i=1:n}$ are coming from the same multivariate distribution. In this general case, the WW-test can be summarized with the following steps (Friedman & Rafsky, 1979): (i) Consider samples of size *m* and *n* respectively from distributions F_x and F_y, both defined in $\mathbf{R}^p$, (ii) Construct the overall MST without encountering the sample identity of each point (iii) Delete the edges for which the defining nodes originate from different samples. Then, based on the sample identities of the points the test statistic R is

computed, defining the total number of *runs*, while a *run* is defined as a consecutive sequence of identical sample identities. R can be also defined as the number of disjoint subtrees that finally result. In order to illustrate the WW-test for ease in understanding, two randomly selected samples of size $m=5$ and $n=8$ are used in the 2-D of Fig. 1. After deleting those edges coming from different distribution, the number of disjoint subtrees is calculated and found equal to $R=5$. It must be pointed out here that, the MST possesses two significant properties which make it appropriate for application to the multivariate two-sample problem (a) it connects all the $N-$nodes with $N-1$ edges, which comes from the fact that the MST is a spanning tree and (b) the node pairs defining the edges represent points that tend to be close together, which stems from the requirement that the sum of the edge weights is minimum.

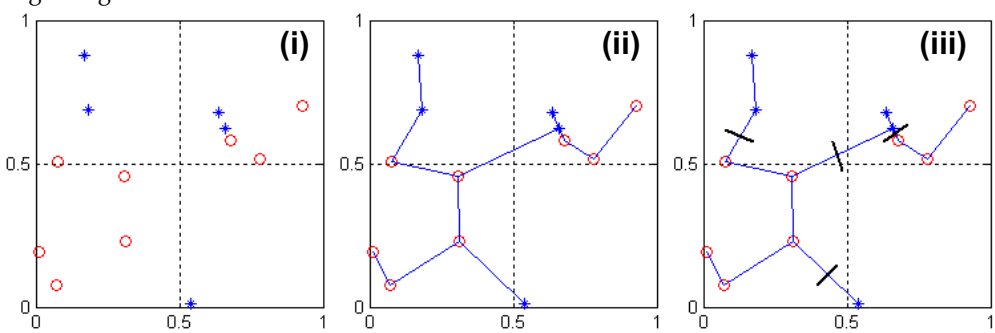


Fig. 1. Visual configuration of the multivariate WW-test algorithmic procedure for two randomly sampled distributions: (i) consider the two sample-distributions, (ii) construct the overall MST and (iii) delete the edges of the nodes originating from different distributions.

The null distribution of the test statistic is derived, based on the combinatorial analysis given by Friedman and Rafsky (1979). Let $N=m+n$, C be the number of edge pairs of MST sharing a common node, and d_i be the degree of the ith node. Then, $C=\frac{1}{2}\sum_{i=1}^{N} d_i(d_i-1)$. Number the $N-1$ edges of the MST arbitrarily and define Z_i, $1 \le i \le N-1$, as:

$$Z_i=\begin{cases}1 & \text{if the } i\text{-th edge links nodes from different samples}\\ 0 & \text{otherwise}\end{cases}$$

Then, $R=\sum_{i=1}^{N-1} Z_i+1$. Under H_0, the mean and variance of R can be calculated using a systematic analysis as follows:

$$E[R]=\frac{2mn}{N}+1,\ Var[R\,|\,C]=\frac{2mn}{N(N-1)}\times\left\{\frac{2mn-N}{N}+\frac{C-N+2}{(N-2)(N-3)}[N(N-1)-4mn+2]\right\}$$

It has been shown that the quantity:

$$W=\frac{R-E[R]}{\sqrt{Var[R]}} \tag{2}$$

approaches (asymptotically) the standard normal distribution while $E[R]$ and $Var[R]$ are given in closed form based on the size of the two samples (Friedman & Rafsky, 1979). The

importance of the previous is that by using simple formulae, the *significance level* (and *p-value)* for the acceptance of the hypothesis H_0 can be readily estimated.

4. Comparing Color Distributions via a self-organizing algorithm

Regarding the plethora of methods and feature extraction techniques, image indexing and similarity is associated with different levels of image understanding. Provided that a number of feature vectors are given, the resulted feature space may not be uniformly occupied. Sheikholeslami et al. (2002) studied the way clustering individuates the sparse and dense pixel-areas in the image, revealing the underlying distribution of the feature space. In addition, a vector quantization scheme realizes a concise representation of the input data regardless of the actual meaning and significance of the clusters (Gdalyahu et al., 2001). The resulting codebook vector can be considered as a compact description of the data distribution (e.g. the color information of database images), providing effective and alternative ways to portray image content.

To avoid missing the generality of the approach and at the same time propose its efficient/intelligent version, the use of two sequential modules is illustrated in the specific domain of color image information management, which considers the RGB-vectors corresponding to individual pixels (i.e. points in $\mathbf{R}^3$). In a nutshell, using Neural-Gas based prototyping a data-summary will be produced, which constitutes a meaningful sampling from the underlying color distribution of each image. With the subsequent application of WW-test to compare samples of color prototypes, robust and economical comparisons regarding color content will be achieved (Theoharatos et al., 2006a).

4.1 Representation of Color Distributions via Self-Organizing Networks

Summarizing data distributions via prototypes has roots in the theory of VQ, which is a powerful strategy for data compression and can be accomplished via different techniques (Gray, 1984). Briefly, a vector quantizer encodes a data manifold $V \subseteq \mathbf{R}^p$ utilizing only a finite set of reference or "codebook" vectors $O_j \in \mathbf{R}^p$, $j = 1,\ldots,k$, which are also called cluster centers. Each data vector $X \in V$ is described by the best-matching reference vector $O_{j(X)}$ for which the distortion error $d(X, O_{j(X)})$, usually measured via the squared Euclidean distance, is minimal. The main core of the procedure depends on the division of the original manifold V into a number of subregions V_i called Voronoi polygons or Voronoi polyedra, out of which each data vector X is described by the corresponding reference vector O_j. The efficient application of VQ mainly depends on the codebook design, i.e. the proper selection of reference vectors. For this critical step, the use of traditional clustering algorithms like the k-means had been originally proposed. However, it was experimentally verified later that these algorithms often lead to a suboptimal choice of reference vectors O_j in the case of nontrivial data distributions, as well as in the case of an inappropriate selection for the number of reference vectors. Such a suboptimal solution can have a significant impact on the subsequent encoding of the data and even result to highly distorted representations.

The tremendous development of neural theory of unsupervised learning and the related algorithms of Self-Organizing Neural Networks (SONNs) revitalized the field of VQ. The ability to efficiently deduce prototypes from the data, common in many SONNs like the Kohonen's feature map (Kohonen, 1997) and the Neural-Gas (Martinez et al., 1993), could be

exploited in the reliable codebook design. For a thorough treatment of SONNs and their applications related with VQ, the interested reader can refer to the seminal study of Martinez and Schulten (1994).

Stochastic presentation of the input data, competition among the neural nodes (to which weight vectors $A_j \in \mathbf{R}^p$ have been assigned) and a 'soft max' adaptation rule are the common characteristics of these networks that guarantee the fast convergence to a set of weight vectors (i.e. prototypes), which can serve as a high-fidelity codebook. The resulting codebook vectors are allocated according to the probability distribution of data vectors over the manifold V, and in such a way that the average distortion error is minimized. The main difference between the SONN-algorithms compared to other traditional clustering methodologies is that not only the best-matching reference vector $O_{j(X)}$ is adjusted every time a data vector X is presented, but also the reference vectors adjacent to it are updated accordingly. Among the SONNs, Kohonen's feature map is the most popular mainly due to the accompanying visualization scheme that enables the projection of the input data nonlinearly onto a lower dimensional lattice (Kohonen, 1997; Haykin, 1999). Inspired by the possibility that some high level organization in the brain may be created during learning through self-organization, Kohonen (1997) presented a self-organizing learning algorithm that presumably produces feature maps similar to those occurring in the human brain. In this way, the self-organizing map (SOM) forms a nonlinear regression of the ordered set of reference vectors into the input space. The reference vectors constitute a low-dimensional network that follows the original data distribution; for this reason, it is also referred to as '*self-organizing semantic map*'. However, to obtain efficient quantization results with Kohonen's feature map algorithm, the topology of the lattice has to match the topology of the data manifold V that is to be represented. Since the primary interest in our study lies in the precise quantization of the data and not in dimensionality reduction, we avoided the use of Kohonen's network. Instead, we resorted to Neural-Gas network, which had been proven to quickly converge to distortion-errors lower than the ones achieved using Kohonen's algorithm or other classical clustering algorithms (Martinez et al., 1993).

4.2 The Neural-Gas Algorithm for Vector Quantization

For the purposes of vector quantization, the Neural-Gas algorithm is presented in this step and utilized in the dual segregation algorithmic procedure for our efficient image similarity methodology. It is a neural network algorithmic procedure that sustains specific properties that make it appropriate as a feature extraction scheme: (1) it converges quickly to low distortion errors, (2) it reaches a distortion error much lower than the corresponding using the K-means clustering and other traditional techniques or the one resulting from the SOM-approach, and (3) it obeys a gradient descent on an energy surface, in contrast to the Kohonen's feature map network (Martinez et al., 1993).

In the Neural Gas network algorithm, a stochastic sequence of incoming data vectors $X(t)$, $t = 1,2,\ldots,t_{\max}$, which is governed by the distribution $P(X)$ over the manifold V, drives the adaptation step for adjusting the weights of the *k* neurons $\{A_j\}_{j=1:k}$ (i.e. the reference vectors)

$$\Delta A_j = \varepsilon\, h_\lambda\left(f_j\left(X(t),\{A_i\}_{i=1:k}\right)\right)\left(X(t) - A_j\right),\ j = 1,\ldots,k,\ \forall\, t = 1,\ldots,t_{\max} \tag{1}$$

The function $h_\lambda(y)$ in the above equation has an exponential form $e^{-y/\lambda}$ and $f(X,\{A_i\})$ is an indicator function that determines the 'neighbourhood-ranking' of the reference vectors according to their distance from the input vector X. For both parameters ε and λ, an exponentially decreasing schedule is followed, with $t_{\max}$ being the final number of adaptation steps that can be defined from the data based on simple convergence criteria (for analytical details refer to Martinez et al. (1993), see also Martinez & Schulten (1994)).
Martinez et al. (1993) mathematically proved that the asymptotic density distribution of the codebook vectors $P(A)$ was proportional to the data density $P(A) \propto P(X)^{\underline{d}/(\underline{d}+2)}$, where $\underline{d} \leq d$ is the intrinsic dimension of the input data. This theoretical proposition along with the accompanying experimental evidence, showing that the Neural-Gas network is indeed capable of representing successfully data-manifolds with even intricate intrinsic geometries (Martinez & Schulten, 1994), motivated our conjecture that the designed codebook could serve as a faithful representation of the vectorial distribution in color-space. Therefore, it could be utilized in the subsequent comparisons regarding color content.
Fig. 2 illustrates the color-content representation through Neural-Gas prototypes, which clearly evidences that the distribution of the codebook vectors follows very closely the corresponding color distribution (Theoharatos et al., 2006a). In the depicted figure, three images are included (two of which "look similar" to each other), while their RGB distributions corresponding to all the pixels are shown in the left column along with their representations using the associated codebooks. In addition, the entire set of pixels comprising each RGB-image is presented as a black dot-swarm. It is clearly evident that the distribution of the codebook vectors follows very closely the corresponding color distribution. Therefore, each codebook can be thought of as a properly "down-sampled" version of the original RGB-distribution (Laskaris & Fotopoulos, 2004). Aiming at higher computational efficiency, an intermediate step of subsampling has been introduced between embedding an image in RGB-space and Neural-Gas based prototyping. Within this step, only a small portion (~5 %) of the pixels in the image is selected using uniform random sampling, and the associated vectors are used as input data to the neural network. The comparison of the codebooks designed with (right column) and without (left column) the subsampling step, shows only slight differences.

4.3 Comparing Color Signatures using the WW-test

In order to assess the similarity between two color images, the WW-test is utilized as follows. Provided the two color codebooks $\{A_i\}_{\text{i=1:k}}$ and $\{B_i\}_{\text{i=1:k}}$ extracted from a pair of images, the WW-procedure follows, with the extracted prototypes playing the role of the input point-samples $\{X_i\}_{\text{i=1:m}}$ and $\{Y_i\}_{\text{i=1:n}}$ respectively. W is computed based on the involved codebook vectors and used as a similarity measure in a way that the more positive its value is, the more similar the color distributions in the two images are (Theoharatos et al., 2005). The $W-$quantity computed between pairs of images plays the role of a "distributional distance" and therefore inherits interesting invariant-characteristics. In the past, a few other statistical indices have been proposed, as well, as means of measuring similarity between color distributions. These distances, for instance the Kolmogorov-Smirnov distance (KS), the chi-square test (χ^2-statistic), etc. (Rubner et al., 2001), measure

how unlikely it is that one distribution is drawn from the population represented by the other.

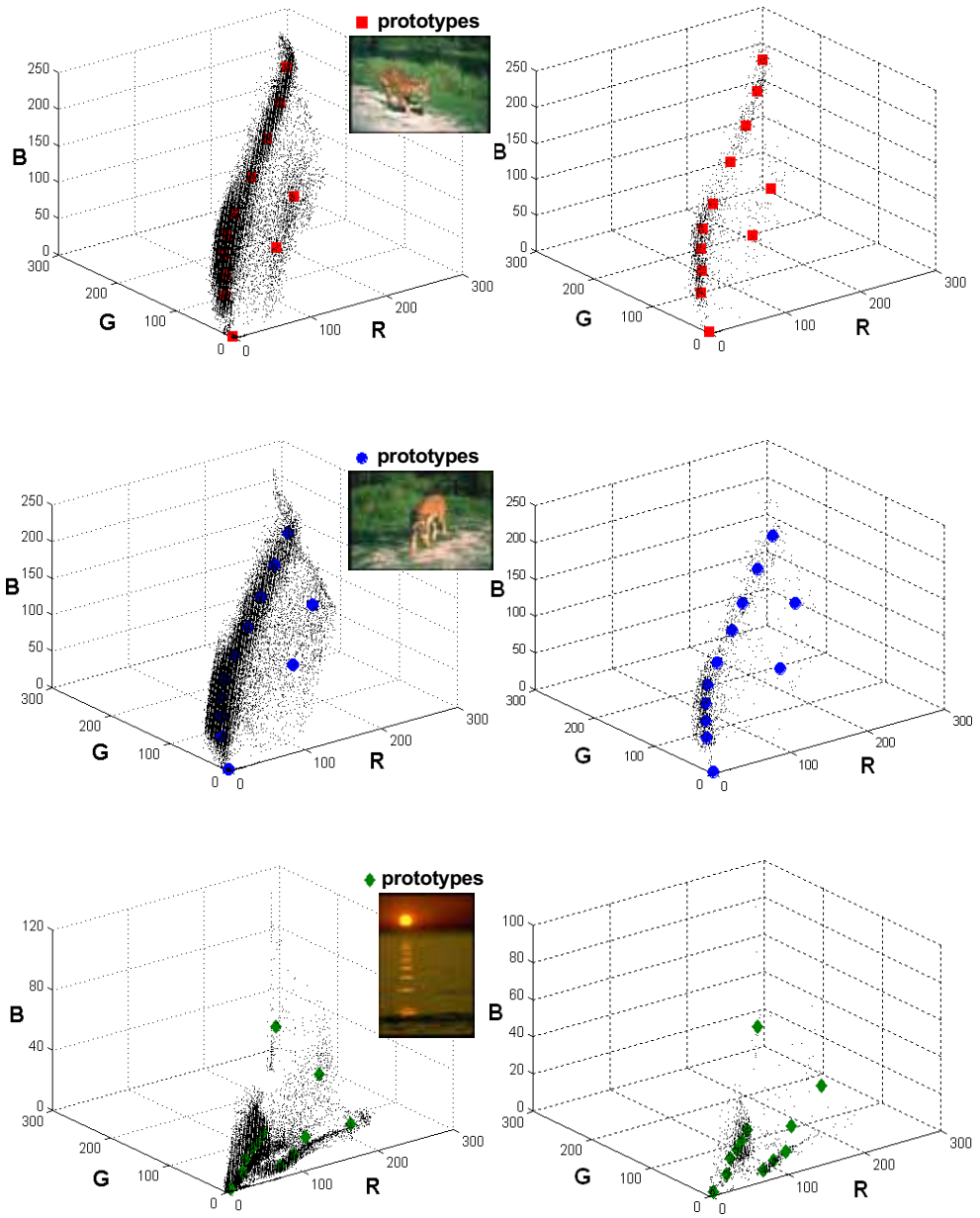


Fig. 2. Codebook color representation based on the Neural-Gas algorithm. For each image, a point-distribution is formed using the RGB-vectors corresponding to all (left panel) or a small portion (right panel) of the pixels, summarized through 12 prototypes.

Fig. 3 provides a demonstration of the test performance using the images and the codebooks presented in Fig. 2. In both panels different labels are associated with each of the two images to be compared. The $k = 12$ color prototypes extracted from each image are located in the RGB-space and the points indicating their position have been labeled according to the image they are coming from. By contrasting the two MSTs, it becomes evident that in the case of similar images (Fig. 3a) there are many edges having different labeled nodes as endpoints, while only a few in the case of dissimilar ones (Fig. 3b).

The unique benefit of WW-approach is that since it engages "distributional distance" acting on samples of image constituents, the emerging similarity measure possesses desirable invariant characteristics, such as rotation and translation invariance. Part of the flexibility is due to the statistical nature of the core procedure, the WW-test, and specifically its multivariate orientation. Theoharatos et al. (2005, 2006a, 2006b) have shown that not only different image characteristics can - in principle - be combined naturally in one type of query (i.e. color plus texture features), but also different types of queries can evolve independently and their results can be compared across types, as in the case of an image retrieval system. The latter is a direct consequence of the fact that the measured W-index relates directly to significance level and therefore can be used as an absolute measure to rank among the results of different types of query. Under these perspectives, the WW-test can be directly incorporated in retrieval processes from large image libraries, with the great advantage of being suitable for dealing with multivariate distributions.

4.4 Experimental evaluation via a query-by-example image retrieval scheme

In order to demonstrate and validate the effectiveness of the proposed methodology, a query-by-example image retrieval system was built. The image database included in the retrieval scheme contains a subset of $D = 1000$ color images from the Corel Collection. The utilized image-set was formed by pre-assigning the images into 20 distinct classes of $S = 50$ semantically similar images. A subset of $Q = 60$ query images from this heterogeneous set was also included in our retrieval system (three images per category). For the evaluation of the retrieval results, the *precision* (Pr) and *recall* (Re) indices (Castelli & Bergman, 2002) were adopted.

In the introduced methodology, the results are coming from different settings of the involved parameters. For different codebook sizes, the Precision was computed as a function of the number of RGB-vectors randomly sampled from each image and used as input data to the Neural-Gas network. The graphs obtained in this way showed that the Pr-index approached a relatively high value very soon (~1% of the pixels) and remained practically constant beyond the number of approximately 5% of the pixels (Theoharatos et al., in press a), which was the typical value chosen used throughout our evaluation study. In addition, by experimenting with the size of codebook vectors k that need to be drawn from each color image, extensive measurements have confirmed that after extracting $k = 25$ prototypes the Pr-index remained almost constant (Theoharatos et al., in press a). These results show that our method reaches the maximum performance for a moderate size of codebooks ($k \approx 25$) and therefore a more detailed representation of color distribution is unnecessary. This observation is very important for finding the best trade-off between effectiveness and efficiency when applying our algorithmic procedure.

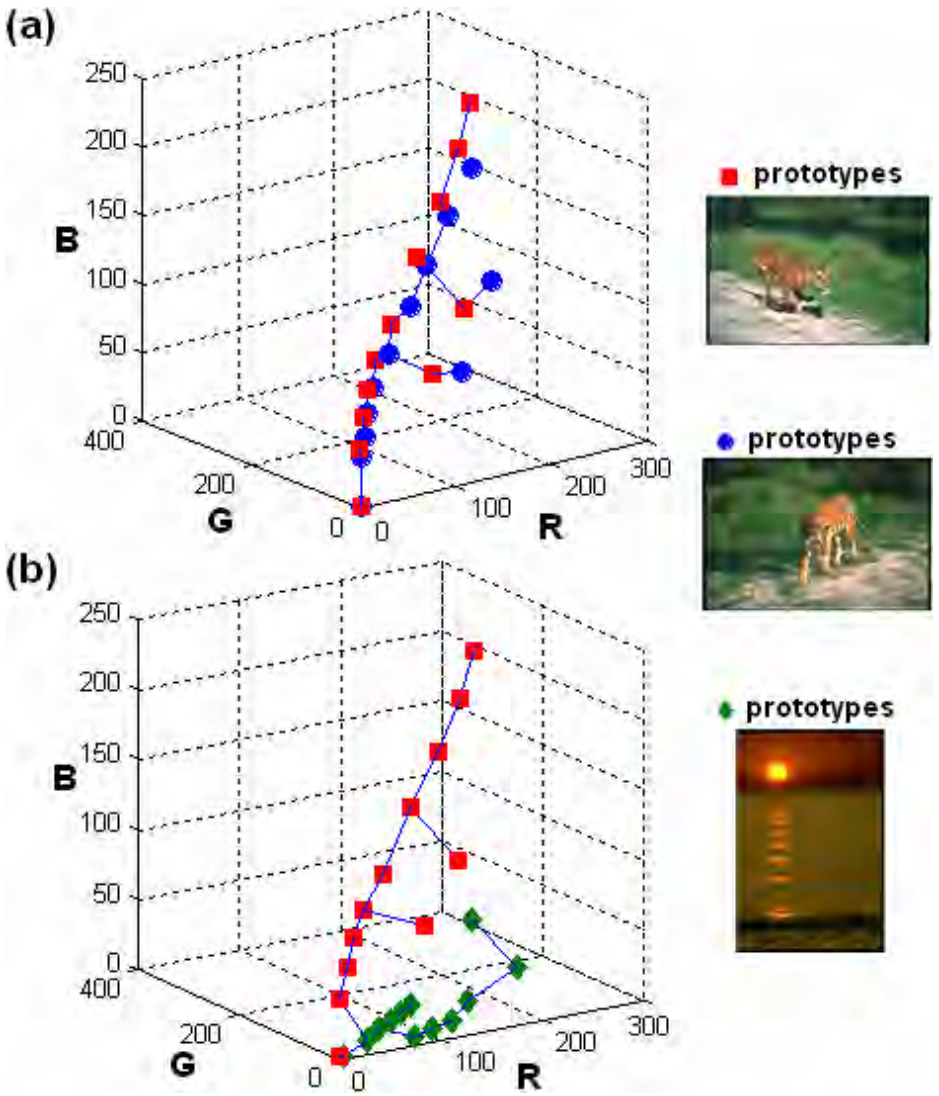


Fig. 3. WW-test for a pair of similar images (a) and dissimilar images (b), based on the k=12 color prototypes shown in Fig. 2. In the top panel, there are 19 edges having differently labeled nodes as endpoints and therefore splitting the overall MST into 20 subgraphs, thus R=20 (W=2.6523). On the contrary there are only 2 such edges in the bottom panel, thus R=3 (W=-4.8361).

For the full justification of our proposal, precision measurements regarding query-by-example search in the specific database are included. A plot of Pr-index as a function of the codebook size k is presented in Fig. 4, for the $T = 10$ top retrieved images of the selected list. The performance of the Neural-Gas based WW-test is compared to the one using the EMD-

metric (Rubner et al., 2000) when applied upon the corresponding Neural-Gas based color-signatures. Although the depicted curves follow - as theoretically expected - a relatively similar trend, the WW-test outperforms the EMD-measure; for $k=10$ a satisfactory improvement of ~5% is apparent, while for $k \geq 15$ a significant increase in performance (~10%) is depicted. In addition, the general trend of the depicted curve is very interesting. The Pr-index reaches a plateau pretty soon and remains almost constant above the codebook size of $k=25$. This observation is of great importance regarding the involved computational load of our method and will be discussed in the last Section. The slightly decreasing trend that becomes apparent after the size of $k=40$ is a by-product of the fact that the number of extracted codebook vectors is increased without increasing the number of sampled vectors from the image. Therefore the Neural-Gas network attempts a detailed representation that is adapted to the idiosyncrasies of the random sample and tends to capture stochastic variations as delicate data-structure (a common-place problem in neural networks, usually referred to as over-training). By experimenting with greater sample sizes (~10% of the pixels), this trend is drastically reduced.

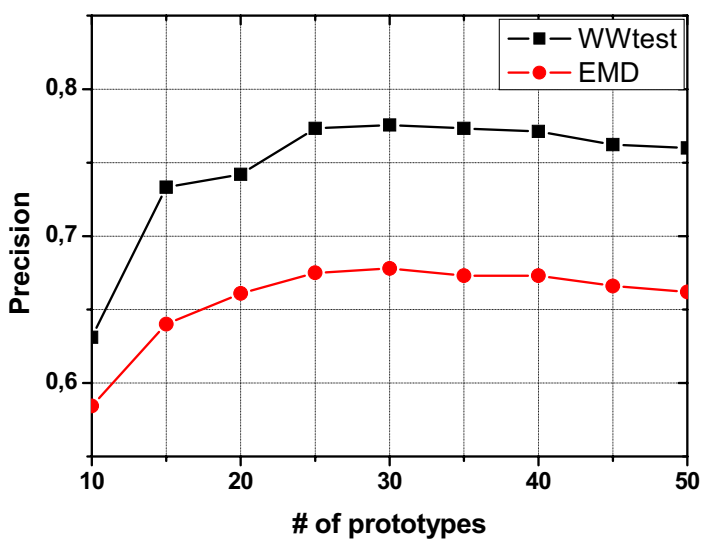


Fig. 4. Precision measurements of the WW-test and the EMD-related metric based on the same color codebooks, as a function of codebook size k.

The performance of the hybrid methodology as a method for accessing image databases was also evaluated following the standard procedure of constructing the Precision vs. Recall diagram. The Pr- and Re- indices were first evaluated for different sizes *T* of the selected list (for $T=5:5:30$), and the computed values were used in the plot of Fig. 5. The corresponding diagrams for other dissimilarity measures (HI, χ^2-test and JD using color histograms introduced in Rubner et al. (2000) and EMD applied on color signatures presented in Rubner et al. (2001)) have also been included in the same figure, enabling the direct comparison of the different approaches. It is clearly obvious that the WW-engine significantly outperforms all other methodologies.

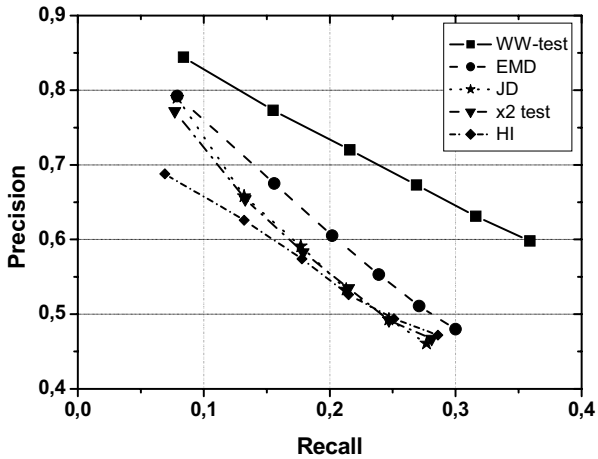


Fig. 5. Precision vs. Recall diagrams for the new hybrid approach, in comparison to other related techniques.

5. Visual similarity in the compressed domain

The flexible character of the WW-methodology relies on the multivariate flavour of the core statistical procedure. By altering the feature-extraction implementation, complementary ways to portray the image content appear without scaling effects or different cardinalities of the feature sets. An attempt is described here to adopt our methodology so as to work in compressed image domains that have recently gained high popularity (e.g. Zhong & Jain, 2000; Ngo et al., 2001; Jiang et al., 2004). This is expected not only to increase the efficiency of WW-based similarity scheme - by avoiding image decompression -, but also to constitute it suitable for novel applications like searching and retrieval in the World-Wide-Web - since the images of the Web are mostly included in a standard compressed format - (Jiang et al., 2004). Within this part we focus specifically on images from the standard JPEG compression scheme (Wallace, 1991). Competent ways to extract feature vectors directly from the zig-zag DCT-coefficients of the images are explored and their effectiveness is studied when exploited within the general framework of WW-methodology (Theoharatos et al., 2006b).

Color and texture features are utilized directly from the DCT-domain in the form of an ensemble of feature vectors represented in the YCrCb tri-chromatic model, in line with the JPEG standard (Wallace, 1991). In order to represent color information from each $N \times N$ pixel-block of a given image, all DC components are separately extracted and used as input vectors in the WW-engine to form a 3-D vector space. Texture features, on the other hand, can be defined as the spectrum energies in different localizations of a local block. Since the DC coefficient $F_{0,0}$ represents the average grayscale value of each $N \times N$ macroblock, it is not considered to carry any texture information. The remaining AC coefficients can be considered to characterize image texture and be used as texture features. Zhong and Jain (2000) pointed out that even though the DC component is used for color feature characterization and the remaining AC components for texture features, color and texture attributes are mixed together in the $(N \times N) - 1$ coefficients contained inside a pixel-block.

Most of the times, it is extremely hard to draw an absolute line between color and texture attributes, since color variation results in color texture. In this way, color is expected to be present at several AC coefficients, packing most of its spectral energy in the fewest number of low-frequency coefficients at the upper left corner of the macroblock. Zhong and Jain (2000) proposed to compute the absolute values of the AC coefficients, selecting those *M* lowest-frequency features carrying most of the energy. By rotating an image-block, the absolute values of the set of contained DCT coefficients remains unaltered, but their position along each zig-zag line is changed. However, by computing the distance between the corresponding matrices for the initial block and its rotated version, a totally false alarm is resulted in accordance with their perceptual similarity.

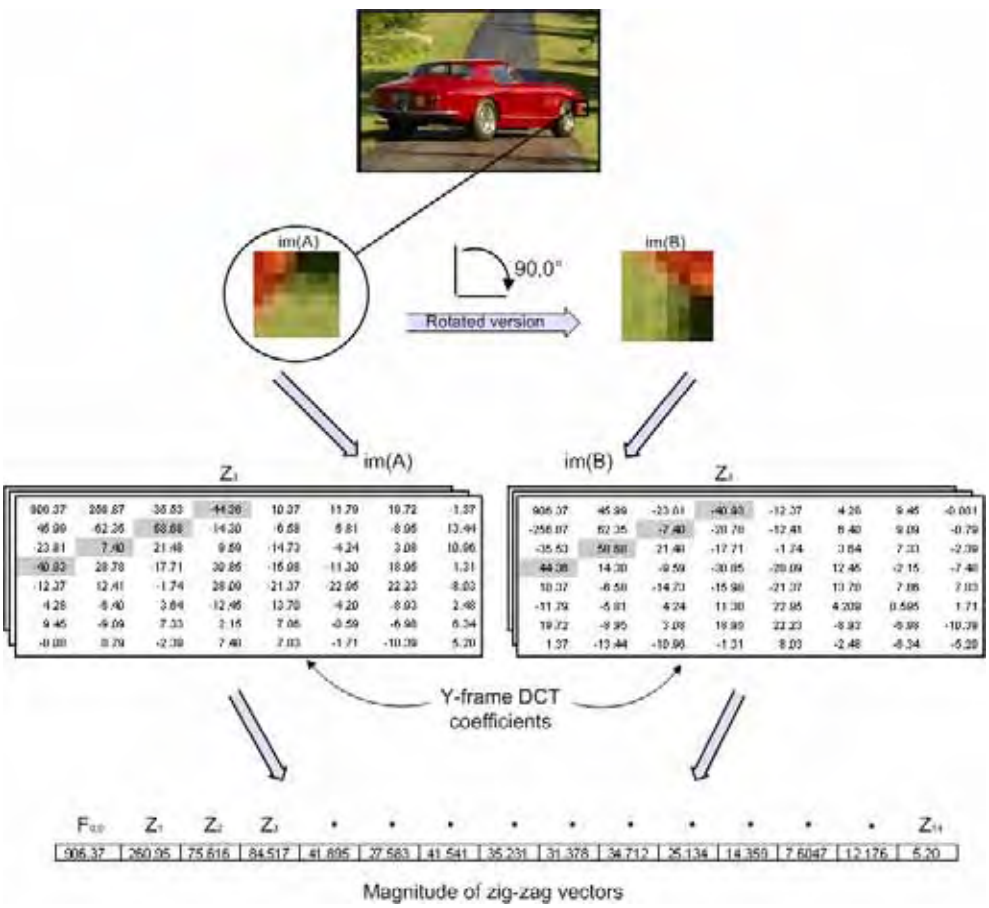


Fig. 6. DCT coefficients in the case of an image macroblock im(A) and its rotated version im(B). Each diagonal line of the zig-zag scheme is considered as a vector. The corresponding vectors (gray-shaded) contain AC coefficients having equal absolute value, although they are located at different positions. However, their magnitudes are the same, as provided at the bottom panel.

In this section, an efficient indexing method is outlined (Theoharatos et al., 2006b). Primarily, $k-$vectors are extracted from the diagonal zig-zag coefficients of each block, where a vector is defined by the AC components contained inside each diagonal line of the zig-zag scheme. The $k-$magnitudes V_k, $k=1,2,\ldots,2N-2$ of the corresponding zig-zag vectors are computed in the sequel, from Z_1 to Z_k (in the case of 8×8 image block, $k=14$ as presented in Fig. 6). This representation has been proven to be robust to image geometric transformations. That is, by applying the DCT transform to an image block and its rotated version, the set of the absolute values of the DCT coefficients is identical, whereas their positions in the zig-zag ordering scheme are different (Theoharatos et al., 2006b). This obvious advantage is illustrated in the example of Fig. 6, where an image block of size 8×8 is extracted along with its 90° right-rotated version and labeled *im(A)* and *im(B)* respectively. By applying the DCT transform to both initial and rotated image block, the set of the absolute values of the DCT coefficients is identical, but their positions are different in the zig-zag ordering scheme (depicted by the shaded lines in both matrices). Estimating the simple Euclidean distance between the corresponding zig-zag vectors of *im(A)* and *im(B)* using the proposed methodology, it was apparently found to be zero.

A critical issue that has to be solved is the number of selected AC components that need to be extracted from each image block, so as to represent effectively and efficiently the color and texture attributes. Owing to the very nature of the DCT, the set of AC coefficients generated for each $N\times N$ block are considered approximately uncorrelated. For an $N\times N$ pixel-block, the general intention is to choose those $M-$features out of the total number of N^2 DCT coefficients (except from the DC component that is always chosen as color attribute) that capture most of the spectral energy, while in our case, to select $k-$vectors out of the $2N-1$ ones estimated inside an image block. The number of selected texture and color-texture features must be extracted separately from each image channel. By testing with several JPEG images and using standard statistical methods (Duda et al., 2001) such as entropy estimation, the number of extracted zig-zag vectors was approximately found to be $k=3$, therefore using the first three zig-zag vectors. It should be noticed here that the extraction of color and texture features (i.e. the DC component and the $k-$zig-zag vectors) from each chromatic frame, increases the dimensionality of the derived feature space. However the computational complexity is not increased due to the fact that the WW-test is a function of the number of input vectors and not of their dimensions. On the other hand, the similarity measure is optimized by the higher number of extracted image features. Additionally, the optimal number of extracted vectors from the Y- frame was experimentally found to be 8, increasing the dimensionality of the feature space to 16 (8-dimensions for the luminance Y-frame and 4-dimensions for each of the two chrominance channels).

The performance of the proposed indexing scheme was evaluated on the same query-by-example retrieval system, using the WW-test as the similarity measure and following the standard procedure of constructing the Precision vs. Recall diagram. The Pr- and Re-indices were first evaluated for different sizes *T* of the selected list ($T=5:5:50$) and the computed values were used in the plot of Fig. 7. The corresponding curves for the other techniques presented earlier have also been included in the same figure. In all curves, the same number of color and color-textures features has been extracted from each image macroblock, enabling the direct comparison of the different approaches. As we can perceive, the

proposed methodology outperforms all other techniques, having in all cases of the selected list T of retrieved images significantly higher precision rate.

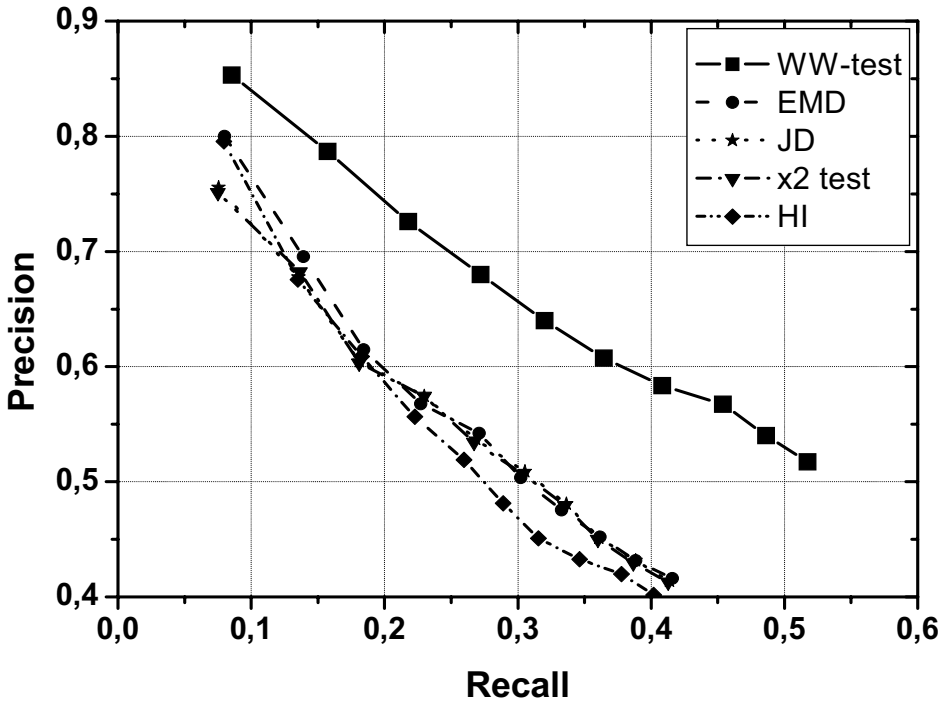


Fig. 7. Precision vs. Recall diagrams for the proposed compressed-domain retrieval scheme, in comparison to other related techniques also applied using the same indexing scheme in the compressed domain.

6. Conclusions

An intelligent strategy to visual information similarity is introduced based on the use of the nonparametric multivariate Wald-Wolfowitz statistical test. Our approach relies on a dual segregation-integration algorithmic step. The set of low-level characteristics is extracted in the form of an ensemble of feature-vectors and then 'set-differences' are computed between pairs of image representations. The new method is built on firm mathematical concepts, providing us with all the practical advantages of employing a suitable distributional distance. Its intelligent character stems from the fact that by altering the involved visual information, we can modify the flavour of formulated queries. By measuring the performance of the proposed distributional measure using some pre-defined feature extraction procedures, we show that it outperforms previously related ones that are considered as classical approaches for image similarity. The suggested methodology is evaluated within a query-by-example image retrieval scheme.

The only seemingly weak point of the proposed scheme is that it relies on the formation of MST, which is known to be a computationally demanding procedure. To provide some

insight about the complexity, the MST construction requires computational time $O(N^2)$ using a standard algorithm (Laskaris & Ioannides, 2001), while the test statistic can be evaluated in time $O(N)$, where N is the number of involved data points. The selection of a small number of input feature vectors can alleviate the computational load of the WW-engine and is fully justified by the presented experimental results. These results show that our method reaches the maximum performance for moderate size of visual attributes and therefore a more detailed distributional representation is unnecessary (Theoharatos et al., 2006a; Theoharatos et al., 2006b). Apart from this experimental fact, it should be noticed that, nowadays, the theory of randomized algorithms (Motwani & Raghavan, 2000) provides alternative fast approximations to the MST construction problem. Using such algorithms, the efficiency of the presented method might improve further.

7. Future trends

Future research remains to examine different/advanced representations of image content so as to be embedded in the WW-engine. For instance, blob representations of images emerging from a context-dependent segmentation algorithmic procedure could be incorporated in the retrieval scheme, also done within the EMD-framework (Greenspan et al., 2004). In this way, we will able to compare images that are considered to be semantically more relevant and which require the identification of specific types of objects and scenes. This can be accomplished by modifying the visual attribute-extraction process from that of *primitive* features (such as color, texture, shape or spatial location of image elements) to that of *logical* features (such as the identity of the objects depicted in an image). The most appealing and simultaneously straightforward adjustment is definitely the engagement of the recently proposed neuromorphic training scheme (Laskaris & Fotopoulos, 2004) that leads to image content representations that are highly relevant to human visual perception.
The problem of modelling image semantics needs to be systematically examined, so as to be incorporated in the standard WW-framework. In this way, techniques that capture the semantic meaning of images have to be studied for perceptual categorization and WW-based similarity of color images, using low-level descriptors derived from high-level semantic primitives. Recent research focuses on implementing perceptually motivated feature extraction algorithms into real-working environments. In their work, Mojsilovic and Rogowitz (2004) performed several subjective experiments in order to understand important semantic categories that drive our visual perception and, afterwards, extracted meaningful low-level descriptors from these semantic categories in order to perceptually characterize the database images. By integrating these features into our WW-engine, enhanced retrieval results and better organization of image databases can be achieved (Theoharatos et al., 2007). Finally, other intelligent methodologies (Eakins, 2002) can be directly adopted in our system in order to improve the matching process and also provide the significance level of perceptual image similarity using semantically relevant visual attributes.

8. Acknowledgements

This work was financed by the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program “New graduate programs of University of Patras”.

9. References

Castelli, V. & Bergman, L.D. (2002). *Image databases: Search and retrieval of digital imagery,* New York: John Wiley & Sons, ISBN 0-471-32116-8, USA.

Duda, R.O.; Hart, P.E. & Stork, D.G. (2001). *Pattern Classification*, 2nd Edition, New York: John Wiley & Sons, ISBN 0-471-05669-3, USA.

Eakins, J.P. (2002). Towards intelligent image retrieval. *Pattern Recognition,* Vol. 35, No. 11, pp. 3-14.

Friedman, J.H. & Rafsky, L.C. (1979). Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistics,* Vol. 7, No. 4, pp. 697-717.

Gdalyahu, Y.; Weinshall, D. & Werman, M. (2001). Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 23, No. 10, pp. 1053-1074.

Gray, R.M. (1984). Vector quantization. *IEEE ASSP Magazine,* Vol. 1, No. 2, pp. 4-29.

Greenspan, H.; Dvir, G. & Rubner, Y. (2004). Context-dependent segmentation and matching in image databases. *Computer Vision and Image Understanding,* Vol. 93, No. 1, pp. 86-109.

Haykin, S. (1999). *Neural networks – a comprehensive foundation,* 2nd Edition, Canada: Prentice Hall Inc., ISBN 0-13-908385-5, USA.

Jiang, J.; Armstrong, A. & Feng, G.C. (2004). Web-based image indexing and retrieval in JPEG compressed domain. *Multimedia Systems,* Vol. 9, No. 5, pp. 424-432.

Kohonen, T. (2001). *Self-organizing maps,* 3nd Edition, Berlin: Springer-Verlag, ISBN 3-540-67921-9, USA.

Laskaris, N.A. & Ioannides, A. (2001). Exploratory data analysis of evoked response single trials based on minimal spanning tree. *Clinical Neurophysiology,* Vol. 112, pp. 698-712.

Laskaris, N.A. & Fotopoulos, S. (2004). A novel training scheme for neural-network based vector quantizers and its application in image compression. *Neurocomputing*, Vol. 61, pp. 421-427.

Leow, W.K. & Li, R. (2004). The analysis and applications of adaptive-binning color histogram. *Computer Vision and Image Understanding,* Vol. 94, No. 1-3, pp. 67-91.

Martinez, T.M.; Berkovich, S.G. & Schulten, K.J. (1993). "Neural-Gas" network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks,* Vol. 4, No. 4, pp. 558-569.

Martinez, T.M. & Schulten, K.J. (1994). Topology representing networks, *Neural Networks,* Vol. 7, No. 3, pp. 507-522.

Mojsilovic, A. & Rogowitz, B.E. (2004). Semantic metric for image library exploration. *IEEE Trans. on Multimedia,* Vol. 6, No. 6, pp. 828-838.

Motwani, R. & Raghavan, P. (1995). *Randomized Algorithms,* U.K.: Cambridge University Press, ISBN 0-521-47465-5, USA.

Ngo, C.-W.; Pong, T.-C. & Chin, R.T. (2001). Exploiting image indexing techniques in DCT domain. *Pattern Recognition,* Vol. 34, No. 9, pp. 1841-1851.

Paschos, G.; Radev, I. & Prabakar, N. (2003). Image Content-based Retrieval using Chromaticity Moments. *IEEE Trans. on Knowledge and Data Engineering,* Vol. 15, No. 5, pp. 1069-1072.

Prim, R.C. (1957) Shortest connection networks and some generalizations. *Bell Sys Tech J*, Vol. 36, No. 6, pp. 1389-1401.

Rubner, Y.; Tomasi, C. & Guibas, L.J. (2000). The earth mover's distance as a metric for image retrieval. *Int. Journal of Computer Vision,* Vol. 40, No. 2, pp. 99-121.

Rubner, Y.; Puzicha, J.; Tomasi, C. & Buhmann, J.M. (2001). Empirical evaluation of dissimilarity measures for color and texture, *Computer Vision and Image Understanding,* Vol. 84, No. 1, pp. 25-43.

Schettini, R.; Ciocca, G. & Zuffi, S. (2001). A Survey on methods for colour image indexing and retrieval in image databases. In: *Color Imaging Science: Exploiting Digital Media*, Luo, R.; MacDonald, L. (Ed.), New York: John Wiley & Sons.

Sheikholeslami, G.; Chang, W. & Zhang, A. (2002). SemQuery: Semantic clustering and querying on heterogeneous features for visual data. *IEEE Trans. on Knowledge and Data Engineering,* Vol. 14, No. 5, pp. 988-1002.

Smeulders, A.W.M.; Worring, M.; Santini, S.; Gupta, A. & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 22, No. 12, pp. 1349-1380.

Stricker, M. & Orengo, M. (1995). Similarity of Color Images, *Proceedings of SPIE Storage and Retrieval for Image and Video Databases III*, pp. 381-392, San Jose, CA, USA, 9 Feb. 1995.

Swain, M.J. & Ballard, D.H. (1991). Color indexing. *Int. J. Computer Vision,* Vol. 7, No. 1, pp. 11-32.

Theoharatos, Ch.; Economou, G. & Fotopoulos, S. (2005). Color edge detection using the minimal spanning tree. *Pattern Recognition,* Vol. 38, No. 4, pp. 603-606.

Theoharatos, Ch.; Laskaris, N.A.; Economou, G. & Fotopoulos, S. (2005). A generic scheme for color image retrieval based on the multivariate Wald-Wolfowitz test. *IEEE Trans. Knowledge and Data Engineering,* Vol. 17, No. 6, pp. 808-819.

Theoharatos, Ch.; Laskaris, N.A.; Economou, G. & Fotopoulos, S. (2006a). Combining self-organizing neural nets with multivariate statistics for efficient color image retrieval. *Computer Vision and Image Understanding,* Vol. 102, No. 13, pp. 250-258.

Theoharatos, Ch.; Pothos, V.K.; Laskaris, N.A.; Economou, G. & Fotopoulos, S. (2006b). Multivariate image similarity in the compressed domain using statistical graph matching. *Pattern Recognition, Special Issue on Similarity-based Pattern Recognition,* Vol. 39, No. 10, pp. 1892-1904.

Theoharatos, Ch.; Laskaris, N.A.; Economou, G. & Fotopoulos, S. (2007). On the perceptual organization of image databases using cognitive discriminative biplots. *EURASHIP Journal on Applied Signal Processing, Special Issue on Image Perception*, Volume 2007, Article ID 68165, 15 pages, in press.

Wallace, G.K. (1991). The JPEG still picture compression standard. *Communication ACM,* Vol. 34, No. 4, pp. 31-45.

Zahn, C.T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. on Computers,* Vol. C-20, No. 1, pp. 68-86.

Zhong, Y. & Jain, A.K. (2000). Object localization using color, texture and shape. *Pattern Recognition,* Vol. 33, No. 4, pp. 671-684.

18

The Theory of Edge Detection and Low-level Vision in Retrospect

Kuntal Ghosh [1], Sandip Sarkar [2] and Kamales Bhaumik [3]
[1]*Indian Statistical Institute,*
[2]*Saha Institute of Nuclear Physics,*
[3]*West Bengal University of Technology*
India.

1. Introduction

Talking about human eye and how its astounding complexity seemingly challenged the very laws of evolution, Charles Darwin observed the following while discussing about "organs of extreme perfection and complication", in the Chapter VI titled Difficulties of the Theory of his revolutionary work, *The Origin of Species*:

"To suppose that the eye with all its inimitable contrivances for adjusting the focus to different distances, for admitting different amounts of light, and for the correction of spherical and chromatic aberration, could have been formed by natural selection, seems, I freely confess, absurd in the highest degree. When it was first said that the sun stood still and the world turned round, the common sense of mankind declared the doctrine false; but the old saying of Vox populi, vox Dei, as every philosopher knows, cannot be trusted in science. Reason tells me, that if numerous gradations from a simple and imperfect eye to one complex and perfect can be shown to exist, each grade being useful to its possessor, as is certainly the case; if further, the eye ever varies and the variations be inherited, as is likewise certainly the case and if such variations should be useful to any animal under changing conditions of life, then the difficulty of believing that a perfect and complex eye could be formed by natural selection, though insuperable by our imagination, should not be considered as subversive of the theory."

The purpose of the present chapter would be to understand and explain some of the aspects of this highly complicated organ and how it is likely to coordinate with the brain at the stage of early vision. Pioneering contributions in this domain came from renowned philosophers and vision scientists like Wilhelm Wundt, Hermann von Helmholtz (Helmholtz, 1867) and Ernst Mach (Mach, 1865). The British empiricist school of Locke, Hume and Berkeley led to the structuralist viewpoint of Wundt and the empirio-critical view of Mach, that defined visual perception as a *process* arising out of certain basic sensory atoms which act as primitive, indivisible elements of visual experience spanning each tiny localized region of the visual field, presumably resulting from the activity of the individual rods and cones in the retina. Analogous to the structural relation between primitive atoms and the more complex molecules, this structuralist theory relied upon the concept of gluing together of

many simple sensations (like colour) into more complex perceptions of a whole entity.

As a reaction to such mechanical materialist viewpoint arose the Gestalt movement that was led by Max Wertheimer who in the guise of rejecting the structuralist viewpoint, actually attacked the very base of scientific materialist viewpoint by claiming that perceptions can only have their own intrinsic whole structures that cannot, by any means, be reduced to parts or even to piecewise relations among the parts. As evidence of holism, Gestaltists pointed to those examples in which configurations have emergent properties, not shared by any of their local parts. Thus, while the structuralist viewpoint represented an inconsistent materialistic approach where "part" assumes the role of almighty and the "whole" is merely its follower, the Gestalt school on the other resorted to idealism where "part" is devoid of any identity with respect to "whole". The dialectical relation between part and whole − that is the science of transformation of quantity to quality, which is responsible for any emergent behaviour was temporarily dissolved in the fog of subjectivism, until the time was ripe for the advancement of science and philosophy to free the domain of vision science from such cloaks of mysticism. Emerged a new school of vision scientists to whom vision is first and foremost, an information-processing task whose study should invariably include not just how to extract from images the various aspects of the world that are useful to us, but also an enquiry into the nature of internal *representations* by which we capture this information and make it available for *processing* as a basis for decisions about our thoughts and actions. The use of computer simulations to model the cognitive processes, the application of information processing approach to psychology and the rapid advancement in neurophysiological techniques that led to the emergence of the idea that the eye-brain system is a biological processor of information, changed the way in which scientists understood vision. The remarkable works of Golgi, Cajal, Adrian, Granit, Hartline and other physiologists along with the advent of the modern computer age led by Alan Turing and John von Neumann served to establish the fact that starting from the two dimensional intensity array formation on the retina to the three dimensional object reconstruction and recognition in higher regions of the brain, the entire process is controlled and executed by networks of neurons of different types and that there is no "soul" sitting anywhere and interpreting things from the neuronal outputs. Rather visual perception is a collective, step-by-step synchronization of the outputs at various stages in the eye and the brain, no matter how complex that process is. It was this approach that led to the notion of a cell's "receptive field" that becomes evident so clearly from the study by H. B. Barlow of the ganglion cells of the frog retina where he said (Barlow, 1953):

"If one explores the responsiveness of single ganglion cells in the frog's retina using handheld targets, one finds that one particular type of ganglion cell is most effectively driven by something like a black disc subtending a degree or so moved rapidly to and fro within the unit's receptive field. "

The corresponding mathematical approach of creating computer programs to extract useful information about the environment from optical images was articulated most effectively by David Marr and his colleagues (Marr, 1982). It dealt in details with how the luminance structure in two-dimensional images may provide information about the structure of surfaces and objects in three-dimensional space, though the pioneering mathematical analysis in this field was contributed by the Dutch physicists Jan Koenderink and Andrea van Doorn who dealt with sophisticated mathematical techniques from differential

geometry to the three dimensional orientation of surfaces from shading information. But in this chapter we shall restrict ourselves only to the receptive field structure relevant to the Theory of edge detection (Marr & Hildreth, 1980) that, according to its authors, is responsible for a "raw primal sketch" of the world around us. For this, we first elucidate a few basic things associated with the processing of the digital images by computers, which would be extensively used in the present chapter.

2. Preliminary Concepts in Computer Vision

An image is a two-dimensional representation of a three dimensional object or scenario. A monochrome image is characterised by a continuous intensity function $I(x,y)$ at every point (x,y) in the image plane. The final goal of image processing is to extract information from $I(x,y)$ to reconstruct the 3-D view of the original object or scenario. In a digital image the abstract concept of points is replaced by a realistic concept of infinitesimal identical areas (such as pixels in a computer screen). These infinitesimal areas span the entire image plane and are numbered in an ordered fashion both horizontally and vertically. Moreover, the continuous intensity function is replaced by values from a discrete gray scale. As a result the continuous intensity function $I(x,y)$ is replaced by a discrete function $I(x_i,y_i)$, in which (x_i,y_i) denotes the pixel position and $I(x_i,y_i)$ denotes the average discrete gray scale value of that pixel.

Let us now discuss the salient points about the concept of an edge. Location of an edge is the most crucial information that is to be extracted during the primary processing of any image. Any sharp change of intensity qualifies for an edge (Fig. 1a). Accurate detection of these transitions along with their correct locations is the purpose of edge detection algorithms. In a digital image an edge occurs at the boundary between two pixels provided the gray values of the pixels differ considerably from one another. From the vagueness of the word "considerably" it is obvious that identification of an edge is a subjective procedure. In one extreme any difference of intensity may be assumed to be an edge, so that the processed image would become a messy assemblage of edges leaving no scope for feature extraction. In its other extreme, the important edges may get lost thereby forsaking valuable information. Sudden transition of a continuous function is best identified by differentiating it, which gives a large value at the point of transition and zero value at the points of no transition. For a discrete function, the differentiation operation is replaced by difference operation. One can use either first order directional derivatives, like $\partial/\partial x$ or $\partial/\partial y$ in which case one would have to search for their crests and troughs at each orientation (Fig. 1b) when applied to a 2-D image, or one can also use second order directional derivatives, like $\partial^2/\partial x^2$ or $\partial^2/\partial y^2$ in which case the directional intensity change would correspond to their zero-crossings (Fig. 1c). Using finite difference approximation, the corresponding spatial organizations for some these operators or "receptive fields" as they are neurophysiologically termed, are displayed below:

$\partial/\partial x \equiv$

-1	+1

$\partial^2/\partial x^2 \equiv$

-1	+2	-1

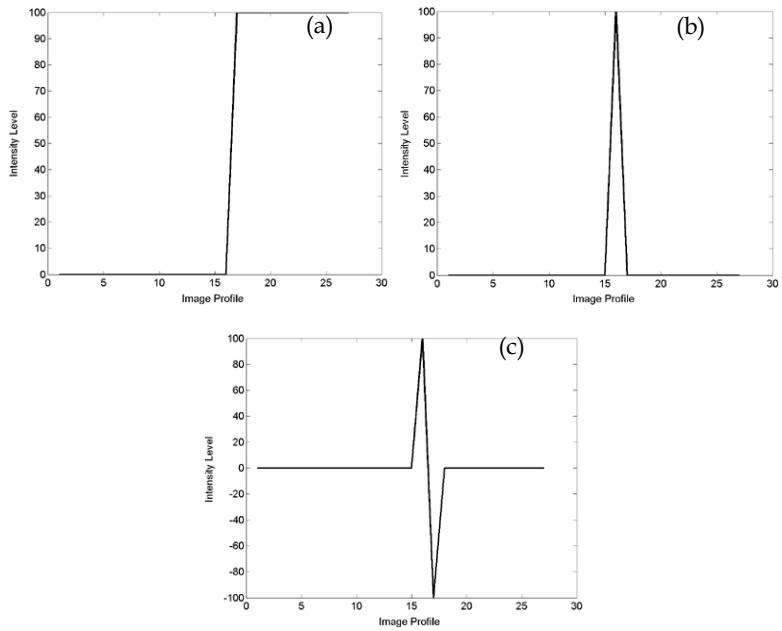


Fig. 1. (a) A function showing simple one-dimensional step edge. (b) First order derivative of a step edge showing zero value at all points except the transition point. (c) Second order derivative of a step edge. It is to be noted that the location of the zero crossing faithfully reproduces the location of the edge.

These are also called masks and all the operations can be performed on a digital image by convolving $I(x_i, y_i)$ with such a mask. Convolution of a digital image with a finite mask is the process of converting the gray value of each of its pixel with weighted sum of the gray values of the pixels in its neighbourhood. In this way a one dimensional image corresponding to the step function shown in Fig. 1a has been convoluted with the two masks shown above and the intensity distributions of the convoluted images have been shown in Fig. 1b and 1c. However, a major disadvantage of these operators is that, they are all directional. Thus in order to use first order derivatives, both $\partial I/\partial x$ and $\partial I/\partial y$ have to be computed, where I represents the intensity distribution of the image. Then the crests and troughs in the overall amplitude have to be found i.e. $\left[(\partial I/\partial x)^2 + (\partial I/\partial y)^2\right]^{\frac{1}{2}}$ must also be computed. Using second order directional derivative operators will lead to similar and worse problems. The only way to avoid these extra computational burdens is to choose an isotropic differential operator and such an operator of the lowest order happens to be the Laplacian (∇^2). It is also interesting to note at this point that a role of the same operator in visual perception was suggested by Ernst Mach (Mach, 1865). Mach relied upon psychophysical observations to arrive at this conclusion empirically. This we shall explain in section 4. Presently we shall discuss the role of Gaussian blurring in the edge detection problem.

Edge detection being a problem of numerical differentiation, is a weakly ill-posed problem since every realistic image is contaminated by some noise and these small variations in

input lead to large changes in output. Since a noise point has a likelihood of having an intensity difference with its neighbours, in edge analysis this may create spurious edge points. It is, therefore, desirable that before processing the image, the intensity of a noise point should be brought closer to the intensity of its neighbourhood. Any filter operated over the image to achieve such a smoothing should make the spatial variation of intensity as small as possible or in other words the spatial variance Δx of the filter should be small. On the other hand, the filter's spectrum should be band-limited in the frequency domain. Consequently its variance $\Delta\omega$ should also be small. There is a conflict between these two localisations through an uncertainty principle: $\Delta x \Delta\omega \geq \frac{\pi}{4}$. The only function that optimises this relation is the Gaussian function. This is the reason why the images are generally smoothened by convolving with a Gaussian function prior to the differentiation operation.
A one-dimensional Gaussian function is defined as:

$$G(x,\sigma) = \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad \text{so that} \quad \int G(x,\sigma)dx = 1 \tag{1}$$

Here σ is the standard deviation (or scale parameter) of the Gaussian function. Convolution of an image with the Gaussian function effectively wipes out all structures at scales smaller than the space constant σ of the Gaussian function. It may easily be verified that the Fourier transform of a Gaussian function is also a Gaussian.
In 2-D, the Gaussian is defined as:

$$G(r,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \quad \text{with} \quad r^2 = x^2 + y^2 \tag{2}$$

For an image, the Gaussian filtering has the added advantages. Since a 2-D Gaussian function is rotationally symmetric, it preserves the neighbourhood characteristics both in the spatial and frequency domain. It is also computationally handy because it can be decomposed into two 1-D Gaussians i.e. $G(x,y) = G(x)G(y)$. In fact Gaussian is the only rotationally symmetric function that is separable. The two types of filters, discussed above viz. the derivative operator and the smoothing operator, are both used extensively in digital image processing. In effect, initially the unwanted noises are to be removed (smoothened) from the image by convoluting it with a Gaussian function. Then a derivative filter is operated to detect the edge points. From the discussion presented above, it is becoming increasingly clear why in their classical theory of edge detection (Marr & Hildreth, 1980), the authors argued in favour of the Laplacian of Gaussian ($\nabla^2 G$) based structure of receptive field. But before we deal with this operator in more detail, it is first important to look into the mechanism of image processing in mammalian eye and what the receptive field is.

3. A Brief Overview of Mammalian Retina

It is known from neurophysiological experiments on cat and monkey that a good deal of processing of images falling on the eye occurs in the retina and primary visual cortex itself. This is known as primary or low-level visual processing. We shall now give a brief overview of the physiology related to primary visual processing and its role in edge detection.

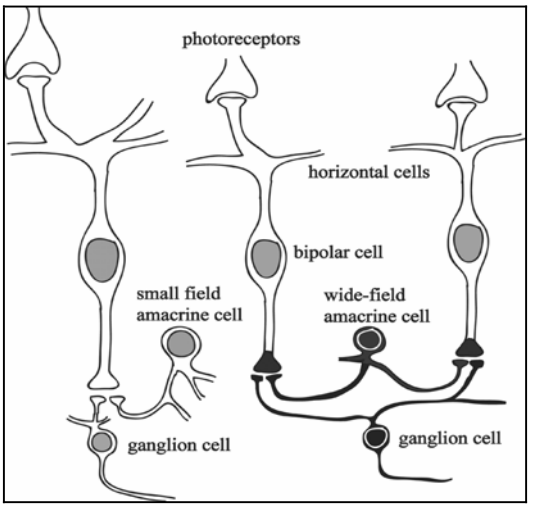


Fig. 2 A schematic drawing showing the retinal network. Rods and cones, known as photoreceptors, receive the light, get excited, send information to the bipolar cells, either directly or through the network of horizontal cells. The bipolar cells, in their turn, send information to ganglion cells, again either directly or through the network of amacrine cells. From ganglion cells the information travels to the primary visual cortex through optic nerves.

In the mammalian retina, the primary photoreceptors are the rods and cones (Fig. 2), which are spread over a surface. For simplicity if we neglect the aspect of colour, the retinal images can be approximated by $I(x_i, y_i)$ as argued in the previous section. Classical investigations by neurophysiologists have shown that information about the input image is extracted in the successive layers of the retina (Fig. 3a). For example, a bipolar cell receives information from a large number of photoreceptors distributed over a circular zone, mainly through a network of horizontal cells and the ganglion cell receives information from the bipolar cells through another network of amacrine cells. It is easy to understand that any particular bipolar or ganglion cell cannot receive information from all the photoreceptors (rods and cones) of the entire retina. Only a small area of the retina would be responsible for eliciting response in that cell. That area (assumed to be circular or elliptical in shape) is called the receptive field of that bipolar or ganglion cell. A schematic diagram is shown in Fig. 3b. Physiologists further observed that while the receptors in the central region of this zone send information to a bipolar cell in a positive fashion, the information from the peripheral cells arrives with a reversal of signature (Fig. 4). As a result a central bright spot with dark background is the best stimulus for exciting the bipolar cell. (These bipolar cells are known as on-centre cells. There are also off-centre bipolar cells for which a dark spot with bright background is the most appropriate stimulus.) Information of such an antagonistic effect from a large number of bipolar cells is collected and transmitted by the ganglion cells.

For simplicity in understanding the organization of a receptive field structure, let us consider a one-dimensional retina in which the photoreceptors are spread over a line. Strength of the output from a photoreceptor to the ganglion cell should be maximum when the two cells are in closest proximity. It is also natural to assume that the contributions

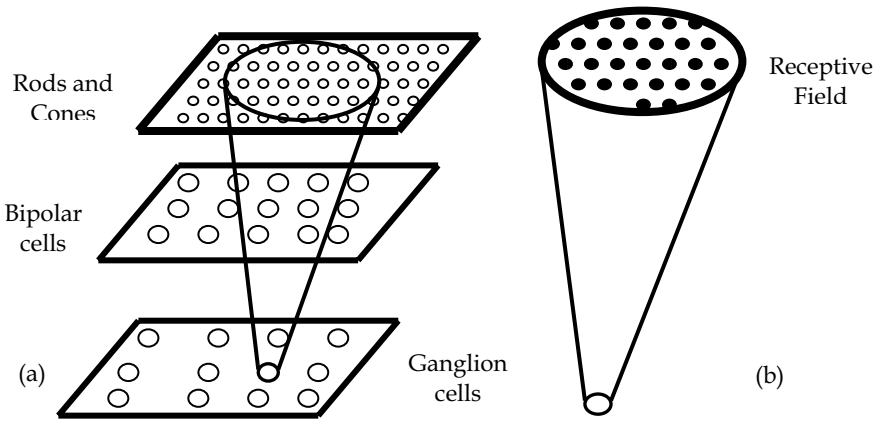


Fig. 3 (a) Information processing occurs in the retina through successive layers. (b) Receptive field of a bipolar or ganglion cell is a circular or elliptical area on the photoreceptor layer that elicit response in that cell

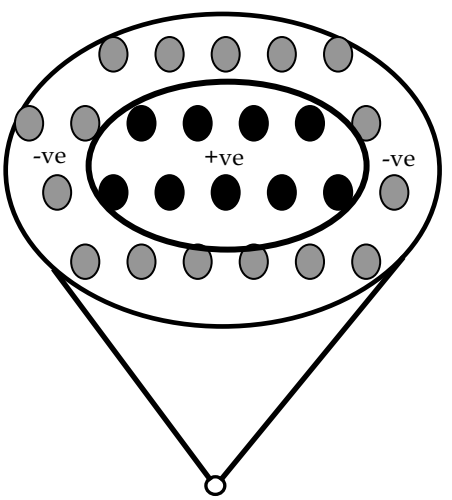


Fig. 4 The classical excitatory-inhibitory centre-surround receptive field structure of retinal bipolar and ganglion cells.

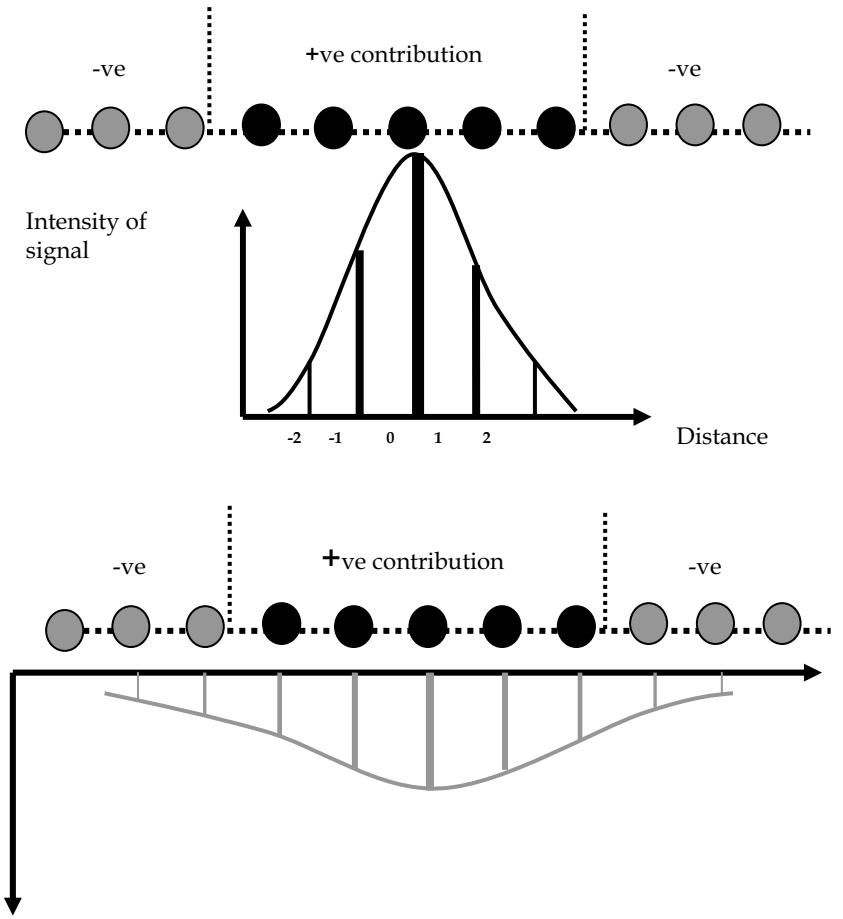


Fig. 5. The centre and surround responses of a ganglion cell has been fitted with two Gaussian curves in opposite phase. The surround is represented by a broader Gaussian compared to the central one

received by a ganglion cell from other receptors will smoothly fall off with the distance. Such a distribution can be safely assumed to be a Gaussian. This would be true for both positive (centre) and negative (surround) inputs (Fig. 5). Consequently the net input to a ganglion cell is obtained from a difference of two Gaussian inputs, the central one (positive) having a smaller variance than the surround (negative). This prompted the physiologists to develop a model of Difference of Gaussian or DOG for the receptive field of retinal ganglion cells. A DOG function in one dimension, will be:

$$DOG(\sigma_1,\sigma_2) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x^2}{2\sigma_1^2}} - \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{x^2}{2\sigma_2^2}} \tag{3}$$

This model can be easily extended for two-dimensional images by using 2-D Gaussians. The DOG model is very effective in explaining a large number of experimental findings in retinal responses as we shall see in sections 4 and 5.3. Essentially DOG is the classical model for the centre-surround antagonistic effects observed at the retinal ganglion cell.

4. The Classical Receptive Field and Theory of Edge Detection

As discussed previously, from the computational point of view the most natural filter for edge detection should have a combination of derivative and smoothening filter. As established before, a Laplacian of Gaussian (LOG) filter is the best alternative for combining the smoothening and derivative operation for the image. Laplacian operated on a 2-D Gaussian will give:

$$\nabla^2 G(r,\sigma) = -\frac{1}{\pi\sigma^2}\left[1 - \frac{r^2}{2\sigma^2}\right] e^{-\frac{r^2}{2\sigma^2}} \tag{4}$$

Marr and Hildreth (Marr & Hildreth, 1980) further argued that for a certain ratio of the scale parameters in DOG (i.e. for a certain value of $\sigma_1 : \sigma_2$), LOG can be considered to be a good approximation to DOG. We have already said that even without any knowledge of the DOG based classical receptive field structure from physiologists, since those experiments were actually performed almost a century after he carried out his psychophysical experiments, Ernst Mach, could still visualize empirically the centre-surround structure in retina and predict the Laplacian operation in early vision as well. This is what Mach said (Mach, 1865):
"The illumination of a retinal point will, in proportion to the difference between this illumination and the average of the illumination on neighboring points, appear brighter or darker, respectively depending on whether the illumination of it is above or below the average. The weight of the retinal points in this average is to be thought of as rapidly decreasing with distance from the particular point considered."
Furthermore, he went on to state:
"Let us call the intensity of illumination $u = f(x, y)$. The brightness sensation v of the corresponding retinal point is given by

$$v = u - m(d^2u/dx^2 + d^2u/dy^2) \tag{5}$$

where m is a constant. If the expression in parentheses is positive, then the sensation of brightness is reduced; in the opposite case, it is increased. Thus, v is not only influenced by u, but also its second differential coefficients."
Let us now see, what led Mach to arrive at such revolutionary conclusions on visual perception. Mach was experimenting with rotating white discs with black sectors of varying size, when he came across the phenomenon that is now commonly referred to as Mach band illusion. The most commonly used image for understanding the Mach band illusion is shown in Fig. 6a. By scanning this image in a direction in which the luminance increases or

decreases our visual system perceives an actually non-existent darker bar at the location where the figure just starts getting lighter. Similarly, a brighter bar is perceived at the point where brightness just stops increasing. However, a horizontal line scan of this image (Fig. 6b) clearly establishes that what we see is a mere illusion and the image represents a simple staircase function only devoid of any special border effect. It was the observation of this illusory phenomenon that prompted Mach to arrive at his inferences quoted above. In order

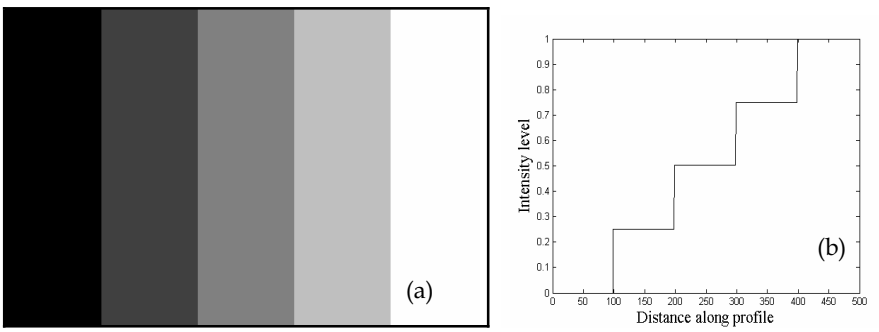


Fig. 6 (a) The Mach band illusion of dark and bright borders around bright and dark regions respectively (b) A horizontal profile of this image is obviously a simple staircase function that bears no signature of the illusory perception.

to conceive Mach's arguments let us resort to the receptive field mode of spatial organizations of the Laplacian operator as has been initiated for derivative operators in section 2. We have stated there that a finite difference approximation of the horizontal directional second order partial derivative, $\partial^2/\partial x^2$ may be written as:

$$\partial^2/\partial x^2 \equiv \begin{array}{|c|c|c|} \hline -1 & +2 & -1 \\ \hline \end{array}$$

Consequently, the vertical directional operator $\partial^2/\partial y^2$ may be represented by the transpose of the above vector. When these two are combined together, we obtain the kernel for the isotropic ∇^2 (i.e. $\partial^2/\partial x^2 + \partial^2/\partial y^2$) operator:

0	-1	0
-1	4	-1
0	-1	0

Using the property of isotropicity of the Laplacian opeartor, the diagonal directions are now incorporated by taking the co-ordinates along these directions applying a 45^0 rotation so that we arrive at a new kernel:

-1	0	-1
0	4	0
-1	0	-1

By combining the above two kernels, we get the omnidirectional operator corresponding to $\partial^2/\partial x^2 + \partial^2/\partial y^2$:

-1	-1	-1
-1	8	-1
-1	-1	-1

Convoluting any intensity array u with this operator and combining the result linearly with u as has been proposed by Mach in Equation (5), is the same as convolving u with the filter mask given below:

-1	-1	-1
-1	9	-1
-1	-1	-1

Let us now convolve the Mach band image shown in Fig. 6 with this final mask. We find that the edges at each transition have become enhanced by a mechanism where new bands have been formed clearly separating each gray level from the other (Fig. 7a). To demonstrate that this again is not mere illusion, we draw a horizontal line profile through this convolved image to find undershoots and overshoots at each step transition that bears resemblance to our illusive perception of the original image whose line scan is in contrast simply a staircase function (Fig. 6b). So we understand what prompted Mach to propose the Laplacian operation as a model for spatial filtering in the retina and as is apparent from this mask, it is essentially excitatory-inhibitory in character, which also Mach claimed. Since we have already defined image edges as sharp changes in gray levels, therefore we may conclude from these observations that any image convoluted with the omnidirectional Laplacian mask will show pronounced Mach band effect at each edge of the filtered image. In other words, the edges will all be enhanced due to the effect of such a kernel being operated on any image, since new Mach bands will be created that would serve to clearly distinguish

one gray level from another. Edge enhancement by such a mechanism has been shown in Fig. 8. The resultant images clearly show an increase in the level of sharpness compared to the original images. The reason behind such sharpening is that the bright Mach bands around dark regions and the dark ones around lighter regions, apart from being illusions, also play a crucial role in image processing. They actually represent a mechanism of lateral inhibition or the contrast-sensitivity in the eye that enables one to clearly isolate an object

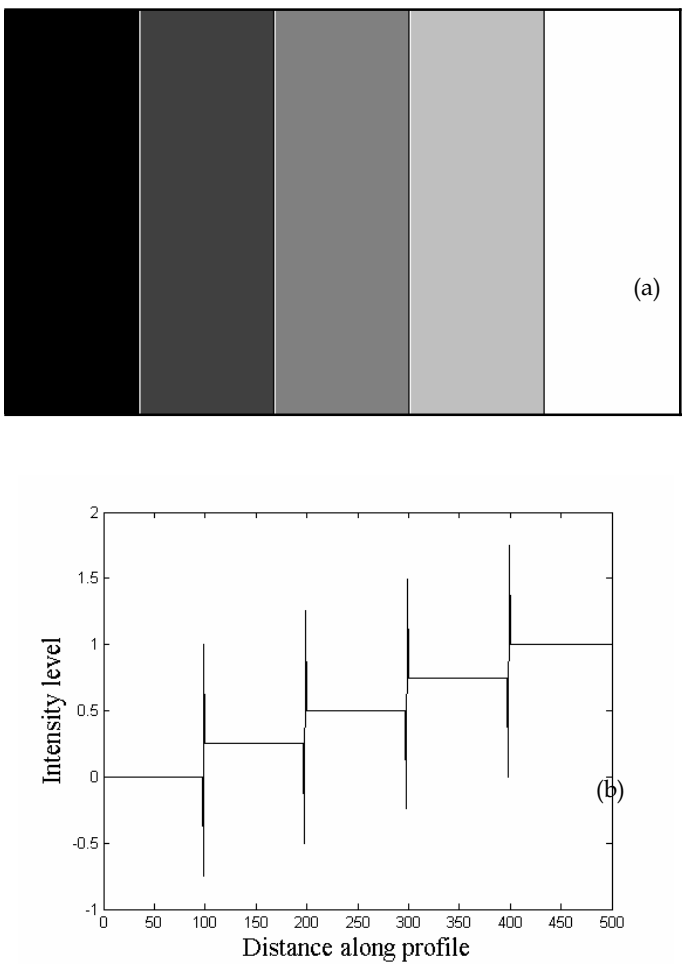


Fig. 7 (a) The effect of convoluting the Mach band image in Fig. 6a with the omnidirectional Laplacian mask clearly shows that new bands have actually been formed clearly separating each gray level from the other (b) This becomes obvious if we draw a horizontal line profile through the convoluted image, that shows the new bands as undershoots and overshoots at each step of the staircase shown in Fig. 6b. In other words a mimetic of the illusory perception of Fig. 6a, has thus been reproduced.

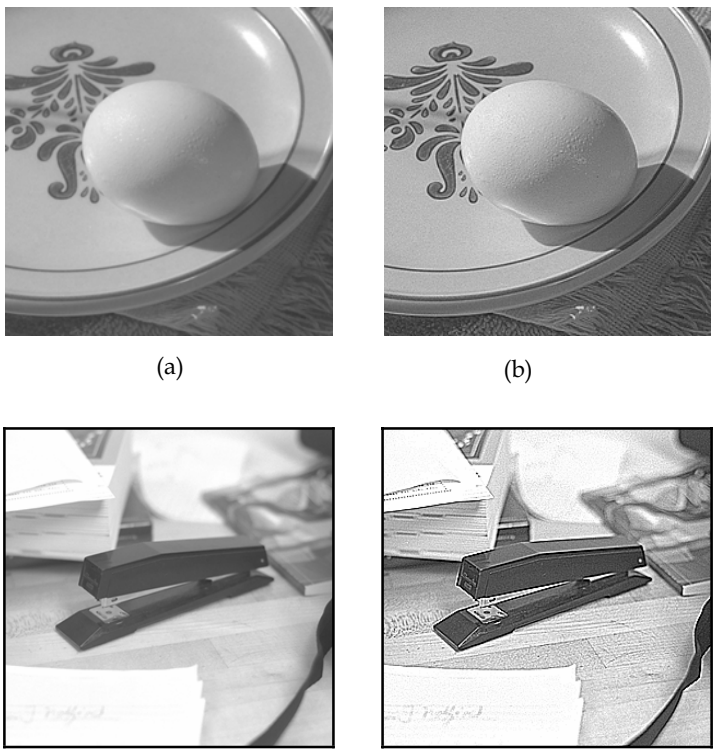


Fig. 8 Result of convoluting two bench-mark images in (a) and (c) with the omnidirectional discrete Laplacian mask has been shown in (b) and (d).

from its background, thus helping in image sharpening. As already mentioned, the polarities in the discrete mask resemble the antagonistic centre-surround receptive field structure shown in Fig. 4. Also, being an orientation independent operator, this mask naturally forms the Mach bands in all directions in an image, thus enhancing the images from objects of any arbitrary shape. What effectively gets sharpened in the process, are the edges in the images. This phenomenon, in fact, mystified Mach's viewpoint about illusion and reality, which finally led him to construct the unscientific philosophy of empirio-criticism.

5. The Non-classical Receptive Field and Low-level Vision in Retrospect

When Marr and Hildreth (Marr & Hildreth, 1980) claimed the equivalence of LoG and DoG for a particular scale ratio between the two Gaussians, they could not provide any strong theoretical basis for the equivalence. That basis was provided much later in a paper by Ma and Li (Ma & Li, 1998), wherein they proved from very general consideration that any derivative filter of a smooth function could be expressed as a linear combination of the smooth function at different scale parameters. Ma and Li have shown that any $2kth$ order

derivative filter can be designed as the weighted sum of any $(k+1)$ even functions, every function having the same kernel, but different scales. Also $(2k+1)th$ order derivative filter can be designed as the weighted sum of $(k+1)$ odd functions of different scales. In the present chapter our discussion, with respect to non-classical receptive field will be confined only within even order derivative filters because we have chosen to construct filters at different scales by using two-dimensional Gaussian function, which happens to be an even function.

But first, it would be appropriate to introduce the concept of non-classical receptive field of retinal ganglion cells. The concept of a centre-surround antagonistic receptive field of retinal ganglion cell, as we have already discussed, evolved on one hand, from Mach's earlier studies in psychophysics and on the other from the later experiments dealing with the neurophysiology of retina. The DOG or LOG models merely follow this studies. Some experimental observations, however are strongly indicative of some necessary modification to this concept of "classical receptive field". From such experiments from seventies onwards of the last century, it was observed that there are many photoreceptor cells outside the classical receptive field, that are capable of modulating the behaviour of the ganglion cells. Presently, there is practically no doubt that the actual receptive field of a ganglion cell is much widely spread than that depicted by the classical picture and that such an extended surround actually disinhibits the response of the classical receptive field. Such a non-classical receptive field containing non-linear sub-units is shown in Fig. 9, following a recent work (Passaglia et al., 2001), where it is conjectured that the mean increasing and mean decreasing units would remain either active or inactive depending on the desired task of the retina.

Although the modulation of the ganglion cells by the non-classical receptive field is probably nonlinear in nature, yet some of the effects of the non-classical receptive field may be emulated by modeling the corresponding response bahaviour simply as a linear combination of three or more zero-mean Gaussians at different scales. The narrower two of

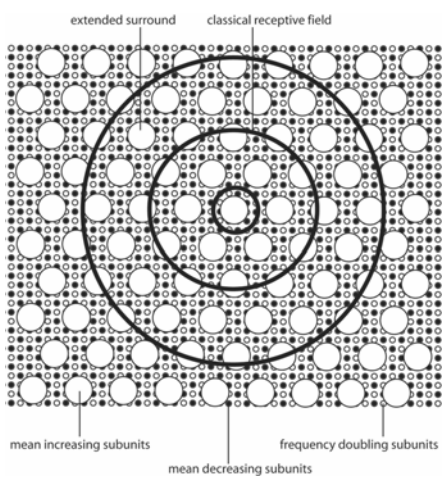


Fig. 9 The non-classical receptive field of retinal ganglion cells is characterised by an extended disinhibitory surround beyond the classical receptive field

these Gaussians may represent the classical center and the classical antagonistic surround while the non-classical extended disinhibitory surround mostly contributed by the amacrine cells in the inner plexiform layer of the retina may be represented by the wider Gaussians. Since, according to Ma and Li (Ma & Li, 1998), such a linear combination of Gaussians could be expressed as equivalent to higher order derivatives, therefore from such an argument it can been shown, that the non-classical receptive field of retinal ganglion cells can be modeled by a fourth or sixth order rotationally symmetric derivative of Gaussian, that is by $\nabla^4 G$ (the Bi-Laplacian or Bi-harmonic of Gaussian) or $\nabla^6 G$ (the tri-Laplacian of Gaussian). The detailed expressions are given in the sub-section 5.1 for the one-dimensional case, where it has also been shown that one could express $\nabla^4 G$ as $DoG + G_1$ where G_1 is the widest Gaussian representing a disinhibitory surround beyond the classical receptive field or in other words the mean-increasing sub-units in Fig. 9. Similarly, $\nabla^6 G$ can be expressed as $DoG + G_1 - G_2$, where G_2 is another wide Gaussian representing the mean-decreasing sub-units in the same figure.

5.1 A Simple Model for the Non-classical Receptive Field Structure

If the positive sub-units of the non-classical receptive field are primarily considered, then in one dimension, following Ma and Li, one can construct a fourth order derivative filter as a linear combination of three Gaussians. For this, let us define a function h_{2k} using the primitive Gaussian filter $g(x,\sigma)$ as:

$$h_{2k}(x) = \sum_{j=0}^{k} \frac{\alpha_j}{\sigma_j} g\left(\frac{x}{\sigma_j}\right), \text{ where } g\left(\frac{x}{\sigma}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{6}$$

Here α_j's are the weight functions. Ma & Li showed that h_{2k} is a $(2k)$-th order derivative filter if the α_j's satisfy the following equations :

$$\begin{aligned}
&\alpha_0 + \alpha_1 + \alpha_2 + \ldots\ldots\ldots\ldots\ldots\ldots + \alpha_k = 0 \\
&\alpha_0\sigma_0^2 + \alpha_1\sigma_1^2 + \alpha_2\sigma_2^2 + \ldots\ldots\ldots\ldots\ldots\ldots + \alpha_k\sigma_k^2 = 0 \\
&\vdots \\
&\vdots \\
&\alpha_0\sigma_0^{2k} + \alpha_1\sigma_1^{2k} + \alpha_2\sigma_2^{2k} + \ldots\ldots\ldots\ldots\ldots\ldots + \alpha_k\sigma_k^{2k} = \frac{(2k)!}{m_{g,2k}}
\end{aligned}$$

and if the matrix M_σ

$$M_\sigma = \begin{pmatrix} 1 & 1 & . & . & . & 1 \\ \sigma_0^2 & \sigma_1^2 & . & . & . & \sigma_k^2 \\ . & . & . & . & . & . \\ \sigma_0^{2k} & \sigma_1^{2k} & . & . & . & \sigma_k^{2k} \end{pmatrix} \tag{7}$$

is not singular. Here $m_{g,2k}$ is the $(2k)th$ order moment of the function $g(x)$.

Thus, for second order derivative, taking $k = 1$, one gets

$$h_2(x) = \alpha_0 \left(\frac{1}{\sigma_0} g\left(\frac{x}{\sigma_0}\right) - \frac{1}{\sigma_1} g\left(\frac{x}{\sigma_1}\right) \right) \tag{8}$$

Here α_0 is a ratio of scale parameters. For a scale ratio t, i.e. if $\sigma_1 = \sigma$ and $\sigma_0 = t\sigma$

$$h_2(x) = \alpha_0 \left(\frac{1}{(t\sigma)\sqrt{2\pi}} e^{-\frac{x^2}{2(t\sigma)^2}} - \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \tag{9}$$

Similarly, for fourth order derivative filter, let us define a function

$$h_4(x) = \alpha_0 \frac{1}{\sigma_0} g\left(\frac{x}{\sigma_0}\right) + \alpha_1 \frac{1}{\sigma_1} g\left(\frac{x}{\sigma_1}\right) + \alpha_2 \frac{1}{\sigma_2} g\left(\frac{x}{\sigma_2}\right) \tag{10}$$

where α_0, α_1 and α_2 satisfy the following equations

$$\begin{aligned} &\alpha_0 + \alpha_1 + \alpha_2 = 0 \\ &\alpha_0\sigma_0^2 + \alpha_1\sigma_1^2 + \alpha_2\sigma_2^2 = 0 \\ &\alpha_0\sigma_0^4 + \alpha_1\sigma_1^4 + \alpha_2\sigma_2^4 = \frac{4!}{m_{g,4}} \end{aligned}$$

where,

$$m_{g,4} = \int_{-\infty}^{\infty} x^4 g(x) dx \tag{11}$$

Solving these equations, we get:

$$\begin{aligned} \alpha_0 &= k\left(\sigma_2^2 - \sigma_1^2\right) \\ \alpha_1 &= -k\left(\sigma_2^2 - \sigma_0^2\right) \\ \alpha_2 &= k\left(\sigma_1^2 - \sigma_0^2\right) \end{aligned}$$

where

$$k = \frac{24}{m_{g,4}} \frac{1}{\left(\sigma_2^2 - \sigma_0^2\right)\left(\sigma_1^2 - \sigma_0^2\right)\left(\sigma_2^2 - \sigma_1^2\right)} \tag{12}$$

In this case, for the two scale ratios t and p, i.e. if $\sigma_2 = \sigma$, $\sigma_1 = t\sigma$ and $\sigma_0 = p\sigma$, then:

$$\begin{aligned} \alpha_0 &= k\sigma^2\left(1 - t^2\right) \\ \alpha_1 &= -k\sigma^2\left(1 - p^2\right) \\ \alpha_2 &= k\sigma^2\left(t^2 - p^2\right) \end{aligned} \tag{13}$$

If we take a look at the final values of the three coefficients, α_0, α_1 and α_2 as given by Equation (13), we find that a fourth order derivative filter as given in Equation (10) is

essentially a non-classical $DoG + G_1$ model as mentioned in the previous section. Moreover, experimental observations on non-classical receptive fields (Passaglia et al., 2001), indicate that the central region is much smaller than the extended surround, or in other words σ_0 is negligible in comparison to σ_2. Based on these, we can consider the ratio $\sigma_0 : \sigma_2$ to be very small and hence apply a condition $p \to 0$ in Equation (13). Then using Equation (10), we arrive at:

$$h_4(x,\sigma) \to mh_2(x,\sigma^{/}) + h_2(x,\sigma^{//}) \tag{14}$$

where, $\sigma^{/}$ and $\sigma^{//}$ are two arbitrary scales and m is an amplitude scale factor and $h_2(x)$ is given by Equation (9).

In the same way if we incorporate the negative sub-units of non-classical receptive field, then following the same procedure:

$$h_6(x) = \frac{\alpha_0}{\sigma_0} g\left(\frac{x}{\sigma_0}\right) + \frac{\alpha_1}{\sigma_1} g\left(\frac{x}{\sigma_1}\right) + \frac{\alpha_2}{\sigma_2} g\left(\frac{x}{\sigma_2}\right) + \frac{\alpha_3}{\sigma_3} g\left(\frac{x}{\sigma_3}\right) \tag{15}$$

Here

$$\begin{aligned}
\alpha_0 &= -k\sigma^6\left(t^2 - p^2\right)\left(1 - p^2\right)\left(1 - t^2\right) \\
\alpha_1 &= k\sigma^6\left(1 - r^2\right)\left(1 - t^2\right)\left(t^2 - r^2\right) \\
\alpha_2 &= -k\sigma^6\left(p^2 - r^2\right)\left(1 - r^2\right)\left(1 - p^2\right) \\
\alpha_3 &= k\sigma^6\left(t^2 - r^2\right)\left(t^2 - p^2\right)\left(p^2 - r^2\right)
\end{aligned}$$

where $$\sigma_3 = \sigma, \sigma_2 = t\sigma, \sigma_1 = p\sigma, \sigma_0 = r\sigma \tag{16}$$

If we again take a look at the final values of the four coefficients, α_0, α_1, α_2 and α_3, we find that the corresponding expression matches the $DoG + G_1 - G_2$ model described in the previous section. Then once again following the same procedure described above we assume $\sigma_0 : \sigma_3$ to be very small and apply a condition $r \to 0$ in Equation (16). Putting these values in Equation (15), after some algebraic manipulation, we finally arrive at:

$$h_6(x,\sigma) \to mh_2(x,\sigma^{/}) + nh_2(x,\sigma^{//}) + h_2(x,\sigma^{///}) \tag{17}$$

Then applying Equation (14) in Equation (17), we get:

$$h_6(x,\sigma) \to mh_2(x,\sigma^{/}) + h_4(x,\sigma^{//}) \tag{18}$$

Here $\sigma^{/}$, $\sigma^{//}, \sigma^{///}$ are all arbitrary and hence do not represent any particular scale at any stage of the derivation. So in two dimensions:

$$\nabla^6 G(r) = m\nabla^2 G(r) + \nabla^4 G(r),$$

where $$\nabla^4 G(r) = \frac{1}{2\pi\sigma^6}\left[8\left(1 - \frac{r^2}{\sigma^2}\right) + \frac{r^4}{\sigma^4}\right]\exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{19}$$

Any of these equations viz. Equation (10), (14), (15) or (19) may be considered to be our proposed model for non-classical receptive field, which means the receptive field will not be represented by LOG only whose equivalent physiological model is given by Equation (8), but rather by a linear combination of even order isotropic Gaussian derivatives. So the advantage of economy of computation that was applicable for LOG remains valid, while at the same time apart from the scale of the Gaussians, the factor m can also play a role in visual information processing at low level. To understand this more clearly we have to again resort to a corresponding receptive field like spatial organization as before for such a mathematical function and see whether it also reflects the disinhibitory extended surround in such a form of representation.

5.2 Derivation of a Kernel for the Non-classical Receptive Field

First of all we discuss on the construction of a computationally handy kernel for the ∇^4 operator following the methodology of construction of the convolution matrix for the ∇^2 operator, using finite difference approximation as discussed in section 4. Clearly,

$$\nabla^4 = \nabla^2.\nabla^2$$

$$= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right).\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$$

$$\text{So,} \quad \nabla^4 = \frac{\partial^4}{\partial x^4} + \frac{\partial^4}{\partial y^4} + 2\,\frac{\partial^2}{\partial x^2}\,\frac{\partial^2}{\partial y^2} \tag{20}$$

Utilising the finite difference approximation of the fourth order partial derivative, the kernel for $\partial^4/\partial x^4$ in discrete domain can be represented by the kernel:

$\partial^4/\partial x^4 \equiv$

1	-4	6	-4	1

By transposing this kernel we may construct the corresponding vector for $\partial^4/\partial y^4$, add these, so that we get the corresponding matrix for a linear combination of these two terms, i.e. for $\frac{\partial^4}{\partial x^4} + \frac{\partial^4}{\partial y^4}$:

0	0	1	0	0
0	0	-4	0	0
1	-4	12	-4	1
0	0	-4	0	0
0	0	1	0	0

Using the expressions for $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$ in section 4 we may arrive at a 5×5 matrix for $\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}$:

0	0	0	0	0
0	1	-2	1	0
0	-2	4	-2	0
0	1	-2	1	0
0	0	0	0	0

Then from equation (20), we arrive at the following kernel for the Bi-Laplacian operator:

0	0	1	0	0
0	2	-8	2	0
1	-8	20	-8	1
0	2	-8	2	0
0	0	1	0	0

As in the case of deriving the Laplacian kernel the diagonal directions are now incorporated by taking the co-ordinates along the diagonals through a $\pi/4$ radian rotation. The new kernel thus obtained is then added as before, to the above kernel so that we arrive at the mask:

1	0	1	0	1
0	-6	-6	-6	0
1	-6	40	-6	1
0	-6	-6	-6	0
1	0	1	0	1

But, unlike the Laplacian, this being a 5×5 mask, the asymmetry still remains and in order to arrive at an omnidirectional mask for the isotropic ∇^4 operator, we apply another $\pi/8$ radian rotation so that we may also incorporate the off-diagonal elements. Then once again adding the new kernel thus obtained to the above mask, the final form that the Bi-Laplacian mask assumes is:

1	1	1	1	1
1	-12	-12	-12	1
1	-12	80	-12	1
1	-12	-12	-12	1
1	1	1	1	1

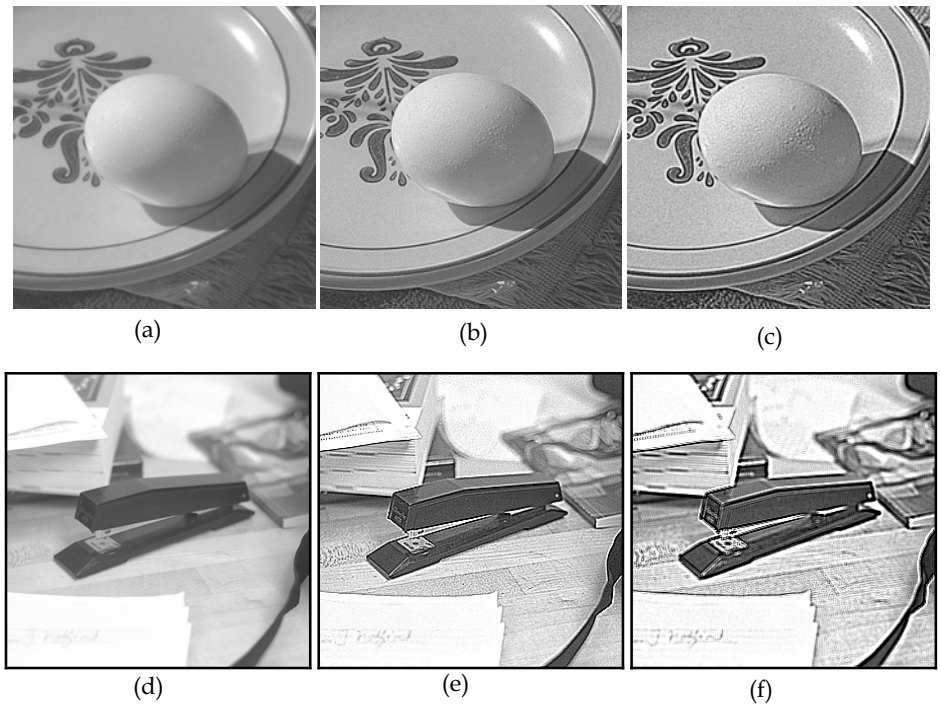


Fig. 10 The two benchmark images (a) and (d) used in section 4, have been enhanced with the discrete Laplacian mask in (b) and (e) and by the derived digital mask in (c) and (f).

Since, the non-classical receptive field has been modelled by Equation (19) in the previous sub-section, therefore we shall now try to arrive at a new omnidirectional mask that is comparable to the omnidirectional Laplacian mask and at the same time whose spatial organization reflects the disinhibitory extended surround as an added feature to the lateral inhibition evident in the spatial organization of the Laplacian mask. We show below one such possiblity. We choose the value of m in Equation (19), so that if we combine the Laplacian and the Bi-Laplacian masks by a ratio of $9:1$, we arrive at such a new 5×5 discrete filter comparable in simplicity to the 3×3 Laplacian mask:

-1	-1	-1	-1	-1
-1	3	3	3	-1
-1	3	-8	3	-1
-1	3	3	3	-1
-1	-1	-1	-1	-1

Correspondingly, by including the original intensity distribution to such a derivative opeartor, as a modification to the proposal of Mach given by Equation (5), we get a new spatial organization for the non-classical receptive field that includes disinhibitory inputs

from the surround extended from the classical excitatory-inhibitory organization of receptive field:

-1	-1	-1	-1	-1
-1	3	3	3	-1
-1	3	-7	3	-1
-1	3	3	3	-1
-1	-1	-1	-1	-1

This is the new omnidirectional mask whose performance in enhancing edges, we can now compare with the omnidirectional Laplacian mask. From visual inspection (Fig. 10) it is clear that this new discrete filter derived from a combination of Laplacian and Bi-Laplacian, indeed performs better compared to the discrete Laplacian mask. The Mach bands have been further enhanced by the new discrete filter as compared to the discrete Laplacian filter, which leads to better segregation of objects from background and hence better edge enhancement. The incorporation of disinhibition has therefore further improved edge enhancement.

5.3 Explanation of Complex Brightness-contrast Illusions

As we have already seen that the Mach band illusion can be well explained by the DOG model of classical receptive field. Some other brightness-contrast illusions like the Simultaneous brightness-contrast effect or the grating induction effect can also be explained by the classical model. The Simultaneous brightness-contrast is usually described as a homogenous brightness change within an enclosed test patch such that a gray patch on a white background looks darker than an equally luminous gray patch on a black background (Fig. 11a). This phenomenon is also well explained by the isotropic DOG model, as shown in Fig. 11b, where we have drawn a horizontal profile through the two test patches in the image that is obtained by convoluting the original image with the DOG function given by Equation (8).

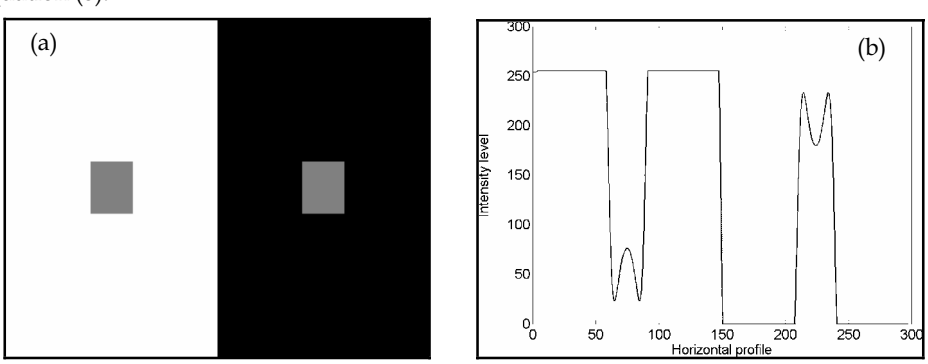


Fig. 11(a) The Simultaneous Brightness-contrast illusion (b) Explanation by convolution with DOG model along a horizontal line profile through the equiluminant test patches in the convolved image, showing the difference in brightness perception.

Grating Induction, on the other hand, refers to a periodic apparent contrast induced in uniform fields by adjacent gratings. This image displays a brightness effect that produces a spatial brightness variation (a grating) in an extended test patch (Fig. 12a). This effect can

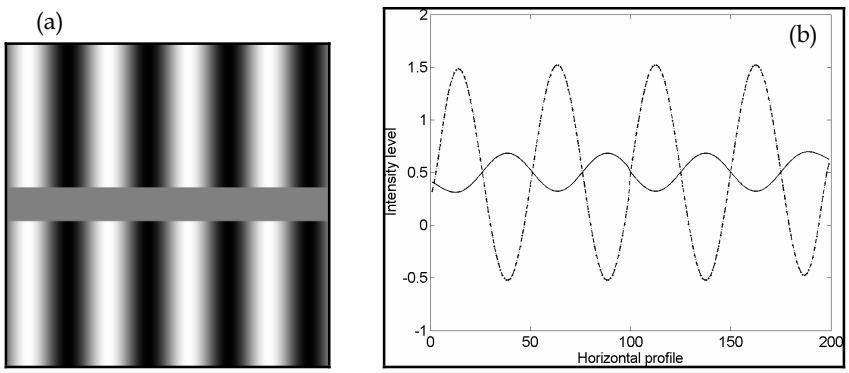


Fig. 12 (a) The Grating Induction illusion. (b) Explanation by convolving the image with DOG model along two horizontal line profiles, one through the constant intensity test patch (solid line) and one through the grating (dotted line) in the convolved image.

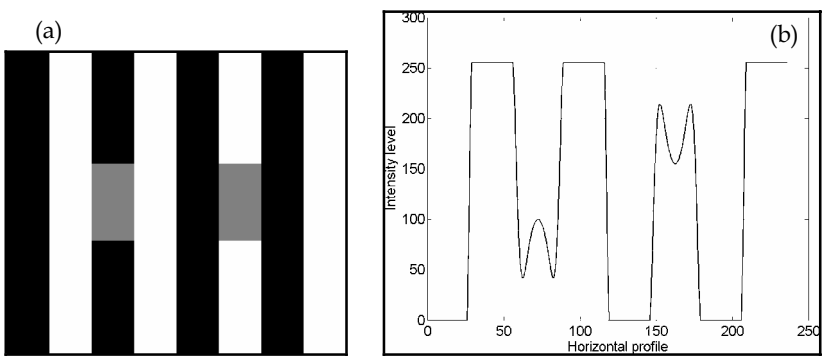
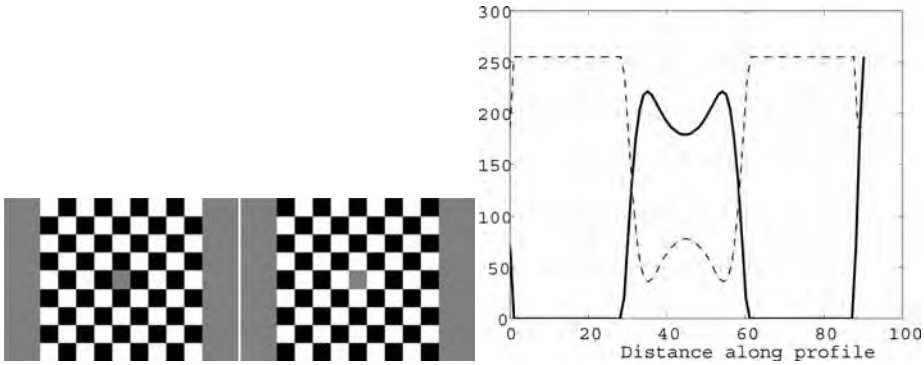


Fig. 13 (a) The White effect illusion. (b) Attempted explanation with conventional isotropic DOG function along a horizontal line profile through the equiluminant gray segments in convolved image, gives results in brightness perception contrary to our visual sensation.

also be similarly explained by the DOG model as has been shown in Fig. 12b, by drawing two horizontal profiles one through the test patch and the other through the grating.

However, many other brightness-contrast illusions like the White effect and the checkerboard illusion cannot be explained using the classical DOG model. In the White effect, for example in a square grating of black and white bars, if identical gray segments are used to replace part of the black bars and also part of the white bars, then the former gray segments look brighter than the later (Fig. 13a). Conventional isotropic DOG filters, fail to simulate this illusion and produce results contrary to our perception (Fig. 13b). The effect is specifically interesting because it does not depend on the amount of dark or white border in the vicinity of the test patch. True, that the effect may be generated if lateral inhibition shows directional properties i.e. inhibition is supposed to be stronger along the bars than

across them, but such a supposed anisotropy in lateral inhibition is not observed in White's effect on checkerboard (Fig. 14a), a symmetric image that cannot be explained with the



(a) (b)

Fig. 14 (a) The checkerboard illusion. The horizontal line profiles through the two test patches of the image obtained after convolution with DOG model. From the horizontal line profiles it is clear that the test patch on the left in darker neighbourhood (solid line representation) appears brighter compared to the one on right (dotted line representation), which is opposite to our perceptual experience.

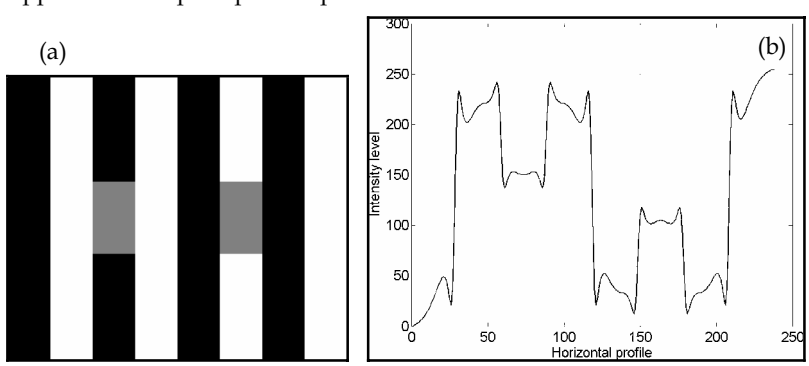
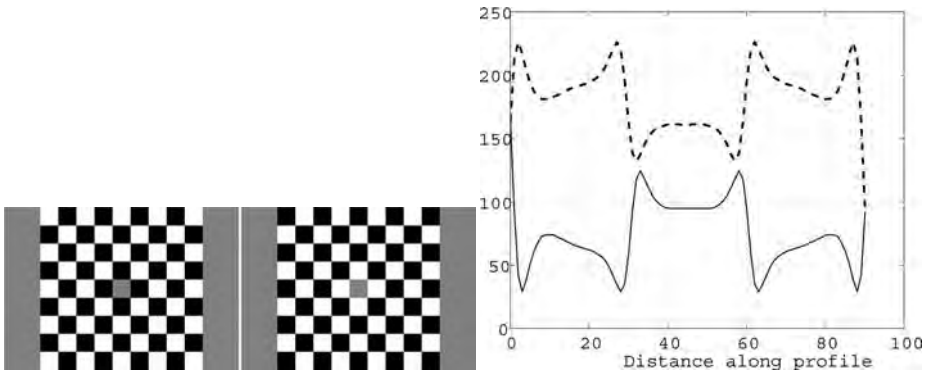


Fig. 15 Explanation of the White effect illusion by convolving the image with the $DoG + G_1$ model which produces results that match our brightness perception.

isotropic DOG model as well (Fig. 14b). Gestalt theorists believe that White effect can be understood only in terms of perception at a higher level and hence such illusions are often considered as more complex brightness-contrast phenomena that fall beyond the scope of low-level vision. Thus to probe whether the explanation of the White effects could have a basis in the retinal physiology, it would indeed be tempting to use the model of non-classical receptive field in the simulation of the White effects (Ghosh et al., 2006). We find that the White effect illusions, for both the anisotropic and isotropic (checkerboard) cases, where the DOG model failed completely, can be faithfully explained by convoluting the images with the function given by Equation (10), i.e. by the non-classical $DoG + G_1$ model. This has been shown in Fig. 15 and Fig. 16.



(a) (b)

Fig. 16 Explanation of the checkerboard illusion by convolving the image with the $DoG + G_1$ model producing results similar to our brightness perception.

5.4 A Possible Explanation of the Filling-in Mechanism in Retinal Blind Spot

It is well-known that human beings have a blind spot in each of their eyes. This blind spot is nothing but the area of the visual space that corresponds to the area on the retina, where all the optic nerves emanate from the retina (Fig. 17). It is called a blind spot because at this corresponding position, the retina is devoid of any rod or cone cell for receiving visual information. The area in visual space, marking the blind spot for one eye, is covered by the retina of the other eye. Curiously however, even in monocular vision, no hole is perceived in the visual field. This phenomenon is referred to as "filling-in" of the blind spot. According to many vision scientists (Ramachandran, 1992), the blind spot is not ignored, but "filling-in" is continually performed by the human visual system, constructing a representation based on the visual stimulation of the area surrounding the blind spot. Such an information processing based approach bears resemblance to David Marr's (Marr, 1982) computational investigation of human representation and processing of visual signals. Marr speculated that the computational theory of vision should cover three different possible phases in information processing: a) an early primal sketch of which "raw primal sketch" or detection of edges is the fundamental step, b) surface interpolation or the filling-in of colour and texture leading to the "two-and-half dimensional sketch" and c) object reconstruction and classification being the final step. So according to this theory, interpolation is an integral part of image retrieval in vision. In a bid to understand the process of interpolation, Ramachandran has performed some psychophysical experiments to come up with very interesting results on the "filling-in" of blind spots. He has shown that this "filling-in" process must occur as early as the detection of edges in the simple cells of primary visual cortex. However such interpolation cannot be explained by the classical DOG function of low-level receptive field. For any kernel $h(x)$ to qualify for an interpolator it must obey the following conditions:

$$\begin{cases} h(x) \equiv 1, \ x = 0 \\ h(x) \equiv 0, \ |x| = 1, 2, 3 \ldots\ldots \end{cases} \tag{21}$$

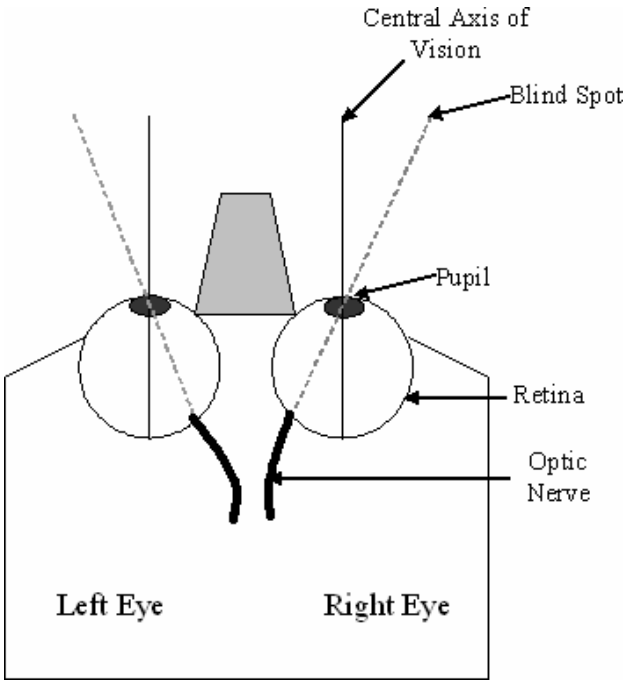


Fig. 17 A rough schematic of the eye that demonstrates the existence of the blind spot in the retina of each of the eyes from where the optic nerves emerge out towards the brain.

Secondly, it must also comply with the condition for dc-constancy, which implies that the sum of the samples of the interpolator should be unity for any displacement $0 \le d < 1$ i.e.:

$$\sum_{c=-\infty}^{\infty} h(c+d) \equiv 1 \tag{22}$$

Functions that do not fulfill Equations (21) and (22) are called 'approximators' and do not represent the ideal interpolators. The ideal interpolation function for convolution is the sinc function:

$$h(x)_{ideal} = \frac{\sin(\pi x)}{\pi x} = \sin c(x)$$

It has an infinite support having innumerable zero-crossings. This needs to be truncated to obtain a finite support interpolation kernel. From this consideration, the DOG response function of the classical receptive field given in Fig. 18a, should be an unlikely contender for performing the task of interpolation. This is because by comparing the kernel, with the second condition in Equation (21), we easily realize that this interpolator can have only one zero-crossing at $|x| = 1$, and can therefore at best mimic the highly truncated sinc interpolator within the interval $-1 \le x \le 1$. It will thus behave poorly in frequency domain and invariably produce erroneous results, unlike our almost perfect visual experience in the filling-in of blind spot. So as in the case of the complex brightness-contrast illusions, we again feel

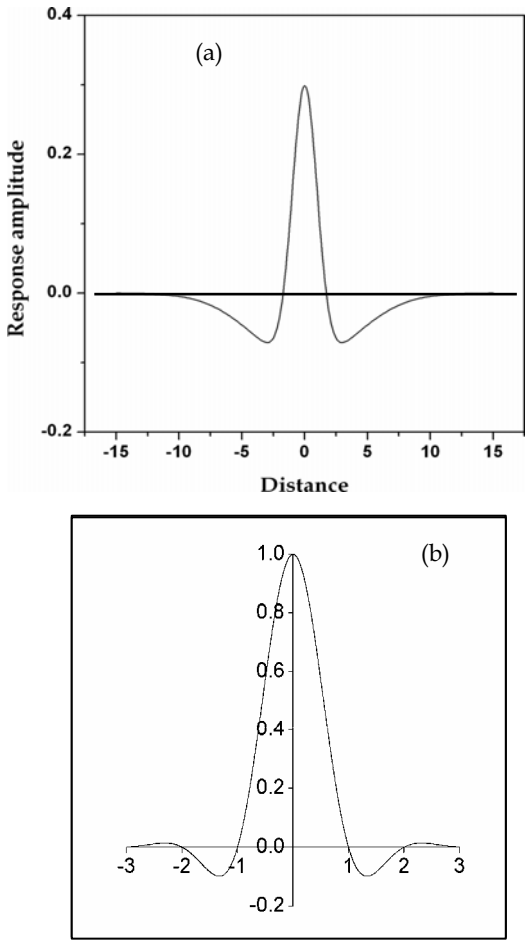


Fig. 18 (a) The DOG kernel in one dimension can have only one zero-crossing at $|x| = 1$. It is therefore not possible to design a good interpolator with such a function. (b) The $DoG + G_1 - G_2$ kernel is being shown here as a near ideal interpolator with 3 zero-crossings at $|x| = 1,2,3$ and $h(x) = 1$ at $x = 0$

tempted to investigate if our model of non-classical receptive field can suit the purpose of near ideal interpolation in low-level vision. For this we use Equation (15) as the convolution function for interpolation or in other words the $DoG + G_1 - G_2$ model of non-classical receptive field. From Fig. 18b we find that four zero-mean Gaussians representing the non-classical receptive field, can produce 3 zero-crossings at $|x| = 1,2,3$ (Sarkar et al., 2005).

This kernel, it is easy to verify will have excellent frequency domain as well as dc-constancy behaviour (Fig. 19) and is therefore a reasonably good contender for performing near ideal interpolation in the blind spot of the retina. Hence the proposal, put forward from the observations on the psychophysical experiments (Ramachandran, 1992) that information

corresponding to the blind-spot can be interpolated out at an early stage of visual processing, is also vindicated, since the interpolation function used here is a low-level receptive field model only.

6. Conclusion

The theory of edge detection and the treatise on low-level vision presented in this chapter in the light of the non-classical receptive field of retinal ganglion cells is a straightforward continuation of the approach of David Marr and his group. The appeal of the present approach lies in its simplicity and easy implementation, although it should be kept in mind that no non-linear model of the extended surround has been proposed here, which could be an interesting direction of future work. Potential applications of the algorithm will include apart from areas of general edge enhancement, designing new robust visual capturing or

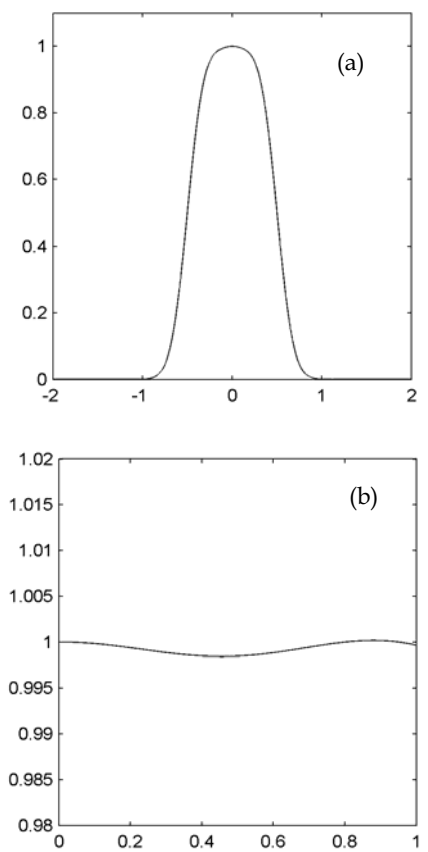


Fig. 19 Representative curves for the interpolation kernel constructed using the $DoG + G_1 - G_2$ function (a) Fourier spectrum of the kernel and (b) dc constancy bahaviour of the kernel

or display systems and automatic detection and correction of perceived incoherence of luminance in video display panels, where accurate perception of intensity level is critical. Such applications will be important particularly in mission-critical domains such as aircraft display panel design. Also, the concept of disinhibition introduced into the low-level receptive field structure, can be extended in future to higher brain functions such as categorization and memory. It is possible that a close analysis of cortical horizontal connections and their physiology under the disinhibition framework can provide us with new insights on their functions. This in turn will allow us to apply the general concept of disinhibition in advanced intelligent systems, firmly based on biological observations.

7. References

Barlow, H. B. (1953). Summation and inhibition in the frog's retina. *Journal of Physiology (London),* 119, 69-88

Helmholtz, H. von (1867/1925). *Treatise on Physiological Optics*, (from Third German Edition, Trans.), Dover Publications, New York

Ghosh, K., Sarkar, S. & Bhaumik, K. (2006) A possible explanation of the low-level brightness-contrast illusions in the light of an extended classical receptive field model of retinal ganglion cells. *Biological Cybernetics*, 94, 89-96.

Ma, S. D. & Li, B. (1998). Derivative computation by multiscale filters. *Image and Vision Computing*, 16, 43-53

Mach, E. (1865/1965). On the effect of the spatial distribution of the light stimulus on the retina, Trans. In: *Mach Bands: Quantitative Studies on Neural Networks in the Retina*, F. Ratliff (Ed.), Holden-Day, San Francisco

Marr, D. (1982). *Vision,* W. H. Freeman and Company, ISBN 0-7167-1567-8, New York

Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London B,* 207, 187-217

Passaglia, C.L., Enroth-Cugell, C. & Troy, J. B. (2001). Effects of remote stimulation on the mean firing rate of cat retinal ganglion cell. *Journal of Neuroscience*, 21, 5794-5803

Ramachandran, V. S. (1992). Blind spots. *Scientific American,* 266 (5), 44-49

Sarkar, S., Ghosh, K. & Bhaumik, K. (2005). A weighted sum of multi-scale Gaussians generates new near ideal interpolation functions, *Proceedings of 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6387-6390, ISBN: 0-7803-8741-4, Shanghai, China, September, 2005, IEEE Press, New Jersey.

19

Green's Functions of Matching Equations: A Unifying Approach for Low-level Vision Problems

José R. A. Torreão, João L. Fernandes, Marcos S. Amaral
& Leonardo Beltrão
Universidade Federal Fluminense
Brazil

1. Introduction

Green's functions are a traditional technique for solving inhomogeneous differential equations which has found several applications in pure and applied science, as, for instance, in Electrodynamics or Quantum Mechanics (Hassani, 2002). Given a one-dimensional linear differential operator, Ω_x, and a set of boundary conditions, the solution to the inhomogeneous differential equation $\Omega_x f(x) = g(x)$ can be expressed as

$$f(x) = \int_D G(x, x_0) g(x_0) dx_0 \tag{1}$$

where D is the domain of interest, and where the operator $G(x, x_0)$, called the Green's function, is the solution to the equation

$$\Omega_x G(x, x_0) = \delta(x - x_0) \tag{2}$$

under the same boundary conditions, with $\delta(x - x_0)$ denoting Dirac's delta function. Operating on both sides of (1) with Ω_x, and making use of equation (2), we obtain

$$\Omega_x f(x) = \int_D [\Omega_x G(x, x_0)] g(x_0) dx_0 = \int_D \delta(x - x_0) g(x_0) dx_0 = g(x) \tag{3}$$

where the sieving property of the delta function has been used, what proves that the function f(x), given by (1), is indeed the solution sought.
It should also be noted that, if $H(x, x_0)$ is another integral operator, satisfying

$$\Omega_x H(x, x_0) = 0 \tag{4}$$

the complex kernel $K(x, x_0) = (G + iH)(x, x_0)$ can be formed, such that

$$\bar{f}(x) = \int_D K(x, x_0) g(x_0) dx_0 \tag{5}$$

is also a solution to the original differential equation, *i.e.*, it satisfies $\Omega_x \bar{f}(x) = g(x)$.

Recently, the use of Green's functions of image matching equations has been proposed as a suitable means for approaching several visual computing problems, including shape from shading (Torreão, 2001; Torreão, 2003; Torreão & Fernandes, 2004), edge detection (Torreão & Amaral, 2002 and 2006), motion simulation (Ferreira Jr. et al., 2004), and video interpolation (Ferreira Jr. et al., 2005).

Image matching equations have been used in computer vision and image processing for modeling such processes as stereoscopy (Barnard, 1986) and optical flow (Horn & Schunck, 1981). For instance, if I_1 and I_2 are two images of a dynamic scene, captured at consecutive times by a static camera, the optical flow constraint can be expressed as the intensity conservation condition

$$I_2(x+U, y+V) = I_1(x,y) \tag{6}$$

where U(x,y) and V(x,y) are the optical flow components along directions x and y, respectively. The goal is then to use such image matching condition for estimating U and V, what is generally done by first expanding equation (6) in a Taylor-series up to first order in the flow, and using it along with other constraints (that would express, for instance, the smoothness of the flow components), in order to allow the solution of such ill-posed problem.

In the Green's function approach, on the other hand, a different use is made of the matching equation: assuming that the flow field is known (*e.g.*, a uniform or an affine flow), equation (6) is solved for the matching image I_2, given I_1. For instance, assuming uniform flow along the direction $\theta = \arctan\gamma$ (*i.e.*, U(x,y) = u and V(x,y) = v, for both u and v constants, with $\gamma = v/u$), and taking a second-order Taylor-series expansion, the matching equation becomes

$$\frac{u^2}{2}\frac{\partial^2 I_2}{\partial x^2} + u\frac{\partial I_2}{\partial x} + I_2 = I_1 \tag{7}$$

with $I_i = I_i(x, y+\gamma x)$. The solution to the above can be expressed as

$$I_2(x, y+\gamma x) = \int_D G_u(x-x_0)\, I_1(x_0, y+\gamma x_0)dx_0 \tag{8}$$

where $G_u(x-x_0)$ is the Green's function to equation (7), *i.e.*, it is the solution to that equation when a delta function is substituted for I_1 on its right-hand side. If we want bounded solutions over an infinite domain $D = (-\infty, \infty)$, G_u will take the form (Torreão, 2001)

$$G_u(x-x_0) = \frac{2}{u}\sin\left(\frac{x-x_0}{u}\right)\exp\left(-\frac{x-x_0}{u}\right) \tag{9}$$

for $x > x_0$, with $G_u = 0$, otherwise. It will thus be a causal, shift-invariant operator.

Different kinds of Green's functions will result from different flow assumptions. If, instead of the uniform flow, we considered a one-dimensional affine model, with $U(x) = u_0 + u_1 x$, for constant u_0 and u_1 and with $V(x) = \gamma U(x)$, the matching equation would become

$$\frac{(u_0 + u_1 x)^2}{2}\frac{\partial^2 I_2}{\partial x^2} + (u_0 + u_1 x)\frac{\partial I_2}{\partial x} + I_2 = I_1 \tag{10}$$

whose Green's function, again if we require bounded solutions over an unbounded domain, will be (Ferreira Jr. et al., 2004)

$$G_U^{(1)}(x, x_0) = \frac{2}{u_1^2 \beta (x_0 - x_U)}\left[\frac{x - x_U}{x_0 - x_U}\right]^{\alpha} \sin\left\{\beta \log\left[\frac{x - x_U}{x_0 - x_U}\right]\right\} \tag{11}$$

for $x > x_0$, with $G_U^{(1)}(x, x_0) = 0$, otherwise. In equation (11), the parameter x_U is defined as $x_U = -u_0 / u_1$, and corresponds to the fixed point of the affine transformation, since we have $U(x_U) = 0$. The parameters α and β are given as

$$\begin{cases} \alpha = -\dfrac{1}{u_1} + \dfrac{1}{2} \\ \beta = \dfrac{1}{u_1}\sqrt{1 + u_1 - \dfrac{u_1^2}{4}} \end{cases} \tag{12}$$

The Green's function, in this case, is a shift-variant operator which remains bounded over a domain $D \subset (x_U, \infty)$, so long as we take $0 < u_1 < 2$. Over finite domains, this solution is valid for $2 - 2\sqrt{2} < u_1 < 2 + 2\sqrt{2}$.

Still another form of Green's function results from considering the matching equation under the guise

$$\frac{u_0^2}{2}\frac{\partial^2 I_2}{\partial x^2} + (u_0 + u_1 x)\frac{\partial I_2}{\partial x} + I_2 = I_1 \tag{13}$$

which is a variant of the affine matching condition, identical to equation (10) up to first order in u_0. The bounded Green's function for the above, over a domain $D \subset (x_U, \infty)$, can be approximated, when $|x_U| >> x, x_0$, as (Torreão & Fernandes, 2004)

$$G_U^{(2)}(x, x_0) = \frac{2|x_U|}{\sigma^2}\sin\left[\frac{|x_U|}{\sigma^2}(x - x_0)\right]\exp\left[-\frac{(x - x_U)^2 - (x_0 - x_U)^2}{2\sigma^2}\right] \tag{14}$$

for $x > x_0$, with $G_U^{(2)}(x, x_0) = 0$, otherwise. It can be easily verified that, similarly to our first affine form, $G_U^{(1)}$, the filter $G_U^{(2)}$ reduces to the uniform Green's function, G_{u_0}, in the limit of $u_1 \to 0$, as should be expected, where $u_0 = \sigma^2 / |x_U|$.

All the Green's functions considered can be interpreted as point spread functions which generate motion through a linear model: when filtering an input image, they induce a

displacement of the image features, accompanied by a loss of high frequencies which can be interpreted as motion blur. The potential of this for motion synthesis in computer graphics is evident, and has been extensively explored, based on the filter $G_U^{(1)}$ (Ferreira Jr. et al., 2004 and 2005).

Here, we will be mainly concerned with the computer vision applications of the approach, which have been based on the forms G_u and $G_U^{(2)}$. Such applications also stem from the motion induction capabilities of these Green's functions. For instance, given a single input image, a second image can be generated which simulates the photometric stereo pair to the input, representing the same scene under displaced illumination. This has been used as the basis to the Green's function shape from shading (Torreão, 2001), which extends, to single-image reconstruction, a two-image photometric-stereo approach (Torreão & Fernandes, 1998). Similarly, a depiction of the scene under a displaced point of view can also be simulated from a given image, what has led to the Green's function photometric motion (Torreão & Fernandes, 2004), also extending, to the single-input case, a multi-image process (Torreão et al., 2007).

Signal differentiation is another computer-vision/image-process application where the Green's functions of matching equations have found use (Torreão & Amaral, 2002 and 2006). This comes along naturally, when we remember that the first-order derivative of a signal can be approximated through the difference of displaced versions of it. Finally, we will here introduce a new application area for our Green's functions, by showing that, through their means, displaced versions of binocular image pairs can be generated whose local degree of matching yields a reliable measure of stereoscopic disparity.

The remainder of this chapter is organized as follows: in Section 2 and Section 3, we review the Green's function approaches to signal differentiation and to shape from shading, both based on the uniform-matching Green's function, G_u; in Section 4, we review the Green's function photometric-motion process, based on the affine Green's filter, $G_U^{(2)}$ (which, for simplicity, will henceforth be referred simply as G_U), and, in Section 5, we introduce the use of the same filter for stereoscopic disparity estimation. Our final remarks close the chapter in Section 6.

2. Signal Differentiation through Green's Functions

The Green's function approach to signal differentiation is based on the following rationale: If $I'(x)$ is the derivative of a signal $I(x)$, it can be expressed as

$$I'(x) = \lim_{u \to 0} \frac{I(x+u) - I(x-u)}{2u} \tag{15}$$

According to the Green's function approach summarized above, an estimate of the signal $I(x-u)$, let us call it $I_-(x)$, can be obtained as (see equation (8))

$$I_-(x) \equiv I(x-u) = \int_{-\infty}^{\infty} G_u(x-x_0) I(x_0) dx_0 \tag{16}$$

where G_u (see Fig. 1) is the uniform-matching Green's function, as presented in (9). The identity in equation (16), valid up to second order in u, results from inverting the matching relation $I_-(x+u) = I(x)$, which is a special case of equation (6).

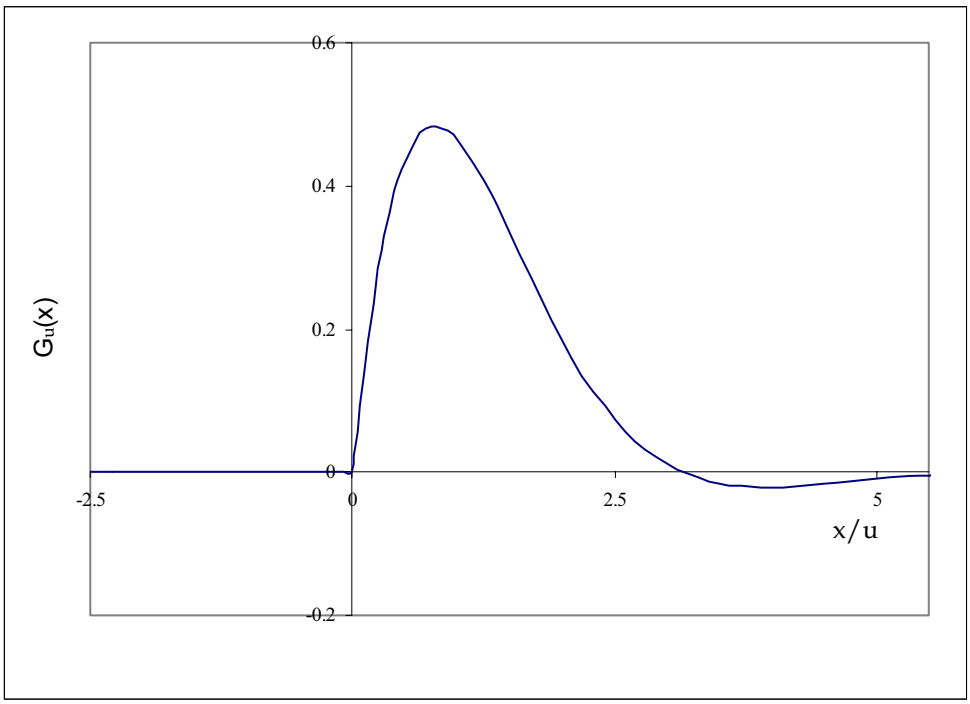


Fig. 1. Green's function $G_u(x)$, as a function of x/u.

Similarly, a signal $I_+(x)$, which is an estimate of I(x + u), can be obtained as

$$I_+(x) \equiv I(x+u) = \int_{-\infty}^{\infty} G_u(x_0 - x) I(x_0) dx_0 \tag{17}$$

where it can be easily verified that the filter $G_u(x_0 - x)$ will be the Green's function of a matching equation of the form $I_+(x-u) = I(x)$.

Using relations (16) and (17) in equation (15), and applying the commutative property of the convolution, there results the derivative estimate

$$I'(x) = \lim_{u \to 0} \frac{1}{2u} \int_{-\infty}^{\infty} \left[G_u(-x_0) - G_u(x_0) \right] I(x - x_0) dx_0 \tag{18}$$

and we have thus arrived at a linear operator, $D_u(x) = \frac{1}{2u}[G_u(-x) - G_u(x)]$, which, in the limit of $u \to 0$, becomes the impulse response of a differentiator. $D_u(x)$ turns out to be a special case of Deriche's well-known edge-detector (Deriche, 1987)

$$d(x) = -\exp(-\alpha\,|\,x\,|)\,\sin\omega x \tag{19}$$

for $\alpha = \omega = 1/u$.

A more general form for our differential operator can be found if we allow for scale factors in the matching equations. For instance, we could consider the relations

$$I_{\pm}(\frac{x}{\eta} \mp u) = I(x) \tag{20}$$

to obtain the derivative estimate

$$I'_{\eta}(x) = \lim_{u \to 0} \frac{1}{2\eta u} \int_{-\infty}^{\infty} [G_{\eta u}(-x_0) - G_{\eta u}(x_0)] I(x - x_0) dx_0 \tag{21}$$

thus arriving at

$$D_{\eta u}(x) = \frac{1}{2\eta u}[G_{\eta u}(-x) - G_{\eta u}(x)] \tag{22}$$

as a generalized version of our differentiator. Multiscale derivative estimates can then be obtained through linear combinations such as

$$I'_{est}(x) = \sum_{\eta} a_n I'_{\eta}(x) \tag{23}$$

where the a_η are real-valued constants, satisfying $\sum_\eta a_\eta = 1$. For instance, we may consider just two terms in the above expansion, to get

$$I'_{est}(x) = \frac{I'_1(x) + aI'_{\eta}(x)}{(1+a)} \tag{24}$$

From equations (18) and (21), we thus see that, in such case, our derivative estimate will be obtained by convolving the input signal with the operator (see Fig. 2)

$$D(x) = \frac{1}{2u}[F(-x) - F(x)] \tag{25}$$

where F(x), for x > 0, is given by

$$F(x) = A\left[\exp\left(-\frac{x}{\eta u}\right)\sin\left(\frac{x}{\eta u}\right) + K\exp\left(-\frac{x}{u}\right)\sin\left(\frac{x}{u}\right)\right] \tag{26}$$

with $A = \frac{a}{(1+a)(\eta u)^2}$ and $K = \eta^2 / a$.

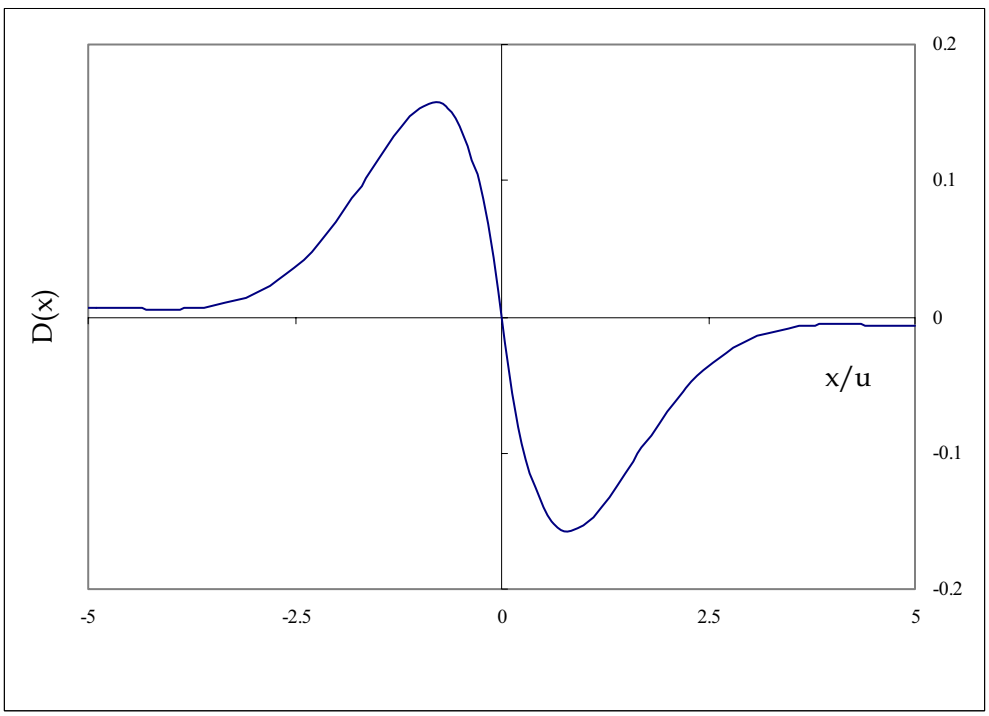
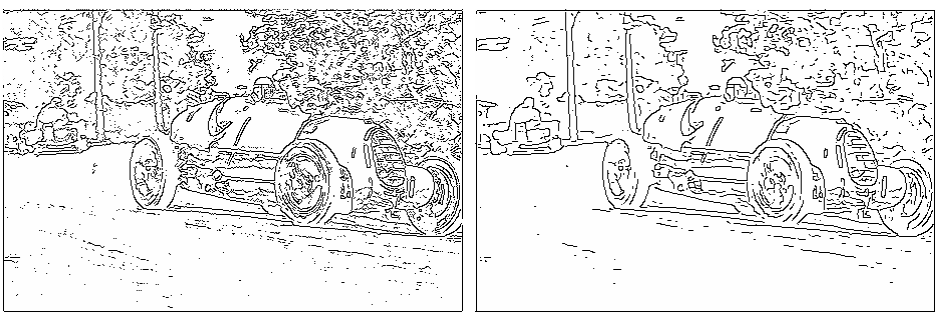


Fig. 2. Filter D(x), as a function of x/u.

In (Torreão & Amaral, 2006), a study was carried out which determined the values for a and η leading to maximum overall performance by the filter D(x), as measured through the (ΣΛ) SRC index introduced by Canny, where Σ, Λ and SRC denote, respectively, the detection, localization, and single-response measures (Canny, 1986). With a=1.1 and η=3.5, a (ΣΛ) SRC of 3.547 was achieved, beating the performance of alternative approaches, such as Sarkar and Boyer's filter, whose best mark is 3.388 (Sarkar & Boyer, 1991). Another advantage of the operator D(x), as also proven in (Torreão & Amaral, 2006), is that it allows simple recursive implementations, being realized as an infinite impulse response filter with only two poles and a single zero.

Fig. 3 shows examples of the use of operator D(x) for edge detection. In such 2D applications, D(x) was employed in the direction perpendicular to the edges sought, while a projection function - chosen here as the integral of D(x) - was used in the direction parallel to the edges. Non-maxima supression and hysteresis thresholding have also been employed, in the usual fashion (Canny, 1986).

(a)



(b) (c)

Fig. 3. Example of the use of operator D(x) for edge detection. (a) : input image. (b) and (c) : edges obtained for u = 0.05 and u = 1.0.

3. Green's Function Shape from Shading

Shape from shading (SFS) and photometric stereo (PS) are 3D shape estimation processes that take shading images as inputs - that is to say, they work with textureless images where a smooth gradient of intensities is observed, resulting solely from the orientation of the observed surfaces. SFS estimates surface orientation from a single shading image, while PS works with two or more monocular shading inputs, acquired under different illuminations. Both processes have been traditionally based on the so-called image irradiance equation, which relates the intensity at each image point to the surface gradient at the corresponding location in the scene, via the reflectance map function (Zhang et al., 1999), as

$$I(x,y) = R(p,q) \tag{27}$$

where

$$p = \frac{\partial Z}{\partial x}, \text{ and } q = \frac{\partial Z}{\partial y} \tag{28}$$

are the gradient components of the observed surface, Z(x,y), and where R is the reflectance map.

In (Torreão & Fernandes, 1998), an approach to photometric stereo was introduced, called the disparity-based photometric stereo (DBPS), whereby a pair of PS images are matched, similarly as a stereoscopic pair (Barnard, 1986), to yield a disparity map from which the shape of the observed surfaces can be recovered. Such disparity map results from the displacement of the irradiance pattern over the imaged surface, due to the change in illumination direction, a displacement that can be generally modeled as a non-uniform rotation, as proven in (Torreão, 2003).
DBPS is based on a pair of equations: a linear image irradiance equation, and a matching (optical flow) equation, that take the form

$$\begin{cases} \Delta I = k_0 + k_1 p + k_2 q \\ \Delta I = u\dfrac{\partial I_2}{\partial x} + v\dfrac{\partial I_2}{\partial y} \end{cases} \tag{29}$$

where $\Delta I \equiv I_1 - I_2$ is the difference of the input images, and where (u,v) denotes the optical flow, or disparity field.
Equating the two expressions for ΔI above, there results a differential equation on Z, whose approximate solution can be found as

$$Z = \frac{uI_2}{k_1} - \frac{k_0(x+\gamma y)}{k_1(1+\gamma^2)} \tag{30}$$

so long as the disparity component u is found by matching the input images along a direction such that $v/u = k_2/k_1 = \gamma$, a constant.
As proven in (Torreão, 2001), the DBPS approach can be extended to the single-input case, with the so-called Green's function shape from shading (GSFS). Here, the idea is to assume that the disparity field is uniform, and to solve the matching equation - considered up to second order in u, such as in (7) - for the matching image, I_2, via Green's function. It has been shown that, in such case, the estimated depth map takes the form

$$Z = \frac{u\bar{I}_2}{k_1} \tag{31}$$

with

$$\bar{I}_2 = I_2 + (aG_u + bG_u * H_u + cH_u) * I_2 \tag{32}$$

where a, b and c here are constants, the operation $*$ denotes a convolution, $I_2 = G_u * I_1$ is the matching pair to the input image I_1, and H_u is the homogeneous integral operator

$$H_u(x - x_0) = \frac{2}{u}\cos\left(\frac{x - x_0}{u}\right)\exp\left(-\frac{x - x_0}{u}\right) \tag{33}$$

for $x > x_0$, with $H_u = 0$, otherwise (see Fig. 4).

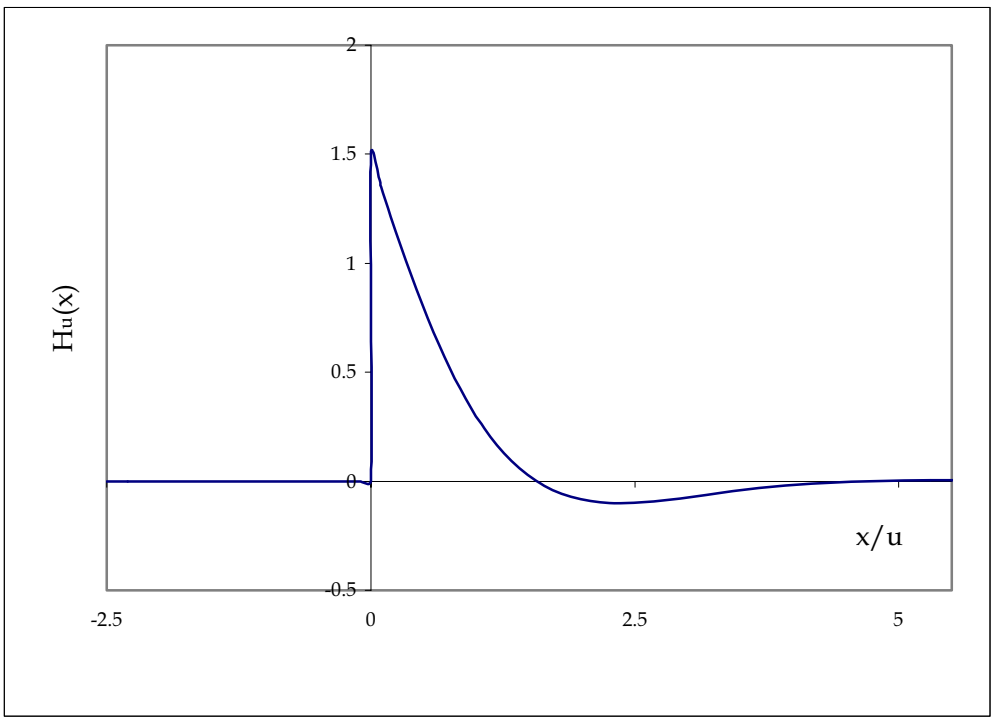


Fig. 4. Filter $H_u(x)$, as a function of x/u.

As can be easily verified, H_u satisfies the homogeneous form of equation (7), *i.e.*,

$$\frac{u^2}{2}H''_u + uH'_u + H_u = 0 \tag{34}$$

Besides the matching constant u, which must be chosen *a priori*, the single free parameter in equation (31) is k_1, and this can be estimated from the input data, as described in (Torreão, 1999). For this purpose, we take into consideration the fact that the displacement of the irradiance pattern over the scene, due to the change in illumination (it should be kept in mind that we are simulating a photometric stereo situation, via the Green's function) can be modeled as a non-uniform rotation. This allows the introduction of a least-squares structure-from-motion formulation that yields k_1. Fig. 5 shows examples of shape estimation via GSFS.

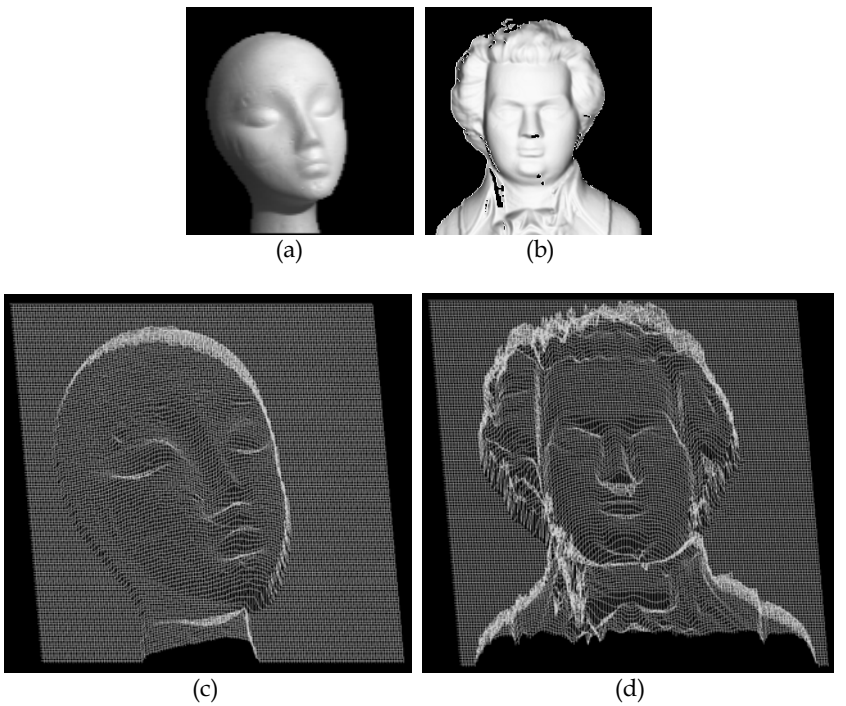


Fig. 5. Examples of shape estimation via GSFS. (a) and (b) : input images. (c) and (d) : estimated depth maps.

4. Green's Function Photometric Motion

Photometric motion is a shape estimation process introduced by Pentland, based on his observation that, for surfaces in rotation relative to the camera, the photometric effects of the motion (*i.e.*, the intensity change of a moving point) can prove more relevant than the geometric effects, due to projective distortion (Pentland, 1991). In his formulation, Pentland considered a quadratic expansion of the reflectance map, supposed symmetric and separable, and he also assumed that regions of approximately linear motion could be identified, allowing the registration of corresponding points in successive frames. Under such conditions, Pentland found that the intensity difference of registered points could be described by a linear reflectance map, and he thus used his linear shape from shading algorithm (Pentland, 1990) to obtain shape estimates of the imaged scene.

An alternative formulation of photometric motion has been recently introduced in (Torreão et al., 2007), along similar lines as followed for the disparity-based photometric stereo. A distinctive feature of this formulation is that of being based on the intensity change, due to the motion, at a fixed location in the image plane, and not, as in Pentland's approach, at a given point on the moving surface. This has the advantage of not requiring warping for the registration of corresponding points in the image sequence.
Similarly as DBPS, our novel approach to photometric motion relies on two expressions for the intensity change, due to the motion, at a given point in the image plane, one of them a

matching (optical flow) equation, and the other involving photometric (reflectance map) considerations.
Assuming a uniformly rotating surface, with angular velocity components A and B, along the x and y directions, such that

$$u = \frac{dx}{dt} = BZ, \quad \text{and} \quad v = \frac{dy}{dt} = -AZ \tag{35}$$

are the optical flow components, and also considering a linear image irradiance equation of the form $I = k_0 + k_1 p + k_2 q$, with $k_2 / k_1 = -A/B = v/u$, similarly as in DBPS, we arrive at the expression

$$\Delta I = \partial_\gamma \left[u(I - k_0) + \frac{k_1 u}{Z}(x + \gamma y) \right] \tag{36}$$

for the intensity difference, $\Delta I = I_1 - I_2$, of successive frames in the image sequence. In the above equation, we have used

$$\partial_\gamma \equiv \frac{\partial}{\partial x} + \gamma \frac{\partial}{\partial y} \tag{37}$$

where γ stands for the ratio v/u.
Now, again as in DBPS, we must couple equation (36) with an image matching equation, in order to find a closed-form expression for the depth map Z(x,y). The appropriate matching equation is here found to describe an affine optical flow field, taking the form

$$\Delta I = \left[u - (\partial_\gamma u) \left(\frac{x + \gamma y}{1 + \gamma^2} \right) \right] \partial_\gamma I \tag{38}$$

Equating (36) and (38), we find a diferential equation on Z, whose solution is given by

$$Z(x,y) = -\frac{k_1 (1 + \gamma^2) u}{(I - k_0)\, \partial_\gamma u + \dfrac{\kappa}{(x + \gamma y)}} \tag{39}$$

where κ is an arbitrary constant, provided that the term in $\partial_\gamma^2 u$ is neglected.
Through the Green's function approach, the above-described photometric motion formulation (whose results can be appreciated in (Torreão et al., 2007)) can be extended to the single-input case. In order to do this, we require a Green's function that will relate the matching image to the input image according to equation (38). Since that is a 1D expression, we may, without loss of generality, take the matching direction as x, to obtain

$$\Delta I \equiv I_1 - I_2 = (u_0 - u_1 x) \frac{\partial I_2}{\partial x} \tag{40}$$

where I_1 is the input image, I_2 is the image derived from it through the Green's function, and u_0 and u_1 are two constants representing, respectively, the disparity map and its

derivative at x = 0 (in the general case, these will mean $u_0 = u$ and $u_1 = \partial_\gamma u$). By comparing equation (40) to equation (13), we find that, up to first order in u_0, our photometric-motion Green's function will take the form of G_U in equation (14) (see Fig. 6).

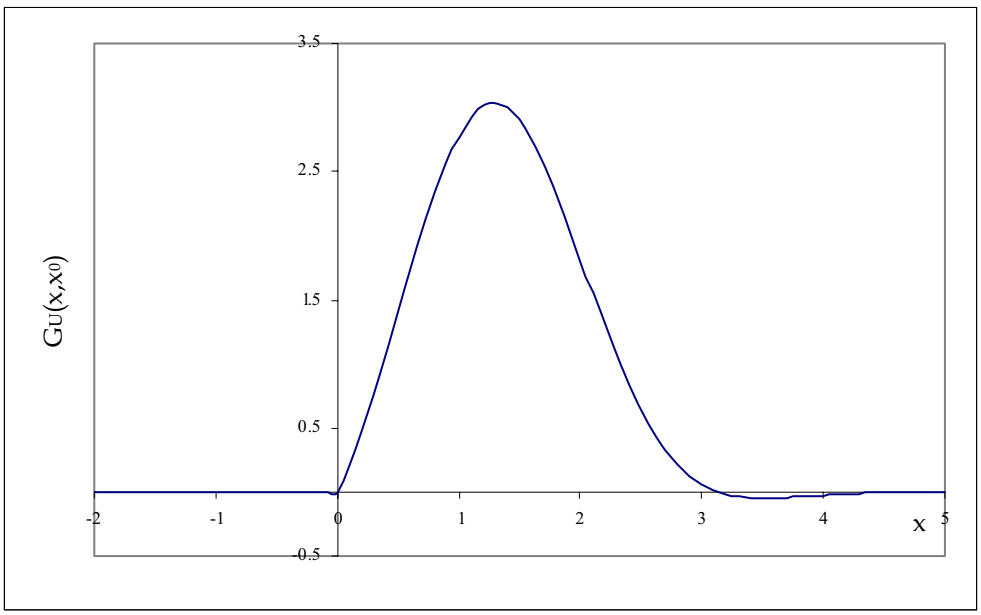
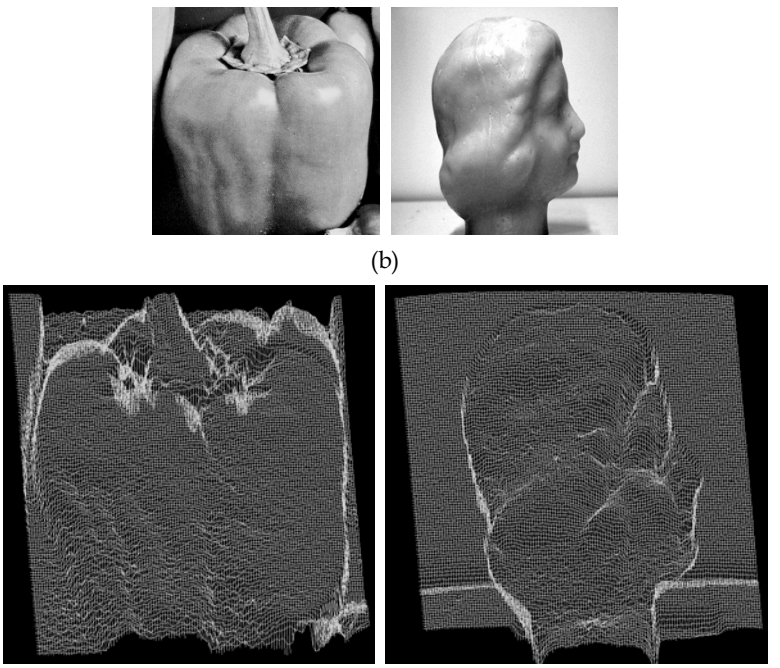


Fig. 6. Green's function $G_U(x,x_0)$, for $x_0 = 0$, $x_U = 1$ and $u_0 = 1$.

Using this to filter the input image, we obtain its matching pair I_2, which, substituted for I in equation (39), yields the shape estimate Z(x,y). Figure 7 illustrates results of this approach.



(b)

(c) (d)

Fig. 7. Examples of shape estimation via Green's Function Photometric Motion. (a) and (b) : input images. (c) and (d) : estimated depth maps.

5. Green's Function Stereoscopy

In a binocular vision system, scene features project at different positions in the two cameras, giving rise to the so-called binocular disparities, which constitute the primary cue for stereo vision (Barnard, 1986). Assuming a horizontal imaging configuration, a pair of left and right images which are projections of the same 3D scene should be related as

$$I_l(x+d,y) = I_r(x,y) \tag{41}$$

where $d \equiv d(x,y)$ here denotes the disparity map. The above is simply a special case of the matching equation (6), and, based on this, we can propose a Green's function approach to stereoscopic disparity estimation: Given the pair of binocular images, we can filter each of them through the appropriate Green's function, to induce different rightwards and leftwards shifts, aiming at the elimination of their intrinsic binocular disparities. By evaluating the degree of matching between the shifted inputs, for instance by computing the squared magnitude of their difference, we can then obtain an estimate of the disparity information encoded by the original stereo pair.

We have implemented such approach using the affine Green's filter of equation (14), keeping its σ parameter fixed, and varying x_U, in order to obtain different image shifts. Our preliminary results have proven encouraging, as shown by Figure 8 below.

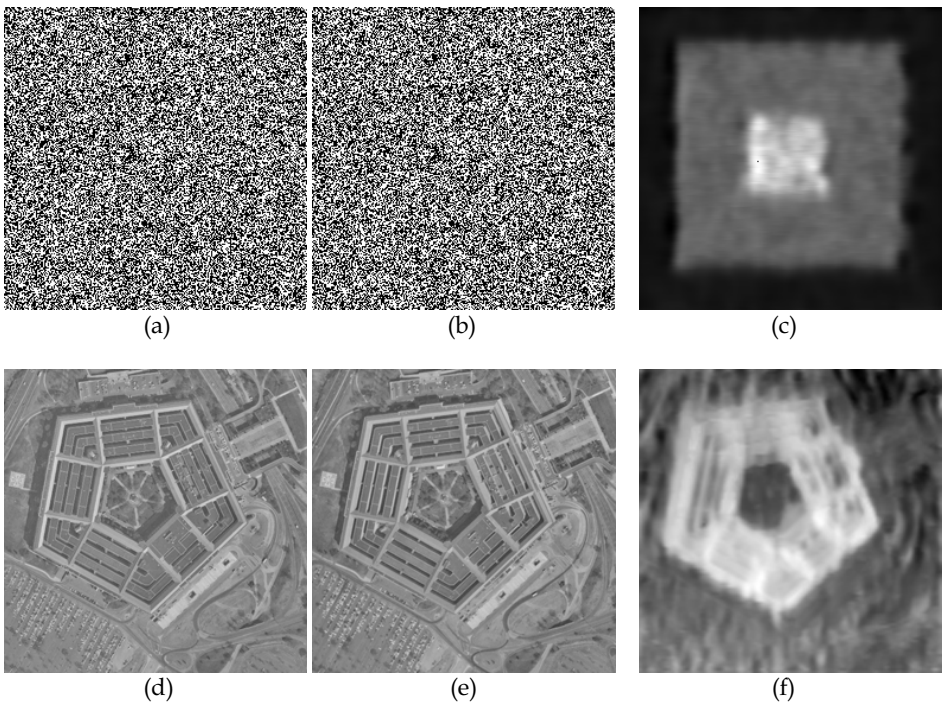


Fig. 8. Green's function approach to stereoscopic disparity estimation. (a) and (b): random-dot stereogram pair. (d) and (e) : real-world stereogram. (c) and (f) : estimated disparity maps.

6. Conclusion

We have reviewed the computer vision applications of Green's functions of image matching equations. Green's functions of both uniform- and affine-matching second-order differential equations have been considered, and we have illustrated their use for the computer vision problems of edge detection, monocular shape estimation, and stereoscopy. The Green's filters considered are essentially point-spread functions which have proven able to model the image-plane projection of a broad class of motions, along with their associated blur effects (Ferreira Jr. et al., 2004). As shown here, such motion modeling capability makes them suitable for a unifying approach to several low-level vision processes, whose full consequences still remain to be explored. This work has been supported by CNPq-Brasil.

7. References

Barnard, S.T. (1986) A stochastic approach to stereo vision, in *Proceedings of the Fifth National Conference on Artificial Intelligence*. Cambridge, Mass., MIT Press: 676-680.

Canny, J. (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6) pp. 679-698.

Deriche, R. (1987) Using Canny's criteria to derive a recursively implemented optimal edge detector, *International Journal of Computer Vision*, pp. 167-187.

Ferreira Jr., P. E.; Torreão, J. R. A. & Carvalho, P. C. P. (2004) Data-Based Motion Simulation Through a Green's Function Approach, *Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 193-199.

Ferreira Jr., P. E.; Torreão, J. R. A.; Carvalho, P. C. P. & Velho, L. (2005) Video Interpolation Through Green's Functions of Matching Equations, *Proceedings of the IEEE International Conference on Image Processing*.

Hassani, S. Mathematical Physics, Springer-Verlag, New York, 2002.

Horn B. & Schunck B. (1981) Determining optical flow, *Artificial Intelligence 17*, pp. 185-203.

Pentland, A. (1990) Linear shape from shading, *International Journal of Computer Vision 4*, pp. 153-162.

Pentland, A. (1991) Photometric motion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(9), pp. 879-890.

Sarkar, S. & Boyer, K.L. (1991) On optimal infinite impulse response edge detection filters,*IEEE Transactions on Pattern Analysis and Machine Intelligence 13(11)* pp. 1154-1171.

Torreão, J. R. A. (1999) A new approach to photometric stereo, *Pattern Recognition Letters 20(5)*, pp. 535-540.

Torreão, J. R. A. (2001) A Green's Function Approach to Shape from Shading, *Pattern Recognition 34*, pp. 2367-2382.

Torreão, J. R. A. (2003) Geometric-Photometric Approach to Monocular Shape Estimation, *Image and Vision Computing 21*, pp. 1045-1061.

Torreão, J. R. A. & Amaral, M. S. (2002) Signal Differentiation through a Green's Function Approach, *Pattern Recognition Letters 23(4)*, pp. 1755-1759.

Torreão, J.R.A. & Amaral, M.S. (2006) Efficient, Recursively Implemented Differential Operator, with Application to Edge Detection, *Pattern Recognition Letters 27(9)*, pp. 987-995.

Torreão, J. R. A. & Fernandes, J. L. (1998) Matching photometric stereo images, *Journal of the Optical Society of America A 15(12)*, pp. 2966-2975.

Torreão, J. R. A. & Fernandes, J. L. (2004) From Photometric Motion to Shape from Shading, *Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 186-191.

Torreão, J. R. A.; Fernandes, J. L. & Leitão, H.C.G. (2007) A novel approach to photometric motion, *Image and Vision Computing 25*, pp. 126-135.

Zhang, R.; Tsai, P.S.; Cryer, J.E. & Shah, M. (1999) Shape from shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21(8)* pp. 690-706.

20

Robust Feature Detection Using 2D Wavelet Transform under Low Light Environment

Youngouk Kim[1, 2], Jihoon Lee[1], Woon Cho[1], Changwoo Park[2], Changhan Park[1], Joonki Paik[1]

[1]*Image Processing and Intelligent Systems Laboratory, Department of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University.*

[2]*Precision Machinery Center, Korea Electronics Technology Institute Korea*

1. Introduction

Simultaneous localization and mapping (SLAM) requires multi-modal sensors, such as ultrasound, range, infrared (IR), encoder or odometer, and multiple visual sensors. Recognition-based localization is considered as the most promising method of image-based SLAM (Dissanayake, 2001). In practice, we cannot rely on the basic encoder output under kidnapping or shadowing environment. IR-LED cameras are recently used to deal with such complicated conditions. Map building becomes more prone to illumination change and affine variation, when the robot is randomly moving. The most popular solution for the robust recognition method is scale-invariant feature transform (SIFT) approach that transforms an input image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation (Lowe, 2004). The feature vector is partially invariant to illumination changes and affine (or three-dimensional) projection. Such local descriptor-based approach is generally robust against occlusion and scale variance. In spite of many promising factors, SIFT has many parameters to be controlled, and it requires the optimum Gaussian pyramid for acceptable performance. Intensity-based local feature extraction methods cannot avoid estimation error because of low light-level noise (Lee, 2005). Corner detection and local descriptor-based methods fall into this category. An alternative approach is moment-based invariant feature extraction that is robust against both geometric and photometric changes. This approach is usually effective for still image recognition. While a robot is moving, the moment-based method frequently recognizes non-planar objects, and can hardly extract invariant regions under illumination change. This paper presents a real-time local keypoint extraction method in the two-dimensional wavelet transform domain. The proposed method is robust against illumination change and low light-level noise, and free from manual adjustment of many parameters. Fig 1 displays whole structure of this paper.

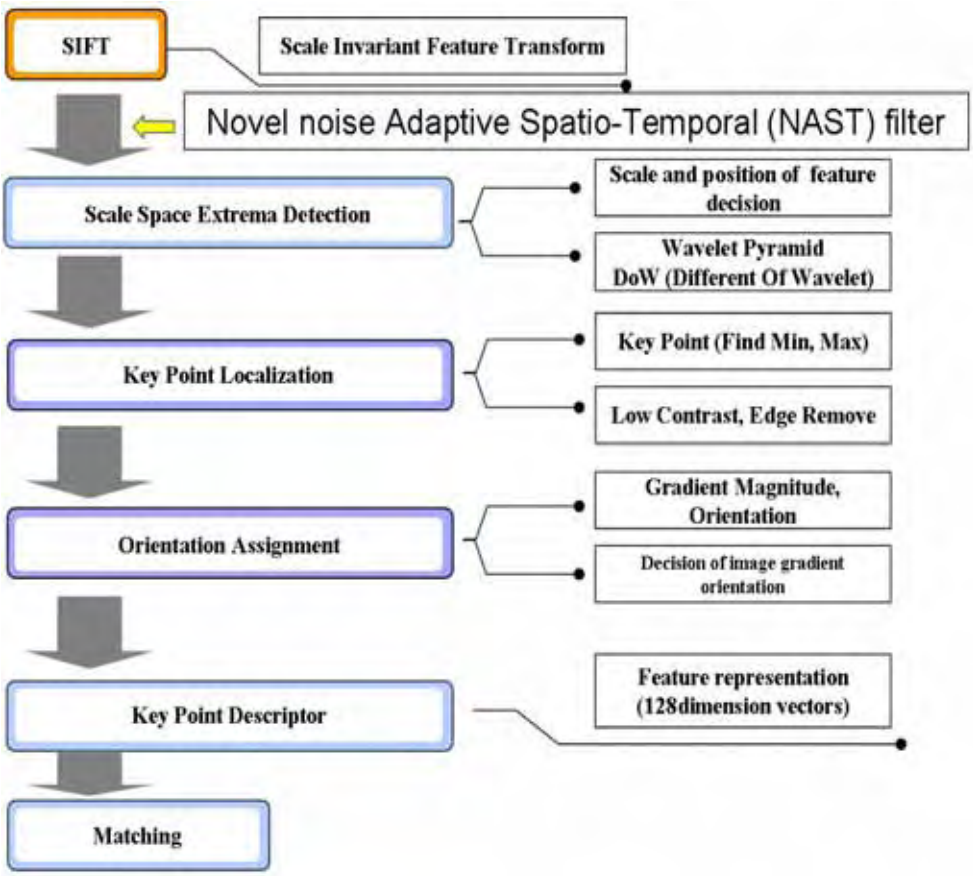


Figure 1. Conceptual flowchart of the whole structure

The paper is organized as follows. In section 2, noise adaptive spatio-temporal filter (NAST) is proposed to remove low light-level noise as a preprocessing step. Section 3 describes the proposed real-time local feature extraction method in the wavelet transform domain. Section 4 summarizes various experimental results by comparing DoW with SIFT methods, and section 5 concludes the paper.

2. Noise Adaptive Spatio-Temporal Filter

The proposed NAST algorithm adaptively processes the acquired image to remove low light level noise. Depending on statistics of the image, information of neighboring pixels, and motion, the NAST algorithm selects a proper filtering algorithm for each type of noise. A

conceptual flowchart of the proposed algorithm is illustrated in Fig. 2. The proposed NAST algorithm has four different operations which are applied to the low light images.

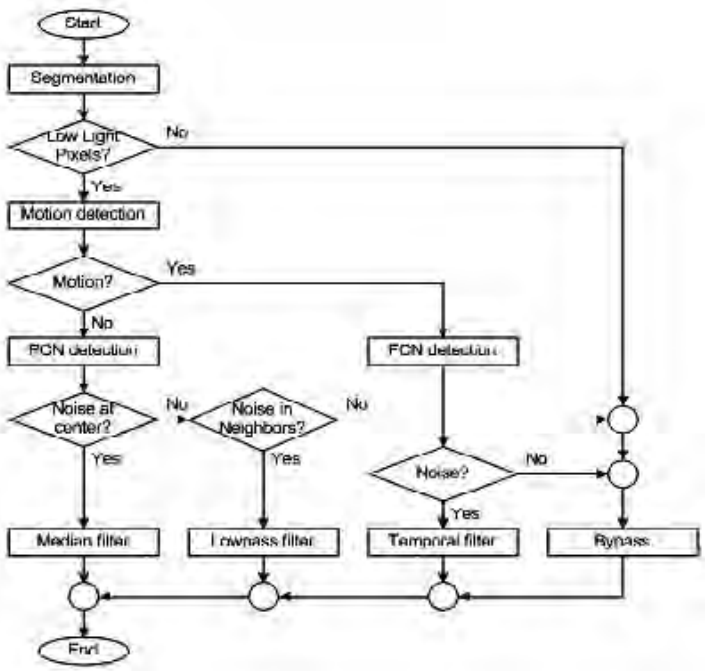


Figure 2. Conceptual flowchart of the proposed algorithm

2.1 Noise Detection Algorithm

The output of the noise detection block determines the operation of filtering blocks. The proposed spatial hybrid filter (SHF) can be represented as

$$y(i,j) = n(i,j) \times \hat{x}(i,j) + (1 - n(i,j)) \times x(i,j) \tag{1}$$

where $\hat{x}(i,j)$ represents a pixel filtered by the SHF and $n(i,j)$, which is the result of the noise detection process, takes 1 for the position of photon counting noise (PCN) pixels and 0 elsewhere. In equation (1), $x(i,j)$ and $y(i,j)$ denote the (i, j)-th pixels in noisy and filtered images, respectively. In the proposed noise detection scheme, $n(i,j)$ forms a binary noise map denoted by N, which is used to filter out uncorrelated noise and to indicate the reference points for the subsequent filtering of correlated noise.

2.2 Filtering Mechanism of SHF

If the central pixel in the window (W) is considered to be noise (i.e., $n(i,j)$ =1in the noise map N), it is substituted by the median value of the window as a normal median filter. Then

the noise cancellation scheme in SHF is extended to the correlated pixels in the local neighborhood ($x(i,j)$ where $n(i,j) \neq 1$ and at least one $n(k,l) = 1$ in W). In order to identify the correlated noise, the de-noised pixel value $x'(i,j)$ can be defined as

$$x'(i,j) = \frac{\sigma^2(i,j) \times x(i,j) + \overline{x}^2(i,j)}{\sigma(i,j) + \overline{x}(i,j)} \tag{2}$$

where $\overline{x}(i,j)$, and $\sigma^2(i,j)$ respectively represent the mean and variance of the window W.

2.3 Statistical Domain Temporal Filter (SDTF) for False Color Noise (FCN) Detection and Filtering

We use a new SDTF for removing FCN. The sum of the absolute differences (SAD) between the two working windows of consecutive frames is used for motion detection to avoid motion blur due to temporal averaging. Let $\hat{x}(i,j,t)$ and $\hat{x}(i,j,t-1)$ denote intensity values at the (i, j)-th pixel in the spatially filtered frames at time t and $t-1$, respectively, the proposed temporal filter can then be realized as

$$y(i,j,t) = \begin{cases} \hat{x}(i,j,t-1), & S_T(i,j,t) > S_T(i,j,t-1) \\ \hat{x}(i,j,t), & S_T(i,j,t) \leq S_T(i,j,t-1) \end{cases} \tag{3}$$

where $y(i,j,t)$ represents the final result of the proposed NAST and S_T is the local statistics defined as

$$S_T(i,j,t) = \left| \left(x(i,j,t) - \overline{x}(i,j,t) \right)^2 - \sigma^2(i,j,t) \right| \tag{4}$$

3. A New Method for Local Feature Detector Using 2D Discrete Wavelet Transform

In this section 2D discrete wavelet transform is briefly described as a theoretical background (Daubechies, 1998). Based on theory and implementation of 2D discrete wavelet transform, the DoW-based local extrema detection method is presented.

3.1 Characteristics of 2D Wavelet Transform

Human visual characteristics are widely used in image processing. One example is the use of Laplacian pyramid for image coding. SIFT falls into the category that uses Laplacian pyamid for scale-invariant feature extraction [3]. On the other hand wavelet transform is a multiresolution transform that repeatedly decompose the input signal into lowpass and highpass components like subband coding [7,8]. Wavelet-based scale-invariant feature extraction method does not increase the number of samples in the original image, which is the case of the Gaussian pyramid-based SIFT method. Wavelet transform can easily reflect

human visual system by multiresolution analysis using orthogonal bases[12]. Because the wavelet-based method does not increase the number of samples, computational redundancy is greatly reduced, and its implementation is suitable for parallel processing.

3.2 Difference of Wavelet in the Scale Space

Most popular wavelet functions include Daubechies [7] and biorthogonal wavelet [10]. Although Daubechies designed a perfect reconstruction wavelet filter, it does not have symmetry. In general image processing applications symmetric biorthogonal filter is particularly suitable[10], but we used Daubechies coefficient set{DB2, DB10, DB18, DB26, DB34, DB42} for just efficient feature extraction purpose.

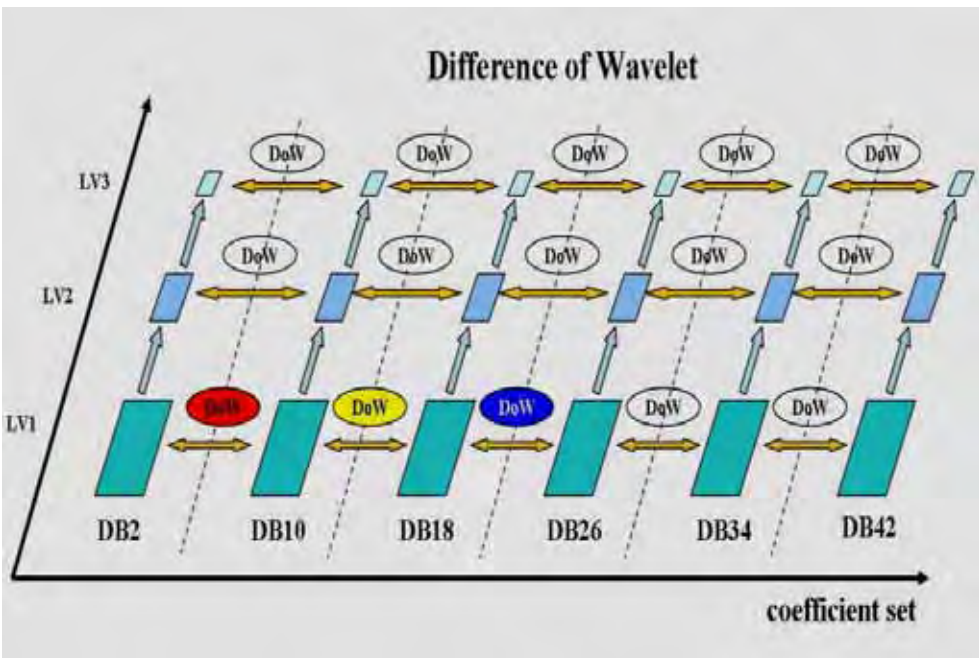


Figure 3. Structure of Difference of Wavelet

A. Parameter Decision for Wavelet Pyramid

In order to construct the wavelet pyramid, we decide the number of Daubechies coefficients and approximation levels, which can be considered as a counterpart of the DoG-based scale expansion. Fig. 4 shows that DB6 provides the optimum local key points, and Fig. 5 shows that approximation level 3 is the most efficient for matching. Although larger coefficients have better decomposition ability, we used DB2 as the first filter, and increased the step by 8. Because all DB filters have even numbered supports, difference between adjacent DB filters' support is recommended to be larger than or equal to 4 for easy alignment. In this work we used difference of 8, because difference of 4 provides almost same filtered images.

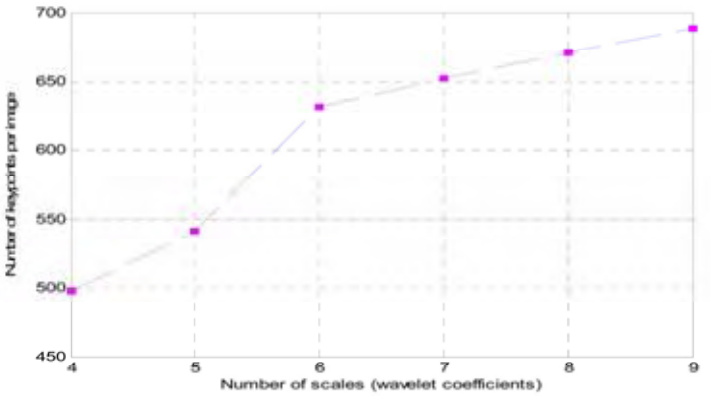


Figure 4. The number of extracted keypoints versus the number of wavelet coefficients

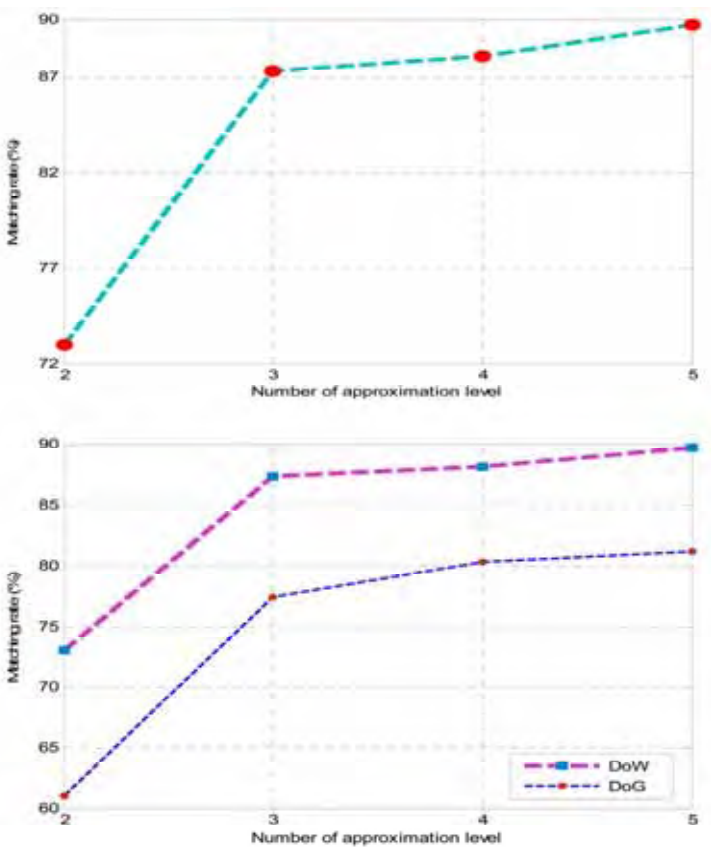


Figure 5. Matching rate versus the number of approximation level

Table 1 summarizes results experimental of processing time and matching rate using different wavelet filters in the SIFT framework. Coefficient set of the first row provides the best keypoint extraction result with significantly reduced computational overhead. The combination given in the second row is the best in the sense of matching time and rate.

Coefficient set \ Comparison factor	Processing time(msec)	Matching rate(%)
DB2, DB6, DB10, DB14, DB18, DB22	121	34.72
DB2, DB10, DB18, DB26, DB34, DB42	130	71.92
DB2, DB14, DB26, DB38, DB50, DB62	173	72.37
DB2, DB18, DB34, DB50, DB68, DB86	213	72.87
SIFT[4] (σ=1.6, k= $\sqrt{2}$, 1D Gaussian kernel size = 11) (Images per octave = 6, Number of octaves = 3)	925	57.68

Table 1. Various coefficient sets of Daubechies coefficients in the SIFT framework for measuring processing time and matching rate under low light(0.05lux) condition.

B. Wavelet-like Subband Transform

As shown in Fig. 3, the proposed wavelet pyramid is constructed using six Daubecies coefficient sets with three approximation levels. Because the length of each filter is even number, we need appropriate alignment method for matching different scales, as shown in Fig. 6, where DB10 is used for 320×240 input images.

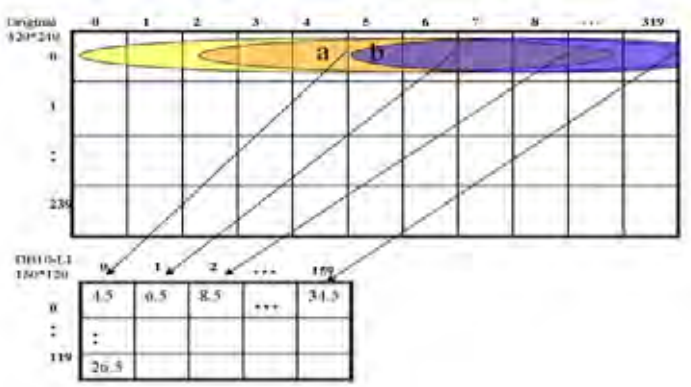


Figure 6. Proposed alignment method for different approximation levels

3.3 Local Extrema Detection and Local Image Descriptors

In the previous subsection we described the detail construction method for wavelet pyramid and DoW. In keypoints extraction step, we used min-max extrema with consideration of aligning asymmetrically filtered scales. In order to extract scale-invariant feature points, we compute DoW in the scale space, and locate the minimum and maximum pixels among the neighboring 8 pixels and 18 pixels in the upper and lower-scale images. Such extrema become scale-invariant features. DoWbased scale space is constructed as shown in Fig. 7.

For each octave of scale space, the initial images are repeatedly convolved with the corresponding wavelet filter to produce the set of scale space images shown in the left. DoW images are shown in the center, and in the right maxima and minima of the difference of wavelet images are detected by comparing a pixel, marked with ×, to its 26 neighbors in three 3 ×3 templates, marked with circle. For discrete wavelet transform, we used six different sets of Daubechies coefficients to generate a single octave, and make each difference image by using three octaves as

$$DoW_1 = DB10_L1 - DB2_L1, \quad DoW_2 = DB18_L1 - DB10_L1$$
$$DoW_3 = DB26_L1 - DB18_L1, \quad DoW_4 = DB34_L1 - DB26_L1$$
$$DoW_5 = DB42_L1 - DB34_L1 \tag{5}$$

Equation (5) defines how to make a DoW image using two wavelet transformed images. Feature points obtained by the proposed method are mainly located in the neighborhood of strong edges. DoW also has computational advantage to DoG because many octaves can be generated in parallel.

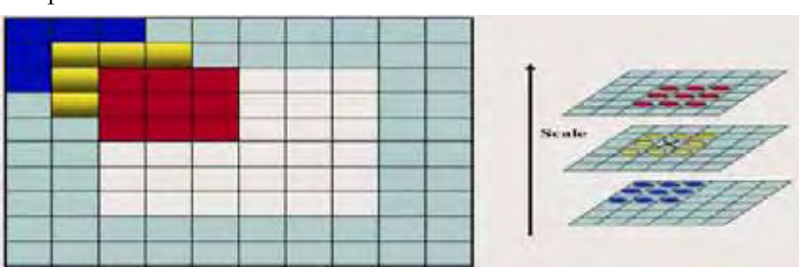


Figure 7. Maxima and minima of the difference of Wavelet images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3×3regions at the current and adjacent scales (marked with circles)

4. Experimental Result

We first enhanced a low light - level image using the proposed NAST filter, as shown in Fig.8.

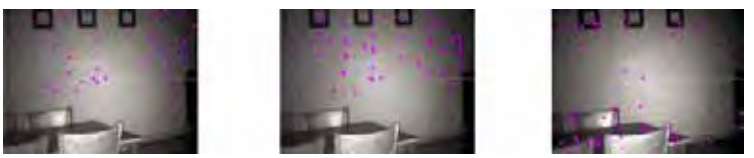
(a) (b)

Figure 8. (a) Input low light-level image with significant noise and (b) NAST filtered image

Comparison between DoG-based SIFT and the proposed DoW methods is shown in Fig. 9.As shown in Fig. 8, the proposed DoW method outperforms the DoG-based SIFT in the sense of both stability of extracted keypoints and computational efficiency.Fig. 10, Compares performance of combined NAST and DoG method with the DoG-based SIFT algorithm.

(a) (b) (c) (d)

Figure 9. Keypoint extraction results: (a) DoG, (b) DoW, and (c, d) translation of (a) and (b), respectively



(a) (b) (c)

Figure 10. Keypoints extraction results under low light-level condition using DoG, (b) DoG with NAST, and (c) DoW with NAST

Table 2 shows performance evaluation for processing time, matching rate and the PSNR in dB is obtained by using pre-filtering algorithm. The low pass filter(LPF)[13] were simulated for comparison with the NAST filter. In order to measure PSNR, we add synthetic noise (20dB PCN, and 15dB FCN) to the acquired low light images. This work was tested using a personal computer with Pentium- 3.0GHz.

Comparison Factor / Type of method	Processing time (msec)	PSNR(dB)	Matching rate (%)
DoG under low light	925	-	68.88
NAST + DoG under low light	1,104	39.48	70.98
LPF + DoW under low light	254	37.13	73.69
NAST + DoW under low light	355	39.50	77.24

Table. 2. Performance evaluation of DoG and DoW with NAST filter

5. Conclusion

The paper presents a local feature detection method for vSLAM-based self-localization of mobile robots. Extraction of strong feature points enables accurate self-localization under various conditions. We first proposed NAST pre-processing filter to enhance low light-level input images. The SIFT algorithm was modified by adopting wavelet transform instead of Gaussian pyramid construction. The wavelet-based pyramid outperformed the original SIFT in the sense of processing time and quality of extracted keypoints. A more efficient local feature detector and a compensation scheme of noise due to the low contrast images are also proposed. The proposed scene recognition method is robust against scale, rotation, and noise in the local feature space.

6. References

Dissanayake, M.W.M.G. et al. (2001) A solution to the simultaneous localization and map building(SLAM)problem, Robotics and Automation, IEEE Transactions, Vol. 17, Issue 3, pp. 229-241, June 2001

Lowe, D. G. (2004) Distinctive image features from scale invariant keypoints, Int'l Conf. on Computer Vision, vol. 60, No. 2, pp. 91-110, 2004

Lee, S.; Maik, V.; Jang, J.; Shin, S. & Paik, J. (2005) Noise-Adaptive Spatio-Temporal filter for real-time noise removal in low light images, IEEE Trans. Consumer Electronics, Vol. 51, No. 2, pp.648-653, May 2005

Daubechies, I. (1998) Orthogonal bases of compactly supported wavelets, Comun, Pure. Appl. Math, Vol. 41, pp. 909-996, Nov 1998 Irie, K. & Kishimoto, R. (1991) A study on perfect reconstructive subband coding, IEEE Trans. On CAS for Video Technology, Vol. 1, no. 1, pp. 42-48, March 1991

21

Genetic Algorithms: Basic Ideas, Variants and Analysis

Sharapov R.R.
Institute of Mathematics and Mechanics (IMM)
Russia

1. Introduction

Genetic algorithms are wide class of global optimization methods. As well as neural networks and simulated annealing, genetic algorithms are an example of successful using of interdisciplinary approach in mathematics and computer science. Genetic algorithm simulates natural selection and evolution process, which are well studied in biology. In most cases, however, genetic algorithms are nothing else than probabilistic methods, which are based on principles of evolution. The idea of genetic algorithm appears first in 1967 in J. D. Bagley's thesis (Bagley, 1967). The theory and applicability was then strongly influenced by J. H. Holland, who can be considered as the pioneer of genetic algorithms (Holland, 1992). Since then, this field has witnessed a tremendous development.

There are many applications where genetic algorithms are used. Wide spectrum of problems from various branches of knowledge can be considered as optimization problem. This problem appears in economics and finances, cybernetics and process control, game theory, pattern recognition and image analysis, cluster analysis etc. Also genetic algorithm can be adapted for multicriterion optimization task for Pareto-optimal solutions search. But most popular applications of genetic algorithm are still neural networks learning and fuzzy knowledge base generation.

There are three ways in using genetic algorithms with neural networks:

1. Weight learning. Optimal net weights are found with genetic algorithm when conventional methods (e.g. backpropagation) are not applicable. It is suitable when continuous activation function of neuron (such as sigmoid) is used, so error function become multiextremal and conventional method can find only local minimum.
2. Architecture optimization. Genetic algorithm is used for finding optimal net architecture from some parameterized class of net architectures.
3. Learning procedure optimization. In this expensive but effective method genetic algorithm is used for finding optimization parameters of learning function (weight correction function). Usually this method is used with architecture optimization simultaneously.

Genetic fuzzy systems are other popular application of genetic algorithms. Fuzzy system design consists of several subtasks: rule base generation, tuning of membership function and tuning of scaling function. All this tasks can be considered as optimization problem, so genetic algorithm is applicable (Cordon et al., 2004).

The optimization problem solved by genetic algorithms in general can be formulated as:

$$\max_{x \in X} f(x) \tag{1}$$

where X is search space, objective function f is total function in X, f: $X \to \mathbb{R}$. Some particular cases of this problem are well studied and solution methods are well known. For instance it is mentioned linear and convex programming problem. In general, however, this problem is very complex and non-solvable. It means that solution cannot be obtained in finite iteration steps.

We restrict problem (1) and consider case of compact and simple structure of set X, e.g. X is hypercube and it is known that f reach maximum inside X. In this case complexity of optimization task is depended from complexity of objective function f only. In common case f is non-smooth (non-differentiable) multiextremal function. Even through f is differentiable and conventional optimization methods e.g. gradient descent are applicable there are no guarantee that global optimum will be found.

There are two wide classes of optimization methods to solve global optimization problem: deterministic and stochastic. First obtain solution via almost complete search all over the X, so these methods are slow and non-efficient, but guarantee optimum finding. Also using of these methods requires some restrictions on objective function, so in several cases deterministic methods are not applicable. Second class is stochastic methods, which are faster and more efficient and universal than deterministic but has one essential shortcoming: maximization of objective function is not guarantee. Most of stochastic algorithms evaluate objective function in some random points of search space. Then sample of these points is processed and some pointes are saved for the next iteration.

As the practice shows in many instances it is acceptable to find not best but just well solution, so stochastic methods and genetic algorithms particularly are very effective.

2. Basic Ideas and Concepts

We consider optimization problem (1). Genetic algorithm does not work with problem (1) directly, but with coded version of them. Search space X is mapped into set of string S. Function $c(x)$: $X \to S$ is called coding function. Conversely, function $d(s)$: $S \to X$ is called decoding function and $c \circ d(s) = s$ should be done for any string s. In practice, coding functions, which have to be specified depending on the needs of the actual problem, are not necessarily bijective, so $d \circ c$ is not identical map over X, but it is over $D = d(S)$.

Usually, S is finite set of binary strings:

$$S = \{0,1\}^m \tag{2}$$

where m is obviously length of string. Generally simple binary code or Gray code is used. Note that S is finite, but X is commonly not. So, we quantize search space and algorithm finds solution approximately, but solution precision can be made as high as needed by increasing m.

Thus, we replace problem (1) with follows:

$$\max_{s \in S} f(s) \tag{3}$$

where under *f(s)* we imply *f(d(s))*.
Terminology particularly borrowed from natural genetic and evolution theory is commonly used in framework of genetic algorithms. Below we give some of most often used terms.
Member of set *S* is called *individual*. Individual in genetic algorithm is identified with *chromosome*. Information encoded in chromosome is called *genotype*. *Phenotype* is values of source task variables corresponding to genotype. In other words phenotype is decoded genotype. In simple genetic algorithm chromosomes are binary string of finite length. *Gene* is a bit of this string. *Allele* is value of gene, 0 or 1. *Population* is finite set of individuals. Objective function of optimization problem is called fitness *function*.
Fitness of individual is value of fitness function on phenotype corresponding individual. *Fitness of population* is aggregative characteristic of fitness of individuals. Fitness of best individual or average fitness of individuals is commonly used as population fitness in genetic algorithms.
In process of evolution one population is replaced by another and so on, thus we select individuals with best fitness. So in the mean each next generation (population) is fitter than it predecessors. Genetic algorithm produces maximal fitness population, so it solve maximization problem. Minimization problem obviously reduced to maximization problem.
In simple genetic algorithm size of population *n* and binary string length *m* is fixed and don't changes in process of evolution. We can write basic structure of simple genetic algorithm in the following way:

```
Compute initial population;
WHILE stopping condition not fulfilled DO BEGIN
        select individuals for reproduction;
        create offsprings by crossing individuals;
        eventually mutate some individuals;
        compute new generation;
END
```

As obvious from the above stated algorithm, the transition from one generation to the next consists of three basic components:
Selection: Mechanism for selecting individuals for reproduction according to their fitness.
Crossover: Method of merging the genetic information of two individuals. In many respects the effectiveness of crossover is depended on coding.
Mutation: In real evolution, the genetic material can by changed randomly by erroneous reproduction or other deformations of genes, e.g. by gamma radiation. In genetic algorithms, mutation realized as a random deformation of binary strings with a certain probability.
These components are called genetic operators. We consider these operators more detailed below.
Compared with conventional continuous optimization methods, such as gradient descent methods, we can state the following significant differences:

1. Genetic algorithms manipulate coded versions of the problem parameters instead of the parameters themselves, i.e. the search space is *S* instead of *X* itself. So, genetic algorithm finds solution approximately.
2. 2. While almost all conventional methods search from a single point, genetic algorithm always operates on a whole population of points (strings-individuals). It improves

robustness of algorithm and reduces the risk of becoming trapped in a local stationary point.

3. Normal genetic algorithms do not use any auxiliary information about the objective function value such as derivatives. Therefore, they can be applied to any kind of continuous or discrete optimization problem.
4. Genetic algorithms use probabilistic transition operators while conventional methods for continuous optimization apply deterministic transition operators. More specifically, the way a new generation is computed from the actual one has some random.

3. Simple genetic algorithm

Here we consider simpler genetic algorithm in more detail. As previously noted let m is binary string space dimension, n is population size. The generation at time t is a list of n binary strings, which we will denote with

$$B_t = (b_{1,t}, b_{2,t}, ..., b_{n,t}) \tag{4}$$

Stated above basic structure of genetic algorithm can be written more detailed in the following way:

t := 0;
Compute initial population B_0;
WHILE stopping condition not fulfilled DO BEGIN
    FOR i:=1 TO n DO
        select $b_{i,t+1}$ from B_t
    FOR i:=1 TO n STEP 2 DO
        with probability p_c perform crossover of $b_{i,t+1}$ and $b_{i+1,t+1}$
    FOR i:=1 TO n DO
        with probability p_m eventually mutate $b_{i,t+1}$
    t:=t+1;
END

We don't give concrete expression for stopping condition because these conditions have no features in comparison with other global optimization methods. So, as such conditions we can take restriction on number of iterations or some phenotype convergence conditions. Last can be formulated in terms of maximal or average fitness.

Commonly used procedure to compute initial population consist in random selection of n points uniformly distributed over the search space. If additional information about decision region is presented, it can be used for initial population computation.

3.1 Selection

Selection is the component which guides the algorithm to the solution by preferring individuals with high fitness over low-fitted ones. It realizes "The fittest will survive" principle. Selection can be a deterministic operation, but in most implementations it has random components.

One variant, which is very popular nowadays, is the following scheme, where the probability to choose a certain individual is proportional to its fitness. It can be regarded as a random experiment with

$$P\{b_{i,t}\} = p_{i,t} = \frac{f(b_{i,t})}{\sum_{k=1}^{n} f(b_{k,t})} \tag{5}$$

Of course, this formula only makes sense if all the fitness values are positive. If this is not the case, a increasing transformation φ: $\mathbb{R} \rightarrow \mathbb{R}^+$ must be applied. In simple case shift $\varphi = x+M$ can be used, where M is sufficiently great. M is chosen based upon some information about fitness function. If there no such information other transformations must be applied, such as exponential $\varphi = a^x$ or shifted arctangent $\varphi = arctan(x)+\pi/2$. Then the probabilities can be expressed as

$$P\{b_{i,t}\} = p_{i,t} = \frac{\varphi(f(b_{i,t}))}{\sum_{k=1}^{n} \varphi(f(b_{k,t}))} \tag{6}$$

Everywhere below we suppose that function f is positive.
We can force the property (5) to be satisfied by applying a random experiment which is, in some sense, a generalized roulette game. In this roulette game, the slots are not equally wide, i.e. the different outcomes can occur with different probabilities. Figure 1 gives a graphical interpretation of this roulette wheel game.

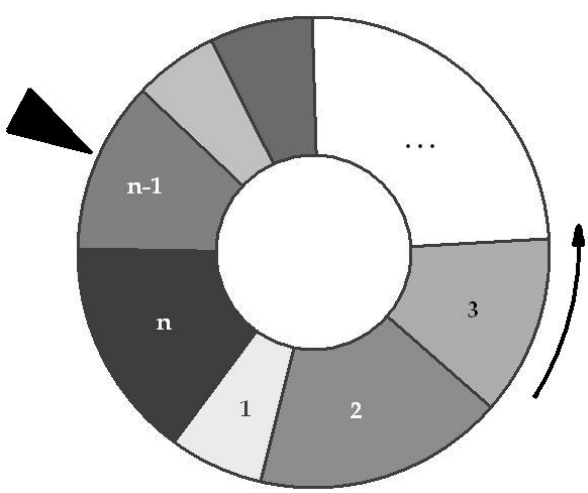


Fig. 1. A graphical representation of roulette wheel selection

For obvious reasons, this method is often called proportional selection. Mean of copies of individual $b_{i,t}$ which will be selected for follows crossover can be expressed as

$$\mathrm{E}(number\ of\ copies\ b_{i,t}) = p_{i,t} n \tag{7}$$

It is easy to see that ill-fitted individuals have slim chance to leave offsprings, so they leave population very early. In some cases, this can be the cause of premature convergence of algorithm into local maxima. On the other hand, refinement in the end phase can be slow

since the individuals have similar fitness values. These problems can be overcome by using alternative selection schemes:

Linear rank selection. Rank of the fitness as the basis of selection is used instead of the values themselves.

Tournament selection. In this scheme, a small group of individuals is sampled from the population and the individual with best fitness is chosen for reproduction. This selection scheme is also applicable when the fitness function is given in implicit form, i.e. when we only have a comparison relation which determines which of two given individuals is better.

3.2 Crossover

In sexual reproduction, as it appears in the real world, the genetic material of the two parents is mixed when the gametes of the parents merge. Usually, chromosomes are randomly split and merged, with the consequence that some genes of a child come from one parent while others come from the other parents.

This mechanism is called crossover. It is a very powerful tool for introducing new genetic material and maintaining genetic diversity, but with the outstanding property that good parents also produce well-performing children or even better ones.

Basically, crossover is the exchange of genes between the chromosomes of the two parents. In the simplest case, this process in genetic algorithms is realized by cutting two strings at a randomly chosen position (crossing point) and swapping the two tails. This process, which called one-point crossover, is visualized in Figure 2. In genetic algorithm selected individuals paired in some way and then crossing over with probability p_c.

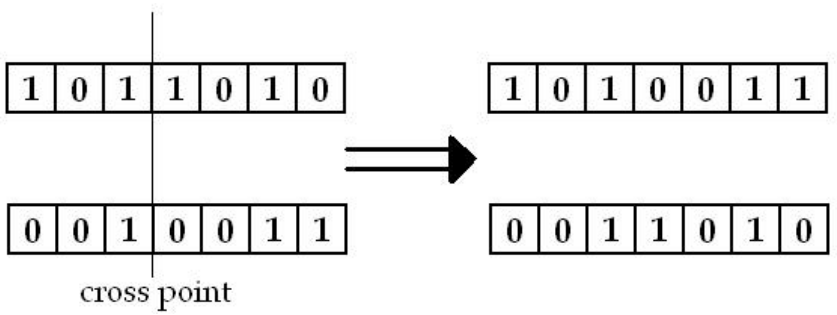


Fig. 2. One-point crossover of binary strings

One-point crossover is a simple and often-used method for genetic algorithms which operate on binary strings. For other problems or different coding function, other crossover methods can be useful or even necessary. We mention some of them, for more details see (Goldberg, 1989).

N-point crossover. Instead of only one, N breaking points are chosen randomly. Every second section is swapped. Among this class, two-point crossover is particularly important.

Segmented crossover. Similar to N-point crossover with the difference that the number of breaking points can vary.

Uniform crossover. For each position, it is decided randomly if the positions are swapped.

Shuffle crossover. First a randomly chosen permutation is applied to the two parents, then N-point crossover is applied to the shuffled parents, finally, the shuffled children are transformed back with the inverse permutation.

3.3 Mutation

Mutation is powerful factor of variability and consists in random deformation of genetic material. In real world these deformations take place as result of radioactivity, ultraviolet radiation or viruses influence. In real reproduction, the probability that a certain gene is mutated is almost equal for all genes. Mutation in genetic algorithm is analogue of natural one: each gene of chromosome is inverted with probability p_m, so this mutation is called uniform mutation. Also, in genetic algorithms alternative mutation methods are used. We mention some of them, more detailed see (Goldberg, 1989).

Inversion of single bits. With probability p_m, one randomly chosen bit is negated.

Bitwise inversion. The whole string is inverted bit by bit with probability p_m.

Random mutation. With probability p_m, the string is replaced by a randomly chosen one.

4. Variants

We consider simple variant of genetic algorithm, but it is sufficiently effective. Thus, there are some ways to improve efficiency and robustness. In this section we consider some of this ways.

Elitism is very effective element that realizes "best must survive" principle. It can be added into any selection scheme and consists in follows: best individual from parent population is compared with best individual from offspring population and best of them is added into next generation. Elitism guarantees that next generation fitness will be better or equal than parent generation fitness. Elitism is often-used element, but it should, however, be used with caution, because it can lead to premature convergence.

Adaptive genetic algorithms are algorithms whose parameters, such as the population size, the crossing over probability, or the mutation probability are varied while the genetic algorithm is running. A simple variant could be the following: The mutation rate is changed according to changes in the population; the longer the population does not improve, the higher the mutation rate is chosen. Vice versa, it is decreased again as soon as an improvement of the population occurs.

Hybrid genetic algorithms are used when additional auxiliary information such as derivatives or other specific knowledge is known about objective function. So, conventional method, such as gradient descent is applicable. The basic idea is to divide the optimization task into two complementary parts. The coarse, global optimization is done by the genetic algorithm while local refinement is done by the conventional method. A number of variants is reasonable:

1. The genetic algorithm performs coarse search first. After it is completed, local refinement is done.
2. The local method is integrated in the genetic algorithm. For instance, every k generations, the population is doped with a locally optimal individual.
3. Both methods run in parallel: All individuals are continuously used as initial values for the local method. The locally optimized individuals are re-implanted into the current generation.

In *self-organizations genetic algorithms* not only data is object of evolution. Parameters of genetic algorithm, such as coding function or genetic operator parameters, are optimized too. If this is done properly, the genetic algorithm could find its own optimal way for representing and manipulating data automatically.

5. Analysis

As stated above, genetic algorithm is stochastic optimization method and not guarantees convergence to solution. Therefore, we consider convergence in terms of mean. Convergence analysis becomes complicated by using three stochastic operators: selection, crossover and mutation that have many variations, so there are many different algorithms. We consider simple genetic algorithm with fixed population size n operates in space of binary string with fixed length m. It is assumed that one-point crossover, uniform mutation and proportional selection are used.

5.1 The Schema Theorem

Analysis of genetic algorithm we start from classic result of Holland – the so-called Schema Theorem. But at first we'll make some definitions.

Definition 1. A string $H = h_1 \ldots h_m$ over the alphabet {0, 1, *} is called a (binary) *schema* of length m. An $h_i = 0$ or 1 is called a *specification* of H, an $h_i = *$ is called *wildcard*. Schemata can be considered as specific subsets of $\{0, 1\}^m$.

If we interpret binary strings space as hypercube with dimension m, then schemata can be interpreted as hyperplanes (see Figure 3).

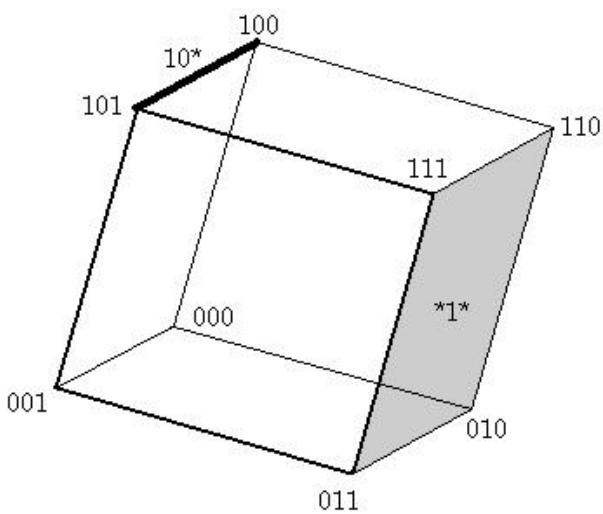


Fig. 3. Schemata as hyperplanes in hypercube

Obviously number of schemata is 3^m.

Definition 2. A string $S = s_1 \ldots s_m$ over the alphabet {0, 1} *fulfills* the schema $H = h_1 \ldots h_m$ if and only if it matches H is all non-wildcard positions:

$$\forall i \in \{j \mid h_j \neq *\} : s_i = h_i \tag{8}$$

Definition 3. The number of specifications of a schema H is called *order* and denoted as

$$O(H) = |\{i \mid 1 \leq i \leq m, h_i \neq *\}| \tag{9}$$

Definition 4. The distance between the first and the last specification

$$\delta(H) = \max\{i \mid h_i \neq *\} - \min\{i \mid h_i \neq *\} \tag{10}$$

is called the *defining length* of a schema H.
Also let us make some notations:
The number of individuals which fulfill H at time step t are denoted as $r_{H,t}$.
The observed average fitness at time t is denoted as:

$$\overline{f}(t) = \frac{1}{n}\sum_{k=1}^{n} f(b_{k,t}) \tag{11}$$

The observed average fitness of schema H in time step t is denoted as:

$$\overline{f}(H,t) = \frac{1}{r_{H,t}} \sum\{f(b_{i,t}) \mid b_{i,t}\, fulfill\ H\} \tag{12}$$

The following theorem holds.
Theorem (Schema Theorem—Holland 1975).
The following inequality holds for every schema H:

$$\mathrm{E}(r_{H,t+1}) \geq r_{H,t}\,\frac{\overline{f}(H,t)}{\overline{f}(t)}(1 - p_c\,\frac{\delta(H)}{m-1})(1-p_m)^{O(H)} \tag{13}$$

where E is mean of number of next generation individuals fulfills schema H. More generally statement of schema theorem can be formulated as follows:

$$\mathrm{E}(r_{H,t+1}) \geq r_{H,t}\,\frac{\overline{f}(H,t)}{\overline{f}(t)}\,P_c(H)\,P_m(H) \tag{14}$$

where estimations P_c and P_m depend only from schema H on one hand and crossover and mutation methods correspondingly on another. Such estimations can be obtained for all considered variants of crossover and mutation operators. One can see (Holland, 1992) for full proof of schema theorem.
The schema theorem answer the question what schemata has more chance to survive, but say nothing about convergence in essence.

5.2 Building blocks hypothesis

As obviously follow from schema theorem high-fitness schemata with low order and short length have more chance to survive in process of evolution. Let p_m is sufficiently small, then (13) takes the form:

$$\mathrm{E}(r_{H,t+1}) \geq r_{H,t}\,\frac{\overline{f}(H,t)}{\overline{f}(t)}(1 - p_c\,\frac{\delta(H)}{m-1})(1-p_m O(H)) \tag{15}$$

If population size is sufficiently great, then deviations from average $\mathrm{E}(r_{H,t+1})$ are very small. If we disregard them follows statement take place:

$$r_{H,t+1} \geq r_{H,t}\,\frac{\overline{f}(H,t)}{\overline{f}(t)}(1 - p_c\,\frac{\delta(H)}{m-1})(1-p_m O(H)) \tag{16}$$

It is obviously follows from this recurrent expression that number of individuals fulfills high-fitness schemata with low $\delta(H)$ and $O(H)$ exponentially grows in process of evolution. Such schemata, i.e. well-fitted schemata with short length and low order, are called *building blocks*. Goldberg conjecture follows: *A genetic algorithm creates stepwise better solutions by recombining, crossing, and mutating short, low-order, high-fitness schemata.* This conjecture is called *building blocks hypothesis* (Goldberg, 1989).
If building blocks hypothesis is true, key role for convergence play coding method. Coding must be realized building blocks hypothesis concept. For example consider two examples of fitness function. First is an affine linear fitness function:

$$f(s) = a + \sum_{i=1}^{m} c_i s_i \tag{17}$$

where s_i is ith allele of chromosome s.
Second function correspond "needle-in-haystack" problem:

$$f(x) = \begin{cases} 1, x = x \\ 0, x \neq x_0 \end{cases} \tag{18}$$

In the linear case, the building block hypothesis seems justified, i.e. the fitness is computed as a linear combination of all genes. It is easy to see that the optimal value can be determined for every gene independently. For the second function, however, it cannot be true, since there is absolutely no information available which could guide a genetic algorithm to the global solution through partial, sub-optimal solutions. In other words, the more the positions can be judged independently, the easier it is for a genetic algorithm. On the other hand, the more positions are coupled, the more difficult it is for a genetic algorithm (and for any other optimization method). There is a special term derived from biology for this phenomena – *epistasis*. High *epistatic* problem are very difficult to solve. Genetic algorithms are appropriate for medium *epistatic* problems, and low *epistatic* problem can be solved much more efficiently with conventional methods.
Follow question may arise after analysis: what a genetic algorithm really processes, strings or schemata? The answer is both. Nowadays, the common interpretation is that a genetic algorithm processes an enormous amount of schemata implicitly and simultaneously. This is accomplished by exploiting the currently available, incomplete information about these schemata continuously, while trying to explore more information about them and other, possibly better schemata.

5.3 The Convergence Theorem

The Schema Theorem clarifies some aspects of the mechanism how genetic algorithm works. Building blocks hypothesis conjecture some assumption about convergence, but it isn't proven. Some results about convergence were obtained by author. Although genetic algorithm not guarantees solution finding, it converge in the mean. Below we formulate Theorem of Convergence of genetic algorithms.
As stated above, we consider simple genetic algorithm with fixed population size n operates in space of binary string with fixed length m. It is assumed that one-point crossover with probability p_c, uniform mutation with probability p_m and proportional selection are used.

Also we assume that elitism is incorporated in selection procedure, so best individual always survive. Hence, the following theorem holds:

Theorem.

Let $p_m \leq 0.5$, and $S = (1 - p_m^m)\ (2 - (1 - p_c)^n) < 1$.

Then,

$$\lim_{k \to \infty} \mathrm{E}(f(B_k)) = f^* \tag{19}$$

where B_k is the population after the kth iteration step of the genetic algorithm, $f(B_k)$ is the maximal fitness over the population B_k, and f^* is the required optimal value. $\mathrm{E}(f(B_k))$converges to f^* non-decreasingly. Proof of this theorem can be found in (Sharapov & Lapshin, 2006).

There is an interesting corollary corresponding case of zero p_c.

Corollary. Let $p_m \leq 0.5$, $p_c = 0$.

Then,

$$\lim_{k \to \infty} \mathrm{E}(f(B_k)) = f^* \tag{20}$$

where B_k is the population after the kth iteration step of the genetic algorithm, $f(B_k)$ is the maximal fitness over the population B_k, and f^* is the required optimal value and $C_m^{[m/2]}$ is binomial coefficient. $\mathrm{E}(f(B_k))$converges to f^* non-decreasingly. Evidently, crossover absence gives us everywhere convergent algorithm.

6. Real-coded evolutionary optimization methods

Most of optimization problems have real-valued parameters (i.e. X is subset of $\mathbb{R}^N$, where N is problem dimension). It is clear that discretization approach applied in simple genetic algorithm has several shortcomings:

1. Continuum set of possible values is reduced to finite set of binary strings. So we limit considered search space, and if solution of task is located outside considered region, we will not find it.
2. The accuracy of the solution is limited by the discretization width $1/(2m-1)$, where m is length of binary string. Although precision can be improved by increasing m, it will require more computer power and time. Computational complexity grows exponentially with m growth.
3. It is complicated to choose appropriate coding method. Most often, no reasonable building blocks exist.

For these reasons, variants of genetic algorithms which are especially adapted to real-valued optimization problems have been proposed.

6.1 Real-coded genetic algorithms

Structure of real-coded genetic algorithm is not to differ from one considered in section 3. But chromosomes in real-coded genetic algorithms are represented as N-dimensional vectors of real numbers, where N is dimension of optimization problem:

$$b = (x_1, \ldots, x_N) \tag{21}$$

All selection schemes are applicable without any modifications. Crossover and mutation must be adapted.
In real-coded genetic algorithms follows crossover operators are used most-often:
Flat crossover. Two parents $b_1 = (x_{1,1}, \ldots, x_{1,N})$ and $b_2 = (x_{2,1}, \ldots, x_{2,N})$ are given, a vector of random values from the unit interval $\lambda = (\lambda_1, \ldots, \lambda_N)$ is chosen. The offspring $b' = (x'_1, \ldots, x'_N)$ is computed as a vector of linear combinations in the following way (for all $i = 1, \ldots, N$):

$$b'_i = \lambda_i x_{1,i} + (1 - \lambda_i) x_{2,i} \tag{22}$$

Second offspring from pair is computed analogously.
BLX-α crossover (Herrera et al., 1998) is an extension of flat crossover which allows an offspring allele x'_i to be also located outside the interval $[\min(x_{1,i}, x_{2,i}), \max(x_{1,i}, x_{2,i})]$. In BLX-α crossover, each offspring allele x'_i is chosen as a uniformly distributed random value from the interval

$$[\min(x_{1,i}, x_{2,i}) - I \cdot a, \max(x_{1,i}, x_{2,i}) + I \cdot a] \tag{23}$$

where $I = \max(x_{1,i}, x_{2,i}) - \min(x_{1,i}, x_{2,i})$.
The parameter α has to be chosen in advance. For α = 0, BLX-α crossover becomes identical to flat crossover.
Simple and discrete crossover is analogous to considered above classical one-point and uniform crossover.
The following mutation operators are most common for real-coded genetic algorithms:
Random mutation. For a randomly chosen gene i of an individual $b = (x_1, \ldots, x_N)$, the allele x_i is replaced by a randomly chosen value from a predefined interval $[a_i, b_i]$.
Non-uniform mutation. In non-uniform mutation, the possible impact of mutation decreases with the number of generations (Michalewicz, 1996). Assume that $t_{\max}$ is the predefined maximum number of generations. Then, with the same setup as in random mutation, the allele x_i is replaced by one of the two values

$$\begin{aligned} x'_i &= x_i + \Delta(t, b_i - x_i) \\ x''_i &= x_i - \Delta(t, x_i - a_i) \end{aligned} \tag{24}$$

The choice which of the two is taken is determined by a random experiment with two outcomes that have equal probabilities 0.5 and 0.5. The random variable $\Delta(t, x)$ determines a mutation step from the range $[0, x]$ in the following way:

$$\Delta(t, x) = x \cdot (1 - \lambda^{(1 - t/t_{\max})^r}) \tag{25}$$

In this formula, λ is a uniformly distributed random value from the unit interval. The parameter r determines the influence of the generation index t on the distribution of mutation step sizes over the interval $[0, x]$.

6.2 Evolutionary strategies

Evolutionary strategies are real-coded global optimization methods were developed in late 1960s mainly by I. Rechenberg independently from Holland's work on genetic algorithms. Chromosome in evolutionary strategies is represented by $2N$ dimensional vector, where N is dimension of problem:

$$b = (x_1, ..., x_N; \sigma_1, ..., \sigma_N) \tag{26}$$

The first half (x_1, …, x_N) corresponds to the potential solution of the optimization problem like in real-coded genetic algorithms. The second half (σ_1, …, σ_N) defines the vector of standard deviations for the mutation operation.

As usual, there are two means of modifying genetic material in evolutionary strategies: a recombination operation that could be understood as some kind of crossover and mutation. Unlike genetic algorithms, mutation plays a more central role in evolutionary strategies. Usually as recombination operator flat or discrete crossover applied in real-coded genetic algorithm (see previous section) are used. Most often in evolutionary strategies flat recombination with $\lambda = 0.5$ is used (so-called intermediate recombination).

Mutation in evolutionary strategies consists of two phases. Firstly, normal distributed noise is added to each allele x_i. More specifically, for all $i = 1, \ldots, N$, the mutated allele is given as

$$x'_i = x_i + N(0, \sigma_i^2) \tag{27}$$

where $N(0,\sigma^2_i)$ is normally distributed random variable with zero mean and standard deviation σ_i.

Secondly, we added logarithmically normal distributed noise to σ_i alleles:

$$\sigma'_i = \sigma_i \cdot \exp(\tau' N(0,1) + \tau N_i(0,1)) \tag{28}$$

The factor exp (τ' $N(0,1)$) is an overall factor increasing or decreasing the "mutability" of the individual under consideration. Note that $N(0,1)$ is chosen only once for the whole individual when it is mutated. The factor exp (τ $N_i(0,1)$) locally adapts the mutation step sizes. Note that, in this second factor, the normally distributed random value $N_i(0,1)$ is chosen separately for each gene. The adaptation of mutation step sizes in evolutionary strategies has the particular advantage that no parameters have to be chosen in advance. Instead, they evolve during the run of an evolutionary strategy in a self-organizing way.

The two parameters τ and τ' have to be chosen in advance. Schwefel has proposed to choose these parameters in the following way (Schwefel, 1995):

$$\tau' = \frac{1}{\sqrt{2N}}, \tau = \frac{1}{\sqrt{2\sqrt{N}}} \tag{29}$$

Selection in evolutionary strategies also has some features in comparison with genetic algorithms. The nowadays commonly accepted selection and sampling schemes in evolutionary strategies are the following:

(μ + λ)-strategy: a number of μ parents are selected from the current generation. These μ parents are used to generate a number of λ offsprings, which have been generated by some recombination and/or mutation operations. Out of the union of parents and offsprings (in total, a number of $\mu + \lambda$), the best μ are kept for the next generation. Note that the (μ + λ)-strategy inherently incorporates elitism.

(μ, λ)-strategy: in this scheme, which is nowadays considered the standard selection/sampling strategy in evolutionary strategies, again μ parents are selected from the current generation and used to generate λ offsprings (with the additional restriction $\lambda \geq \mu$). The parents are discarded completely and the best μ offsprings are kept for the next generation. The (μ, λ)-strategy does not incorporate elitism.

Note that both strategies only use the ranking of fitness values. Therefore, they can be applied both to minimization and maximization problems, without any need for scaling or transforming fitness values.

6.3 Evolutionary programming

Idea of evolutionary programming were proposed by L.J.Fogel in the middle of 1960s and later extended by his son D.B. Fogel (Fogel, 1992). Evolutionary programming solves same tasks in similar ways as real-coded genetic algorithms and evolutionary strategies. An important difference evolutionary programming from real-coded genetic algorithms and evolutionary strategies is consists in following: evolutionary programming does not use crossover or any other kind of exchange of genetic material between individuals. Offsprings are generated by mutation only.

We consider modified evolutionary programming method (Fogel, 1992). As well as evolutionary strategies, in this variant of evolutionary programming individual is represented by $2N$ dimensional vector of real values, where N is dimension of problem:

$$b = (x_1, ..., x_N; v_1, ..., v_N) \tag{30}$$

The second half of the vector (v_1, …, v_N) contains the variances of the mutation step sizes, as the mutation is done in the following way:

$$\begin{aligned} x'_i &= x_i + \sqrt{v_i} \cdot N_i(0,1) \\ v'_i &= v_i + \sqrt{\chi v_i} \cdot N_i(0,1) \end{aligned} \tag{31}$$

Unfortunately, it is not guaranteed that v'_i is positive. Therefore, additional measures have to be taken to avoid that v'_i gets 0 or negative. The parameter χ defines volatility of mutation factors v_i. $N_i(0,1)$ is a value of standard normally distributed random variable which is chosen separately for each gene.

Evolutionary programming uses a kind of combination of tournament and linear rank selection. The fitness of an individual b is compared with q other randomly picked competitors taken from the union of μ parents and λ offsprings. The score w_i of the individual b is computed as the number of individuals within the q selected ones that have a lower fitness than b. The parents and offsprings are ranked according to their score and the best μ are selected for the next generation. Note that this selection scheme inherently incorporates elitism. Moreover, for large q, it behaves almost in the same way as the (μ + λ)-strategy used in evolutionary strategies.

6.4 Analysis of convergence of real-coded methods

We will not investigate convergence detailed here and will make only some assertions about convergence properties.

For simplicity we consider the case of evolutionary programming only (see previous section), where N = 1. Also let $\lambda = \mu$ is used in selection scheme. Let $B_0 = (b_{0,1}, \ldots, b_{0,\mu})$ is initial population. Individuals in population are descending sorted by fitness, so first individual is best of all. After mutation we obtain μ offsprings b'_1, …, b'_μ. Each of them is two-dimensional normally distributed variate. Since genes mutate independently they are independent variates. Then consider aggregate population of parents and offsprings:

$$B' = (b_{0,1}, ..., b_{0,\mu}; b'_1, ..., b'_\mu) \tag{32}$$

Best individual from this population is kept for the next generation, because its rank is maximal among all of them. Let z – best among offsprings $b'_1 \ldots b'_\mu$. Obviously z is two-dimensional random variable. Then best individual $b_{1,1}$ of next generation B_1 is best of $b_{0,1}$ and z, i.e. $b_{1,1}$ = arg max $\{b_{0,1}, z\}$. So

$$f(B_1) = f(b_{1,1}) = \max\{f(b_{0,1}), f(z)\} \tag{33}$$

Assume, that $f(z)$ is absolutely continuous random variate. Let's consider mean of variate $f(b_{1,1})$ (here and below we suppose that all integrals are exist and converge absolutely):

$$\mathrm{E}f(B_1) = \mathrm{E}f(b_{1,1}) = \mathrm{E}\max\{f(b_{0,1}), f(z)\} = \int_{-\infty}^{+\infty} \max\{f(b_{0,1}), f(z)\} d(x)dx \tag{34}$$

where $d(x)$ is density function of variate $f(z)$. Transom (34):

$$\mathrm{E}f(B_1) = f(b_{0,1}) \int_{-\infty}^{f(b_{0,1})} d(x)dx + \int_{f(b_{0,1})}^{+\infty} xd(x)dx = f(b_{0,1}) + \int_{f(b_{0,1})}^{+\infty} (x - f(b_{0,1}))d(x)dx \tag{35}$$

Subintegral expression of second item is obviously non-negative. Therefore

$$\int_{f(b_{0,1})}^{+\infty} (x - f(b_{0,1}))d(x)dx \geq 0 \tag{36}$$

Consider the case of equality more detailed. Obviously equality is realized if and only if subintegral expression is identically zero, so $d(x) \equiv 0$ on interval $(f(b_{0,1}), +\infty)$. Hence probability

$$P\{f(z) > f(b_{0,1})\} = \int_{f(b_{0,1})}^{+\infty} d(x)dx = 0 \tag{37}$$

It means that improvement of fitness of population is impossible event. Since function f is continuous and genes are independent normally distributed variates, it is possible only if range of function f and interval $(f(b_{0,1}), +\infty)$ has no intersections (it could be verified if inverse assumption was made), so $f(b_{0,1})$ is a global maxima of function f.
Thus, we obtain that either $\mathrm{E}f(B_1)=f(B_0)$ and solution is found or $\mathrm{E}f(B_1) > f(B_0)$. This deduction can be made for any step of algorithm, so following assertion holds:
If solution is not found on kth step of evolutionary programming algorithm, then

$$\mathrm{E}f(B_{k+1}) > f(B_k) \tag{38}$$

7. Concluding remarks

We consider simple genetic algorithm and some of variants. Also we have collected several important results which provide valuable insight into the intrinsic principles of genetic algorithms. Finally we consider real-valued optimization problem and some evolutionary method to solve it. Several remarks were made about convergence one of them. But mainly

we consider genetic algorithms in itself. The future of this method, however, is in union with neural networks and fuzzy systems. Below, we mention some perspective approaches:

1. Fuzzy genetic programming. Genetic programming is concerned with the automatic generation of computer programs. Fuzzy genetic programming combines a simple genetic algorithm that on a context-free language with a context-free fuzzy rule language.
2. Genetic fuzzy systems. As mentioned in introduction of this chapter these systems use evolutionary methods for rule base generation and tuning.
3. Genetic fuzzy neural networks. Genetic fuzzy neural networks are the result of adding genetic or evolutionary learning capabilities to systems integrating fuzzy and neural concepts. The usual approach of most genetic fuzzy neural networks is that of adding evolutionary learning capabilities to a fuzzy neural network.
4. Genetic fuzzy clustering algorithm. Genetic algorithms can be used in fuzzy clustering. Most widely used method is to optimize parameters of so-called C-mean FCM-type algorithms, that can improve it performance. Another approach is based on directly solving the fuzzy clustering problem without interaction with any FCM-type algorithm.

8. References

Bagley, J. D. (1967). The Behavior of Adaptive Systems Which Employ Genetic and Correlative Algorithms. PhD thesis, University of Michigan, Ann Arbor.

Cordon, O; F. Gomide, F.; Herrera, F.; Hoffmann, F. & Magdalena L. (2004). Ten years of genetic fuzzy systems: current framework and new trends, *Fuzzy sets and systems*, No. 141, pp. 5-31, ISSN 0165-0114.

Fogel, D. B. (1992). *Evolving Artificial Intelligence.* PhD thesis, University of California, San Diego.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, ISBN 0201157675, Reading, MA.

Herrera, F.; Lozano, M. & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis, *Artificial Intelligence Review*, Vol. 12, pp. 265–319. ISSN 0269-2821.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems,* MIT Press, ISBN 0-262-58111-6, Cambridge, MA.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd extended ed., ISBN 3540606769, Springer, Heidelberg.

Schwefel, H.-P. (1995). Evolution and Optimum Seeking. Sixth-Generation Computer Technology, Series. John Wiley & Sons, New York.

Sharapov, R. R. & Lapshin, A.V. (2006). Convergence of Genetic Algorithms, *Pattern Recognition and Image Analysis,* Vol. 16, No. 3, pp.392-397. ISSN 1054-6618.

22

Genetic Algorithm for Linear Feature Extraction

Alberto J. Pérez-Jiménez & Juan Carlos Pérez-Cortés[1]
Universidad Politécnica de Valencia
Spain

1. Introduction

Feature extraction is a commonly used technique applied before classification when a number of measures, or features, have been taken from a set of objects in a typical statistical pattern recognition task. The goal is to define a mapping from the original representation space into a new space where the classes are more easily separable. This will reduce the classifier complexity, increasing in most cases classifier accuracy. Feature extraction methods can be divided into linear and non-linear, depending on the nature of the mapping function (Lerner et al., 1998). They can also be classified as supervised or unsupervised, depending on whether the class information is taken into account or not. Feature extraction can also be used for exploratory data analysis, where the aim is not to improve classification accuracy, but to visualise high dimensional data by mapping it into the plane or the 3-dimensional space.

The best known linear methods are Principal Component Analysis, or PCA (unsupervised) (Fukunaga, 1990), Linear Discriminant Analysis or LDA (supervised) (Fukunaga, 1990; Aladjem, 1991; Siedlecki et al., 1988), and Independent Component Analysis or ICA (unsupervised) (Cardoso, 1993). Schematically, PCA preserves as much variance of the data as possible, LDA attempts to group patterns of the same class, while separating them from the other classes, and ICA obtains a new set of features by extracting the less correlated (in a broad sense) directions in the data set. On the other hand, well-known non-linear methods are: Sammon's Mapping (unsupervised) (Sammon, 1969; Siedlecki et al. 1988), non-linear discriminant analysis or NDA (supervised) (Mao & Jain, 1995), Kohonen's self-organising map (unsupervised) (Kohonen, 1990) and evolutionary extraction (supervised) (Liu & Motoda, 1998). Sammon's mapping tries to keep the distances among the observations using hill-climbing or neural network methods (Mao & Jain, 1995; Sammon, 1969), NDA obtains new features from the coefficients of the second hidden layers of a multi-layer perceptron (MLP) (Mao & Jain, 1995) and Kohonen Maps project data in an attempt to preserve the topology. Finally, evolutionary extraction uses a genetic algorithm to find combinations of original features in order to improve classifier accuracy. These new features are obtained by multiplying, dividing, adding or subtracting the original features.

In the linear methods, the mapping function is known and simple; therefore, the task is reduced to finding the coefficients of the linear transformation by maximising or minimising

[1] Work partially supported by the Spanish "Ministerio de Educación y Ciencia" under grant DPI2006-15542-C04-02.

a criterion. If a proper criterion function is selected, many standard linear algebra methods can be applied. However, in many cases a linear mapping may not be powerful enough to obtain good results, making it necessary to consider non-linear mappings.
Non-linear mappings present different functional forms and this often makes their application more problem-dependent. Furthermore, since closed-form optimisation methods for many non-linear functions are not known or, are in general less stable and powerful than their linear counterparts when they do exist, non-parametric estimation techniques such as neural networks or iterative optimisation procedures such as hill-climbing or genetic algorithms are commonly used.
In this paper, a new linear supervised feature extraction method referred to as GLP (genetic linear projections) is proposed. The goal of this method is to find the coefficients of a set of linear projections by maximising a certain criterion function. The success confidence rate in the new feature space, a criterion that is directly related to the estimated accuracy of a Nearest Neighbour classifier, is proposed as the function to maximise. Because no closed-form solution exists to maximise this criterion, a well-known numerical optimisation method, genetic algorithms (GA) (Holland, 1975; Goldberg, 1989), has been employed.
In Section 2, we describe the GLP algorithm. In Section 3, we present a comparison between a linear method (PCA), a non-linear method (NDA) and the proposed GLP algorithm over several data sets in terms of both feature extraction and data projection purposes. Finally, we present some conclusions and further works in section 4.

2. Genetic Linear Projection (GLP)

2.1 Linear feature extraction

In linear feature extraction, new features are obtained by means of linear projections (LP). A LP is defined as follow

$$LP(x) = c_1x_1 + c_2x_2 + \dots c_dx_d \tag{1}$$

where x is a d-dimensional vector with components x_i and c_i are the projection coefficients representing the projection axis. By representing the coefficients as a vector, $c=\{c_1, c_2, \dots c_d\}^T$, the application of a LP can be redefined as

$$LP(x) = c^Tx \tag{2}$$

Each LP defines a new feature to represent x. To define m new features, we need m LPs that can be arranged as a $m \times d$ matrix (C). By defining the transformation matrix C in this way, a new representation of x, $y=\{y_1, y_2, \dots y_m\}^T$, can be obtained by means of

$$y = Cx \tag{3}$$

Ideally C should be selected in order to minimise the Bayes error (Duda & Hart, 1973) in the new space. Moreover, this expression depends on the *a posteriori* probability of classes, and in general, is not straightforward to obtain. Even when this expression exists, usually no tractable expression for the gradient can be obtained. For this reason, linear feature extraction methods often employ other less suitable, but simpler, class separability measures in order to use closed-form solutions, or they employ gradient-based numerical optimisation methods in order to obtain C.

2.2 Criteria

In this work, we propose to obtain C by optimising a criterion function that is directly related to the Bayes error. The estimated error rate, $\hat{E}$, of a k-Nearest Neighbour classifier (k-NN) can be a good option. Under certain convergence conditions, the error rate of a k-NN classifier offers an optimistic, but very close estimation of the Bayes error (Devijver & Kittler, 1982). The $\hat{E}$ can be easily calculated by error count over a test set by the expression,

$$\hat{E} = \frac{e}{n} \tag{4}$$

where n is the size of the test set, and e is the number of observations that are not correctly classified by the k-NN classifier. The estimated success rate of a classifier, $\hat{A}$, is directly related to $\hat{E}$ and can be calculated as $\hat{A}=1-\hat{E}$.
Another interesting criterion can be defined using the conditional probability of an observation x belonging to a class w_i, $P(w_i \mid x)$. Most statistic classifiers can provide an estimation of this value that can be used as a confidence measure for the classified observations. In a k-NN classifier, a maximum likelihood estimation of $P(w_i \mid x)$, $\hat{P}(w_i \mid x)$, can be obtained as

$$\hat{P}(w_i \mid x) = \frac{k_i}{k} \tag{5}$$

where k is the number of neighbours employed by the k-NN classifier, and k_i is the number of neighbours of class w_i. We formulate the *estimated success confidence rate*, $\hat{C}_a$, of a classifier as

$$\hat{C}_a = \frac{1}{n}\sum_{x \in X} \delta(w_{\theta'_x}, w_{\theta_x})\hat{P}(w_{\theta_x}, x) \tag{6}$$

where n is the number of observations of a sample X, w_{θ_x} is the real class of $x \in X$, $w_{\theta'_x}$ is the class assigned to x by the employed classifier, and $\delta(w_i, w_j)$ is defined as

$$\delta(w_i, w_j) = \begin{cases} 1 & if \quad w_i = w_j \\ 0 & if \quad w_i \neq w_j \end{cases} \tag{7}$$

In the case the value $\hat{P}(w_i \mid x)$ is always *1*, the definition of $\hat{C}_a$ equals the value of $\hat{A}$. The criterion can be seen as a confidence measure of the estimated success rate of a classifier.
When projecting data, the use of $\hat{C}_a$ as the optimisation criterion has advantages with respect to $\hat{A}$ (or $\hat{E}$). Two projections with the same $\hat{A}$ value can have different values of $\hat{C}_a$. In this situation, the k-NN classifier implemented in the feature space with a better $\hat{C}_a$ value is expected to show more confidence in its decisions. For this reason, we propose the success confidence rate, $\hat{C}_a$, as the criterion to estimate the linear transformation matrix C.

2.3 Genetic optimisation

Since no closed-form method is known to optimise the proposed criterion, and since there is no tractable expression for its gradient, random numerical optimisation methods must be used.
The number of parameters to be estimated by the optimisation method is $m \times d$, with m being the number of LPs or new features to obtain, and with d being the dimensionality of the original data. If we want to project high-dimensional data, the number of parameters to estimate will be large. For this reason, we propose a GA as an appropriate paradigm to carry out the optimisation.
GAs have proven to be specially useful in large search spaces (Goldberg, 1989). We have used a GA with the following properties:

- An individual is composed of m chromosomes representing the m LPs to search. Each chromosome contains d genes, and each gene contains a binary string of b bits that encodes a coefficient of the LP in fixed point format.
- The fitness function is defined as the computed success confidence rate, $\hat{C}_a$, of a k-NN classifier trained with the projected data obtained from the LPs coded in the individual.
- The genetic selection scheme uses a rank-based strategy (Mitchell, 1996). In this strategy, the probability of being selected is computed from the rank position of the individuals. In our case, this method gave a faster convergence than a fitness-proportionate method.
- The following settings are used for the rest of the parameters: crossover probability is *0.6*, mutation probability is *0.001*, population size is *100* and the maximum number of generations is *300*.

Finally, since estimating the success confidence rate of a k-NN classifier is a time-consuming task, a fast neighbour search by means of kd-trees (Friedman et al., 1977) was implemented to reduce the computational cost. Additionally, a micro-grain parallel GA (Shyh-Chang et al., 1994) was implemented, allowing the use of several computers to compute individual fitness functions, obtaining a linear speedup.
We refer to the described method as Genetic Linear Projections (GLP).

3. Comparative study

3.1 Methodology

In this section, the GLP method is compared with the well-known PCA (linear, unsupervised), and the NDA by means of neural networks (non-linear, supervised). The comparison addresses both, feature extraction and data projection (mapping) applications. The three methods are applied to sixteen data sets in order to obtain different numbers of new features (see Table 1). Since the results obtained by NDA and GLP are not deterministic, for this methods five runs are performed with each parameter combination.
PCA obtains an eigenvector matrix (Φ) and an eigenvalue diagonal matrix (Δ) from the covariance matrix of the original data by means of a closed-form method. The columns of Φ correspond to orthonormal linear projections (eigenvectors) in the directions of maximal scatter. The values in the diagonal of Δ (eigenvalues) allow us to sort these directions

depending on the scatter. To reduce a *d*-dimensional original space to an *m*-dimensional space, with $m<d$, we only have to keep the *m* eigenvectors with the largest eigenvalues.
The NDA method is based on training a two-hidden layer neural network. This is accomplished using the backpropagation algorithm with momentum, obtaining the new features from the response of the units of the second hidden layer. The number of units of the second hidden layer must be selected to equal the number of desired new features.
In order to detect possible overfitting problems with the three methods, each data set is split into a training set (70%) and a test set (30%). The methods are applied to the training sets, testing the performance of the obtained projections in the test sets. In order to estimate the success confidence rate, $\hat{C}_a$, a leaving-one-out procedure is employed in the training set for small data sets (less than *5000* patterns). A hold-out procedure is used with bigger data sets.
In the case of feature extraction, the performance of the methods is compared in terms of the success rate improvement obtained, as well as in terms of the reduction obtained in the number of features. Because the estimate of the success rate is obtained by error count (Duda & Hart, 1973), the 95% confidence intervals are provided to correctly compare the results.
For data projection purposes, the performance of these methods is first compared by means of visual judgement over the 2-dimensional projections obtained from the data sets, and then by means of the success rate of a *k*-NN classifier computed for each data set in the original and projected spaces. This quantitative criterion gives us an idea of how well the class structure is preserved by the projections (Mao & Jain, 1995).

3.2 Corpora

The corpora are selected from well-known data sets from the UCI repository (Blake & Merz, 1998). A self-designed synthetic data set, *cookies*, is also used. This data set has been created to represent a well-known case in which PCA does not work well because the maximal scatter axes are not the most significant. This corpus consists of two *10*-dimensional normal distributions with covariance matrices

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.0001 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

and means $\mu_1=(+0.1,\ 0,\ 0,\ \ldots)$, $\mu_1=(-0.1,\ 0,\ 0,\ \ldots)$. Each class has *1000* patterns. These distributions represent two hyperspheres that are flattened (like cookies) in the dimension that separates them.

Table 1 summarises the features (size, dimensionality, number classes, ...) of every data set used.

Corpora	Size	Dim.	Classes	*k*	New features
german	1000	24	2	23	[1,2 - 20]
glass	214	9	6	3	[1,2 - 8]
cookies	2000	10	2	21	[1,2 - 10]
ionosphere	351	34	2	1	[1,2 - 30]
iris	151	4	3	15	[1,2,4]
digits	3000	196	10	3	[1,2 - 100]
bupa	345	6	2	23	[1,2 - 6]
pima	768	8	2	19	[1,2 - 8]
segment	2310	19	7	1	[1,2 - 15]
sonar	208	60	2	1	[1,2 - 60]
vehicle	846	18	4	3	[1,2 - 15]
wine	178	13	3	15	[1,2 - 10]
waveform	5000	21	3	27	[1,2 - 20]
page blocks	5473	10	5	3	[1,2 - 10]
sat	6435	36	6	5	[1,2 - 35]
musk	6598	166	2	3	[1,2 - 100]

Table 1. Data set features: size, dimensionality, number of classes, optimum *k* value for the *k*-NN classifier and a list of the new number of features searched for the different methods (values with dashes mean that several numbers in the interval have been searched for).

Corpora	Original		PCA		NDA		GLP	
german	0.7278 0.7285	(24)	0.7307 0.7086	(15)	***0.9882*** ***0.7183***	***(6)***	***0.8198*** ***0.7461***	***(6)***
glass	0.7308 0.5690	(9)	***0.7244*** ***0.6897***	***(4)***	0.7485 0.7255	(8)	0.8160 0.7843	(8)
cookies	0.4993 0.4728	(10)	0.4993 0.4729	(10)	***1.0000*** ***1.0000***	***(1)***	1.0000 1.0000	(2)
ionosphere	0.8833 0.8198	(34)	0.9083 0.8378	(8)	***1.0000*** ***0.9174***	***(1)***	0.9876 0.8899	(2)
iris	0.9725 0.9268	(4)	0.9725 0.9268	(4)	0.9818 0.9750	(4)	0.9909 0.9500	(4)
digits	0.9522 0.9504	(196)	***0.9603*** ***0.9559***	***(40)***	0.9948 0.9122	(15)	0.9581 0.9578	(100)
bupa	0.6901 0.5922	(6)	0.6901 0.5922	(6)	0.8092 0.7229	(6)	0.7557 0.6627	(6)
pima	0.7623 0.7014	(8)	0.7605 0.7330	(6)	0.8680 0.7352	(8)	***0.8272*** ***0.7589***	***(4)***
segment	0.9599 0.9607	(19)	0.9599 0.9607	(15)	***0.9876*** ***0.9757***	***(6)***	0.9826 0.9729	(10)
sonar	0.8392 0.8462	(60)	0.8531 0.8462	(10)	***1.0000*** ***0.7458***	***(1)***	1.0000 0.7966	(10)
vehicle	0.7141 0.7179	(18)	0.7059 0.7265	(15)	***0.9397*** ***0.8308***	***(6)***	0.8362 0.7594	(10)
wine	0.9590 0.9464	(13)	0.9836 0.9464	(6)	***1.0000*** ***0.9811***	***(1)***	1.0000 1.0000	(2)
waveform	0.8396 0.8531	(21)	***0.8541*** ***0.8720***	***(2)***	0.9429 0.8375	(10)	0.8649 0.8578	(15)
page blocks	0.9611 0.9685	(10)	0.9616 0.9685	(8)	***0.9746*** ***0.9685***	***(4)***	0.9650 0.9710	(8)
sat	0.9092 0.8965	(36)	0.9116 0.8944	(15)	***0.9555*** ***0.8944***	***(10)***	0.9107 0.8991	(25)
musk	0.9676 0.9641	(166)	0.9673 0.9656	(90)	***1.0000*** ***0.9946***	***(4)***	0.9783 0.9759	(15)

Table 2. The best success rates obtained by PCA, GLP and NDA on the training set (top) and the test set (bottom). The results on the original feature space are also shown. The values in brackets represent the number of features. Values in boldface represent the methods that obtain the highest reduction, maintaining or improving the original correct classification rate.

3.3 Results

Tables 2 and 3 present the best runs obtained for the three methods. The success rate is presented instead of the success confidence rate (the criterion used to optimise) because we are interested in the final classifier performance. Analysing the results and considering their 95% confidence intervals (see Table 3) it can be observe that a significant classifier improvement was only obtained by NDA in three data sets (*cookies*, *vehicle* and *musk*), and by GLP in one data set (*cookies*). In all the other cases, only feature reduction was achieved, i.e., the classifier obtained similar results to the original space but with fewer features.

By analysing all the runs (not only the best ones), and taking into account the confidence intervals, it can be observed that the three methods obtained a similar reduction with the exception of PCA, which obtained a worse reduction in five data sets (*cookies*, *ionosphere*, *segment*, *sonar* and *vehicle*). For instance, Figure 1 shows the results for the *vehicle*, *cookies* and *segment* data sets, for different numbers of features. It also shows that a similar reduction was obtained by NDA and GLP in these cases, while PCA yielded significantly worse results.

It is interesting to note that, although only one linear projection was enough to separate the classes of *cookies* data set, PCA and GLP had problems. PCA was not able to do it because the maximal scatter direction was not the optimal in this case. GLP failed because random optimisation methods have problems finding very isolated solutions. Nevertheless GLP was able to find a good solution with two or more linear projections while PCA continued to fail.

In the data projection context, looking at the success rate obtained by the classifiers when projecting data sets into a 2-dimensional space (see Table 4), it can be observed that NDA and GLP outperformed PCA in most of the data sets. NDA obtained better results than GLP for high dimensional data sets (i.e. the *digits* data set, see Figure 2). Visual analysis of obtained projections confirmed these results showing that GLP and NDA produced less overlapping views than PCA (see Figures 2, 3 and 4).

Finally, with respect to the time complexity of methods, although the estimation of the success confidence rate, $\hat{C}_a$, was optimised by using *kd-trees*, and although the GA was parallelised to speed up the algorithm, the *off line* cost of GLP was higher than the cost for the other two methods. The method with the lowest cost is LDA because the transformations are obtained by means of a closed-form method and there is no need for several runs as in NDA or GLP.

Regarding the *on line* costs, GLP and LDA generate linear transformations and have an application cost that is lower than the application of the non-linear transformations generated by the neural network on NDA.

4. Conclusions

From the results obtained, we can conclude that although NDA obtains good results with non-linear projections in all data sets, similar results can be obtained using GLP in most of them. This indicates to us that, in practice, linear projections can obtain results just as good as non-linear projections in most cases. Even though PCA employs linear projections as well, it performs worse in some data sets probably because it is an unsupervised method. Classical linear, supervised feature extraction methods like LDA have important limitations: first, the number of new features is limited by the number of classes; and second, numerical problems arise when working with high dimensional or small data sets, restricting its use.

The proposed GLP method does not have these limitations. The main drawback of the GLP method is it computational cost; however, this is an *off line* process. Once the

Corpora	Original		PCA		NDA		GLP	
german	0.7278	0.6925 0.7598	0.7307	0.6955 0.7626	***0.9882***	***0.9776*** ***0.9951***	***0.8198***	***0.7895*** ***0.8478***
	0.7285	0.6760 0.7794	0.7086	0.6551 0.7607	0.7183	0.6620 0.7670	0.7461	0.6935 0.7949
glass	0.7308	0.6529 0.8008	0.7244	0.6529 0.8008	0.7485	0.6743 0.8187	0.8160	0.7474 0.8771
	0.5690	0.4482 0.7006	0.6897	0.5594 0.7976	0.7255	0.6091 0.8370	0.7843	0.6603 0.8749
cookies	0.4993	0.4728 0.5258	0.4993	0.4728 0.5258	***1.0000***	***0.9974*** ***1.0000***	***1.0000***	***0.9974*** ***1.0000***
	0.4728	0.4328 0.5142	0.4729	0.4328 0.5142	***1.0000***	***0.9939*** ***1.0000***	***1.0000***	***0.9939*** ***1.0000***
ionosphere	0.8833	0.8391 0.9227	0.9083	0.8672 0.9429	***1.0000***	***0.9851*** ***1.0000***	***0.9876***	***0.9646*** ***0.9975***
	0.8198	0.7319 0.8874	0.8378	0.7535 0.9028	0.9174	0.8435 0.9601	0.8899	0.8089 0.9395
iris	0.9725	0.9329 0.9977	0.9725	0.9329 0.9977	0.9818	0.9481 0.9998	0.9909	0.9655 1.0000
	0.9268	0.8173 0.9860	0.9268	0.8173 0.9860	0.9750	0.8823 0.9994	0.9500	0.8485 0.9946
digits	0.9522	0.9424 0.9611	0.9603	0.9512 0.9684	***0.9948***	***0.9906*** ***0.9974***	0.9581	0.9486 0.9663
	0.9504	0.9337 0.9633	0.9559	0.9400 0.9681	***0.9122***	***0.8918*** ***0.9299***	0.9578	0.9425 0.9700
bupa	0.6901	0.6305 0.7506	0.6901	0.6305 0.7506	***0.8092***	***0.7537*** ***0.8567***	0.7557	0.7003 0.8119
	0.5922	0.4910 0.6880	0.5922	0.4910 0.6880	0.7229	0.6316 0.8112	0.6627	0.5703 0.7594
pima	0.7623	0.7252 0.7988	0.7605	0.7233 0.7971	***0.8680***	***0.8382*** ***0.8970***	0.8272	0.7941 0.8596
	0.7014	0.6408 0.7625	0.7330	0.6728 0.7906	0.7352	0.6728 0.7906	0.7589	0.7004 0.8145
segment	0.9599	0.9490 0.9688	0.9599	0.9490 0.9688	***0.9876***	***0.9810*** ***0.9924***	***0.9826***	***0.9751*** ***0.9885***
	0.9607	0.9438 0.9742	0.9607	0.9438 0.9742	0.9757	0.9610 0.9856	0.9729	0.9575 0.9834
sonar	0.8392	0.7716 0.8967	0.8531	0.7872 0.9081	***1.0000***	***0.9749*** ***1.0000***	***1.0000***	***0.9749*** ***1.0000***
	0.8462	0.7233 0.9198	0.8462	0.7233 0.9198	0.7458	0.6150 0.8447	0.7966	0.6682 0.8834
vehicle	0.7141	0.6763 0.7506	0.7059	0.6676 0.7425	***0.9397***	***0.9168*** ***0.9570***	***0.8362***	***0.8038*** ***0.8651***
	0.7179	0.6597 0.7738	0.7265	0.6680 0.7812	***0.8308***	***0.7823*** ***0.8777***	0.7594	0.7055 0.8139
wine	0.9590	0.9084 0.9868	0.9836	0.9430 0.9980	1.0000	0.9707 1.0000	1.0000	0.9707 1.0000
	0.9464	0.8434 0.9882	0.9464	0.8434 0.9882	0.9811	0.8993 0.9995	1.0000	0.9328 1.0000
waveform	0.8396	0.8271 0.8517	0.8541	0.8419 0.8655	***0.9429***	***0.9346*** ***0.9503***	***0.8649***	***0.8531*** ***0.8760***
	0.8531	0.8344 0.8709	0.8720	0.8540 0.8885	0.8375	0.8177 0.8557	0.8578	0.8393 0.8753
page blocks	0.9611	0.9545 0.9670	0.9616	0.9551 0.9675	***0.9746***	***0.9692*** ***0.9794***	0.9650	0.9587 0.9706
	0.9685	0.9593 0.9768	0.9685	0.9593 0.9768	0.9685	0.9593 0.9768	0.9710	0.9621 0.9789
sat	0.9092	0.9004 0.9174	0.9116	0.9030 0.9198	***0.9555***	***0.9492*** ***0.9614***	0.9107	0.9020 0.9189
	0.8965	0.8825 0.9101	0.8944	0.8803 0.9082	0.8944	0.8803 0.9082	0.8991	0.8852 0.9125
musk	0.9676	0.9620 0.9724	0.9673	0.9618 0.9722	***1.0000***	***0.9992*** ***1.0000***	***0.9783***	***0.9737*** ***0.9823***
	0.9641	0.9550 0.9719	0.9656	0.9566 0.9732	***0.9946***	***0.9901*** ***0.9972***	0.9759	0.9680 0.9821

Table 3. The bests success rates obtained by PCA, GLP and NDA on the training set (top) and the test set (bottom). The results on the original feature space are also shown. Small values represent the 95% confidence intervals for the correct classification rate. Values in boldface represent values that are significantly different from the original.

transformations are computed, the cost of applying them to new data is lower than applying the neural network trained by the NDA method. Moreover, the process of training an NDA neural network is not straightforward in many cases, having convergence problems.
From the point of view of data projection, it can be concluded that NDA projections outperform our GLP method when the intrinsic dimensionality is high. In these cases, the NDA projection is able to obtain a good view of the class structure even in a 2-dimensional projection. Nevertheless, we consider that NDA has one important drawback. Because non-linear transformations are used, an important distortion of the original space occurs, especially when projecting into a 2-dimensional space in an attempt to preserve the class structure (see Figure 3). In this situation, a synthetic view of the configuration of real clusters is obtained. The GLP method uses linear transformations, thereby producing less distorted and more meaningful views of the original space (distortion can appear because the new axes are not necessarily orthogonal). The PCA method is linear and unsupervised; therefore, the projections computed do not always show a good view of the class structure if the discriminant axes are not the ones with the highest variance.

Corpora	Original	PCA	NDA	GLP
german	0.7278	0.7178	***0.9075***	0.7885
glass	0.7308	0.6731	0.6454	***0.6626***
cookies	0.4993	0.3986	***1.0000***	***0.9959***
ionosphere	0.8833	0.7125	***0.9901***	0.9769
iris	0.9725	0.9266	0.9709	***0.9745***
digits	0.9522	0.4364	***0.8508***	0.6336
bupa	0.6901	0.5331	***0.7756***	0.7206
pima	0.7623	0.7130	***0.8586***	0.7825
segment	0.9599	0.6402	***0.9412***	0.9135
sonar	0.8392	0.5664	***0.9879***	0.9289
vehicle	0.7141	0.4935	***0.7921***	0.7438
wine	0.9590	0.9508	***0.9968***	***0.9936***
waveform	0.8396	0.8541	***0.8909***	0.8514
page blocks	0.9611	0.9369	***0.9641***	0.9530
sat	0.9092	0.8322	***0.8756***	0.8380
musk	0.9676	0.8913	***0.9993***	0.9296

Table 4. Mean values for the correct classification rate obtained over the training sets when looking for two new features (exploratory analysis). The bests results for each data set are in boldface.

5. References

Aladjem, M. (1991). Parametric and nonparametric linear mappings of multidimensional data. *Pattern Recognition*, 24, 6, pp 534–551.

Blake, C. & Merz, C. (1998). UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. University of California, Irvine.

Cardoso, J. and Comon, P. (1996). Independent component analysis, a survey of algebraic methods. *Proceedings of ISCAS-96*, Vol. 2, pp 93–96.

Devijver, P. and Kittler, J. (1982), *Pattern Recognition: A Statistical Aproach*. Prentice Hall, London.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York.

Friedman, J.; Bentley, J. & Finkel, R. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathemathical Software*, 3, pp 209–226.

Fukunaga, K. (1990). *Statistical Pattern Recognition*. Academic Press, second edition edition.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Tabor. The University of Michigan Press.

Kohonen, T. (1990). The self-organizing map. *Proceedings IEEE*, 78, 9, pp 1464–1480.

Lerner, B.; Guterman, H.; Aladjem, M.; Dinstein, I. & Romen, Y. (1998). On pattern classification with sammon's nonlinear mapping (an experimental study). *Pattern Recognition*, 31, 4, pp 371 – 381.

Liu, H. & Motoda, H. (eds.). (1998). *Feature Extraction, Construction and Selection. A Data Mining Perspective*. Kluwer Academic Publishers.

Mao, J. & Jain, A. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6, 2.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.

Sammon, J. (1969). A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18, 5.

Shyh-Chang, L.; Punch, W. & Goodman, E. (1994). Coarse-grain parallel genetic algorithm: categorization and a new approach. *6th IEEE Symposium on Parallel and Distributed Processing*, pp 28 – 37.

Siedlecki, W.; Siedlecka, K. & Sklansky, J. (1988). An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21, 5, pp 411 – 429.

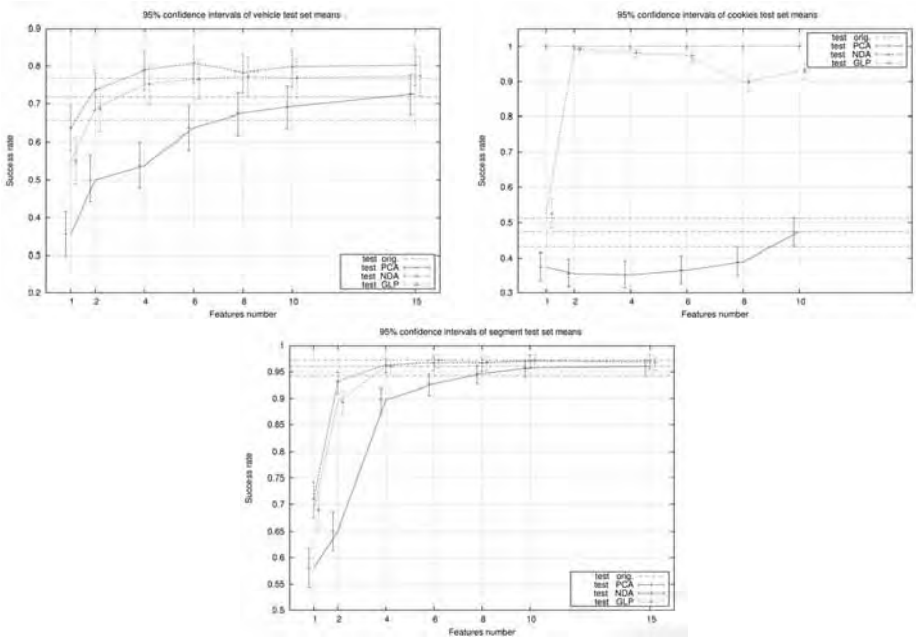


Fig. 1. Correct classification rate results for *vehicle* (top), *cookies* (middle) and *segment* (bottom) data set. The 95% confidence intervals are shown.

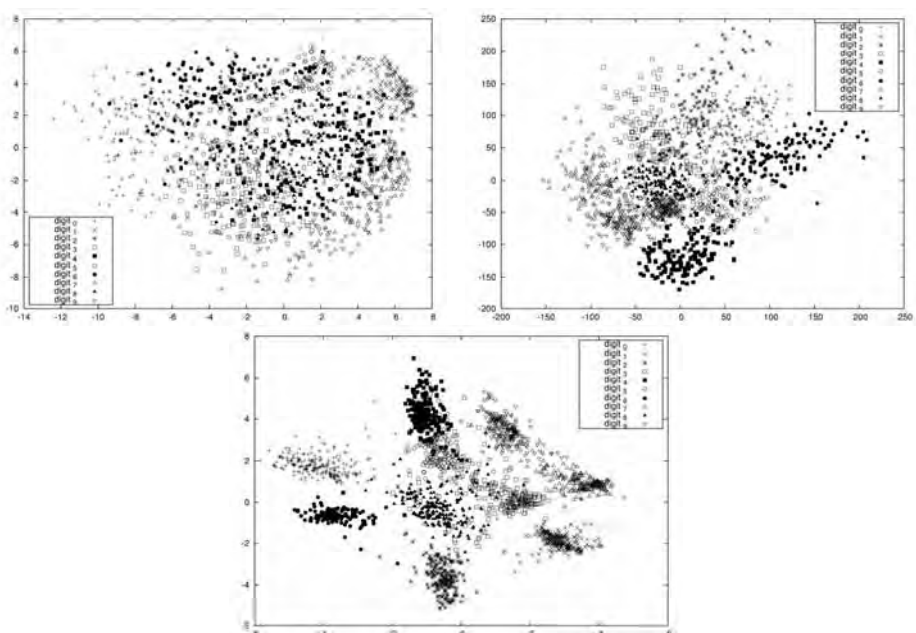


Fig. 2. Projections obtained for the *digits* data set by PCA (top), GLP (middle) and NDA (bottom).

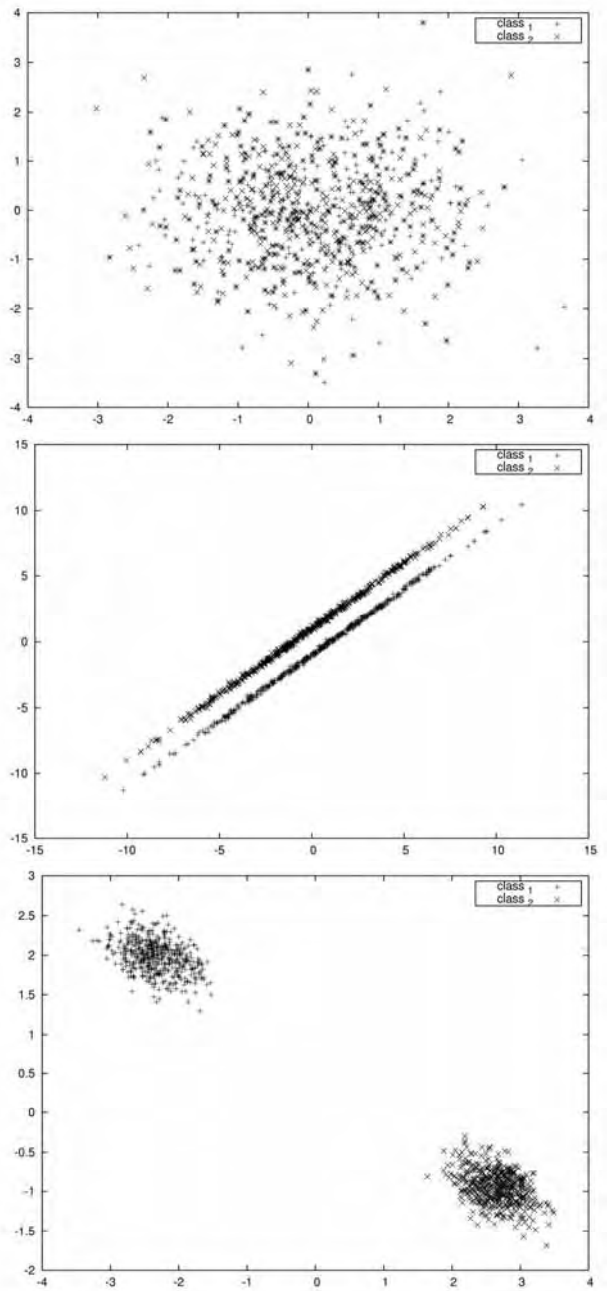


Fig. 3. Projections obtained for the *cookies* data set by PCA (top), GLP (middle) and NDA (bottom).

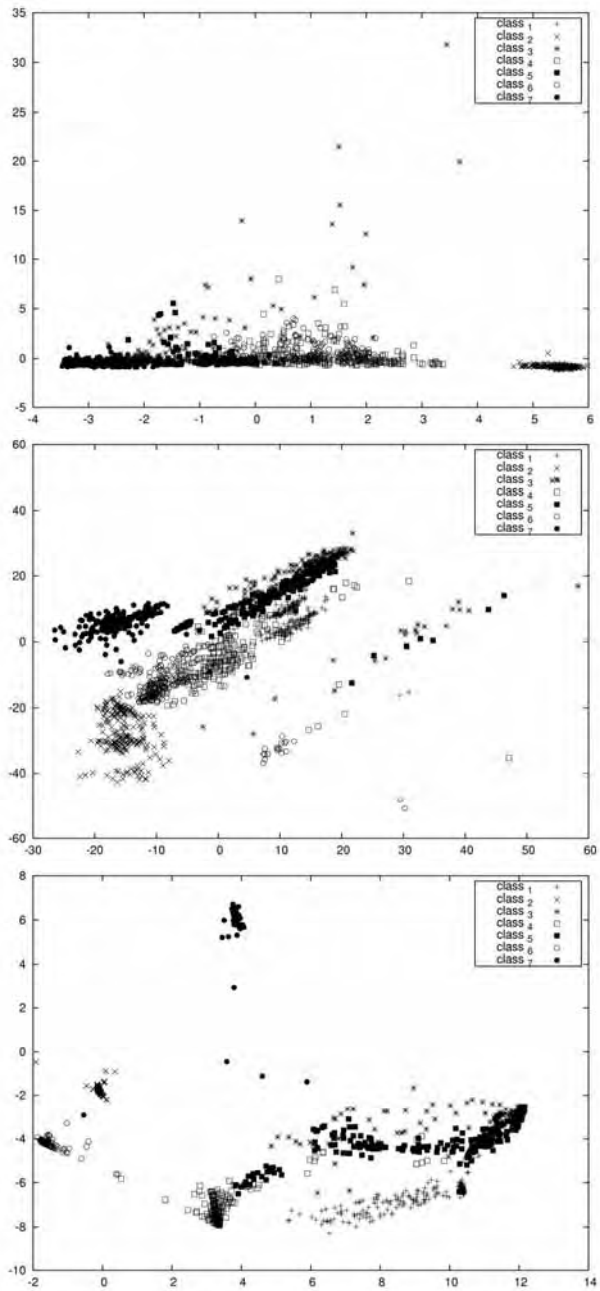


Fig. 4. Projections obtained for the *segment* data set by PCA (top), GLP (middle) and NDA (bottom).

23

Recognition of Partially Occluded Elliptical Objects using Symmetry on Contour

June-Suh Cho[1]
Joonsoo Choi[2]
[1]*Hankuk University of Foreign Studies*
[2]*Kookmin University*
Republic of Korea

1. Introduction

There are many research efforts in object recognition. Most existing methods for object recognition are based on full objects. However, many images contain multiple objects with occluded shapes and regions. Due to the occlusion of objects, image retrieval can provide incomplete, uncertain, and inaccurate results. To resolve this problem, we propose a new method to reconstruct objects using symmetry properties since most objects in a given image database are represented by symmetrical figures.

Even though there have been several efforts in object recognition with occlusion, current methods have been highly sensitive to object pose, rotation, scaling, and visible portion of occluded objects. In addition, many appearance-based and model-based object recognition methods assumed that they have known occluded regions of objects or images through extensive training processes with statistical approach. However, our new approach is not limited to recognizing occluded objects by pose and scale changes, and does not need extensive training processes.

Unlike existing methods, the proposed method finds shapes and regions to reconstruct occluded shapes and regions within objects. We assume that we only consider the elliptical objects in recognition. The proposed approach can handle object rotation and scaling for dealing with occlusion, and does not require extensive training processes. The main advantage of our proposed approach is that it becomes simple to reconstruct objects from occlusions using symmetry. We present a robust method, which is based on the contours of objects, for recognizing partially occluded objects based on symmetry properties. The contour-based approach finds a symmetry axis using the maximum diameter from the occluded object.

In experiments, we demonstrate how a proposed method reconstructs and recognizes occluded shapes and regions using symmetry. Experiments use rotated and scaled objects for dealing with occlusion. We use mirror symmetry to find possible occluded regions in objects. Examples of partially occluded objects are shown in Figure 1.1.

We also evaluate the recognition rate of the reconstructed objects using symmetry and the visible portion of the occluded objects for recognition. The method produces average recognition rates for cups and plates above 88% with 30% occlusion. In this case, part of the

objects needs to be visible for correct recognition of all objects. Specifically, 67% should be visible for the contour-based approach. Experimental results show that the reconstructed objects are properly recognized by our method.

Fig. 1.1 Examples of Outlined Objects including the Occlusion. Objects include cups, bowls, and plates

2. Related Work

Current object recognition methods represent models either as a collection of geometric measurements or as a collection of images of an object. Some researchers proposed learning control strategies and methods of probabilistic models for object recognition based on local appearance.

There have been several research efforts in object recognition. (Krumm, 1997) proposed a new algorithm for detecting objects in images which uses models based on training images of the object, with each model representing one pose. (Williams, 1997) proposed a method for the reconstruction of solid-shape from image contour using the Huffman labeling scheme. Also, (Williams & Hanson, 1996) described a method for visual reconstruction of visible and occluded forward facing surfaces from image contour.

For object recognition, (Rajpal et al., 1999) introduced a method for partial object recognition using neural network based indexing. They used invariants of local contour segmentation for indexing. (Chang & Krumm, 1999) used the color cooccurrence histogram based on pairs of pixels. To recognize objects in images, they abstracted away unimportant details by using subtemplates, normalized correlations, and edge features. They also recognized occluded objects using probability approximation for parameters. (Schiele & Pentland, 1999) proposed a method to perform partial object recognition using statistical methods, which are based on multidimensional receptive field histograms.

A number of more recent works have used edges for object recognition. (Mikolajczyk et al., 2003) generalized Lowe's SIFT descriptors to edge images, where the position and orientation of edges are used to create local shape descriptors that are orientation and scale invariant (Lowe, 1999). (Carmichael & Hebert, 2004) proposed a method to use a cascade of classifiers of increasing aperture size, trained to recognize local edge configurations, to discriminate between object edges and clutter edges; this method, however, is not invariant to changes in image rotation or scale. (David & DeMenthon, 2005) proposed a method to use model and image line features to locate complex objects in high clutter environments.

In appearance-based object recognition, (Edwards & Murase, 1997) addressed the occlusion problem inherent in appearance-based methods using a mask to block out part of the basic eigenimages and the input image. (Leonardis & Bischof, 1996) handled occlusion, scaling, and translation by randomly selecting image points from the scene and their corresponding points in the basis eigenvectors. (Rao, 1997) applied the adaptive learning of eigenspace basis vectors in appearance-based methods. The dynamic appearance-based approach is used to predict spatial and temporal changes in the appearance of a sequence of images. (Ohba & Ikeuchi, 1997) were able to handle translation and occlusion of an object using eigenwindows. The eigenwindows encode information about an object's appearance for only a small section of its view.

In model-based object recognition, (Jones & Bhanu, 1999) described a model-based object recognition method using the combination of a SAR approach, model for azimuthal variance, articulation invariants, and the resolution of the sensor data. (Boshra & Bhanu, 2000) also described a model-based object recognition method using the probability of correct recognition.

Current methods for dealing with occlusion have been based on template matching, statistical approaches using localized invariants, and recognition of occluded regions based on local features. In addition, there are many efforts in ellipse construction and detection (Ho & Chan, 1995; Wu & Wang, 1993). In this paper, we propose unique methodologies in

object recognition for dealing with occlusion based on symmetry properties through the ellipse reconstruction.
Even though there have been several efforts in object recognition with occlusion, current methods have been highly sensitive to object pose and scaling. In addition, many appearance-based and model-based object recognition methods assumed that they have known occluded regions of objects or images through extensive training processes. However, our proposed method is not limited to recognizing occluded objects by pose and scale changes, and do not require extensive training processes.

3. The Proposed Method

We discuss the object reconstruction and the parameter estimation method to find the best matching class of input objects using the classification method. (Cho & Choi, 2004) extracted shape parameters from reconstructed objects using RLC lines, such as roundness, aspect ratio, form factor, surface regularity (Adam et al., 2000).
In the following section, we discuss an approach for partial object recognition, which is focused on the contour of objects. This approach tries to find occluded shapes within partially occluded objects. The basic assumption is that most objects are represented by symmetrical figures. When a symmetric object is partially occluded, the symmetry measure to evaluate the symmetric shape is used. After estimating the most similar parameters of occluded shape and region of objects, objects that have the estimated parameters of occluded objects are retrieved.
A basic idea of reconstruction and estimation of occluded objects is to use symmetry properties within objects and to use the contour of objects. Fortunately, most products in electronic catalogs have symmetry in their shapes and they are represented by symmetrical figures. Symmetrical descriptions of shape or detection of symmetrical features of objects can be useful for shape matching, model-based object matching, and object recognition (Bischof & Leondardis, 1998; Blum & Nagel, 1978).
In the given database, we have elliptical and roughly-rounded objects such as plates, cups, pans, and pots, depending on their poses and shapes. First, we consider elliptical objects in which the occlusion changes values of measurements and parameters related to diameters. We assume that we can get diameters from elliptical objects, which are partially occluded.

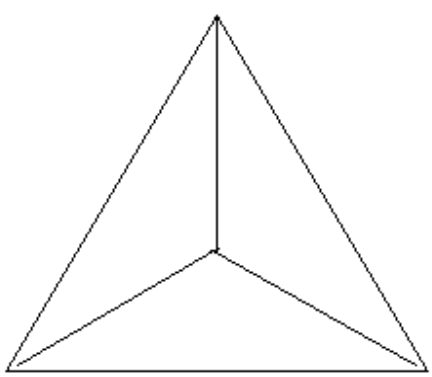


Fig. 3.1 Three-Spoke from the Triangle.

However, the elliptical objects are limited to the shape of objects. Therefore, it may not be applied to other types of shape such as irregular shapes. In this case, since we cannot easily detect the symmetry axes, we introduce the three-spoke type symmetry method as shown in Figure 3.1. We apply this approach to roughly-rounded objects such as cups.

For roughly-rounded objects, we use the three-spoke type method, which is derived from the triangle. The triangle is a basic model to represent figures such as circle, rectangle, and polygon. We use extended lines of the triangle to make axes as shown in Figure 3.1. The three-spoke type symmetry axes, which are equally assigned by 120 degrees, provide the possibility to detect proper symmetry axes on roughly-rounded objects. Therefore, this method can detect symmetry axes in roughly-rounded objects.

In order to perform the following procedures, we assume that objects are represented by symmetrical figures.

- We have an occluded elliptical object in Figure 3.2 and roughly-rounded object in Figure 3.6, we can get cutting points of the occlusion $(x, y)'$ and $(x, y)''$, that are given by overlapping or cutting.

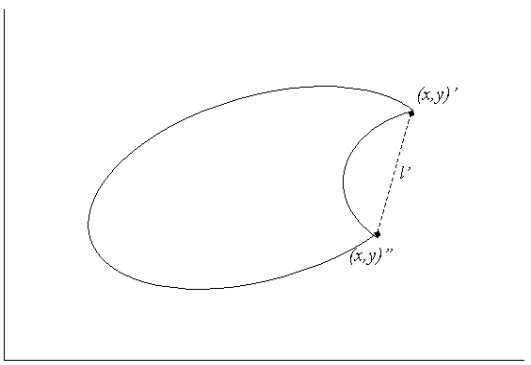


Fig. 3.2 The Occlusion Area Estimation using Symmetry: Get cutting points *(x,y)'* and *(x,y)''* and get a distance *l'*.

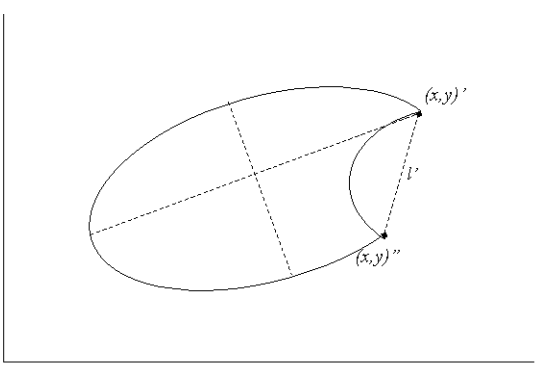


Fig. 3.3 The Occlusion Area Estimation using Symmetry: Get the maximum diameter and the symmetry axis.

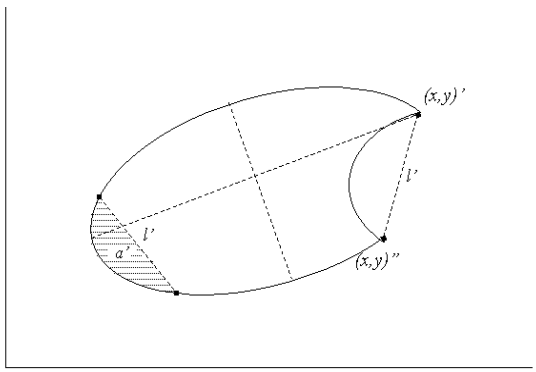


Fig. 3.4 The Occlusion Area Estimation using Symmetry: Get the estimated region a' using a line l' and the symmetry axis.

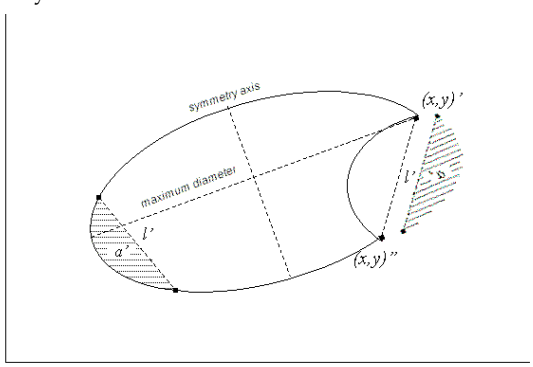


Fig. 3.5 The Occlusion Area Estimation using Symmetry: Add region a' to occluded shape and region and re-captured the estimated shape of an object.

- Compute a distance between two cutting points from $(x, y)'$ and $(x, y)''$, which is called a line l' as in Figure 3.2 and 3.6.
- Based on a line l', make a connection between two points, fill the concave region and re-captured the shape. It is important to compute a centroid in an object.
- Get the maximum diameter from re-captured shape using extremal points as shown in Figure 3.4 and 3.7. Two extremal points *(r, l)* and *(r, l)'* derives from re-captured shape as in Figure 3.7. The distance between two extreme boundary points is represented by the maximum diameter.
- In elliptical objects, one of the maximum and minimum diameters can be a symmetry axis. In roughly-rounded objects, we use the three-spoke type symmetry, one spoke can be a symmetry axis to find occluded region within an object.
- Centroid Detection: In case of elliptical objects, we find a centroid based on the maximum diameter and a line perpendicular to the maximum diameter, which is located in the center of the length of the maximum diameter. We select symmetry axes based on one of these lines as in Figure 3.3. In roughly-rounded objects, we get a centroid, based on whole region of an object. Equation 2 is adapted from (Russ, 1998). If

the centroid is calculated by equation 1 using the boundary pixels only, the results may not be correct. The calculated points will be biased toward whichever part of the boundary is most complex and contains the most pixels. The correct centroid location uses the pairs of coordinates x_i, y_i for each point in the shape boundary. The centroid of an irregular shape is calculated correctly using all of the pixels in an object.

$$C_x = \frac{\sum_{i=0}^{k} x_i}{Area}, C_y = \frac{\sum_{i=0}^{k} y_i}{Area} \tag{1}$$

$$C_x = \frac{\sum_{i=0}^{k} (x_i + x_{i-1})^2 (y_i - y_{i-1})^2}{Area}, C_y = \frac{\sum_{i=0}^{k} (y_i + y_{i-1})^2 (x_i - x_{i-1})^2}{Area} \tag{2}$$

- In roughly-rounded objects, a centroid is put at the same position at the center of the three-spoke type symmetry axes.

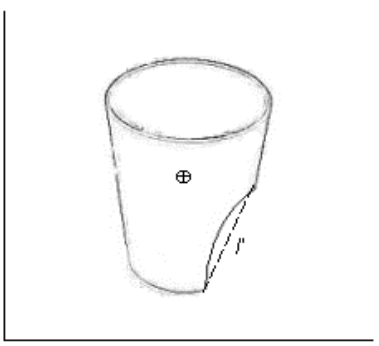


Fig. 3.6 The Occlusion of a Cup: Get a centroid after re-captured a shape.

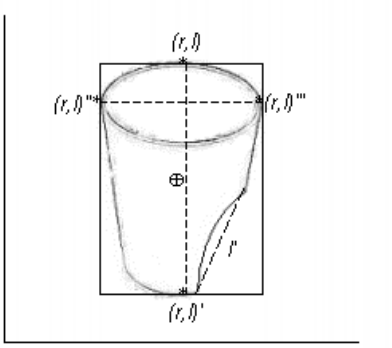


Fig. 3.7 Get extremal points *(r,l), (r,l)'* and *(r,l)'',(r,l)'''* and the maximum diameter of an object.

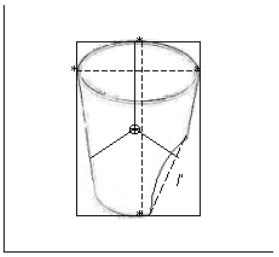


Fig. 3.8 Use the Three-Spoke Type Symmetry: Match a center of the spoke to a centroid and parallel one of axes to the maximum diameter

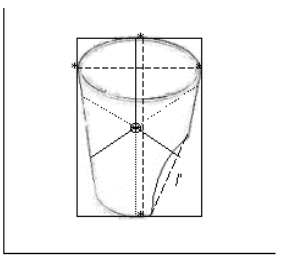


Fig. 3.9 Extend axes and make symmetry axes.

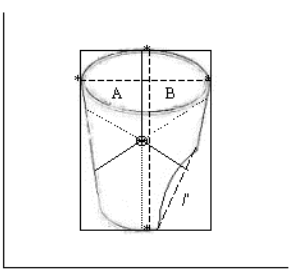


Fig. 3.10 Select a symmetry axis based on two regions, which are A and B.

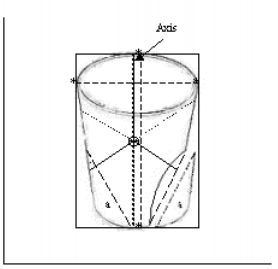


Fig. 3.11 Find a region a' of occluded shape using a symmetry axis and add to an occluded shape.

- Axis Detection: The midpoint of the major axis is called the center of the ellipse. The minor axis is the line segment perpendicular to the major axis which also goes through the center and touches the ellipse at two points. In elliptical objects, we detect a symmetry axis based on the maximum diameter or the minimum diameter. To find a symmetry axis in roughly-rounded objects, one of axes of the three-spoke type symmetry axes is in parallel with the maximum diameter of an object as shown in Figure 3.8.

Based on occluded shape and region, we select a symmetry axis to estimate this region within an object. Figures 3.9 and 3.10 show how to select a symmetry axis. When we select an axis in roughly-rounded objects, we consider conditions as follows:

- Select axes, which don't intersect the occluded region.
- Figures 3.9 and 3.10 show how to select a symmetry axis. Select axes, which have a region with the maximum diameter $\geq l'$.
- Area and perimeter are invariants as in equation 3, compare the proportion of region A and B.

$$\left(\frac{Perimeter}{Area}\right)^{A} \cong \left(\frac{Perimeter}{Area}\right)^{B} \tag{3}$$

- Using mirror symmetry, we can get points across an axis. We find points on the contour across an axis which have the same length l' and the same angle corresponding to the axis that is perpendicular to a symmetry axis, but the distance between axis and points may or may not be the same.
- Capture a region a', move the captured region to the occluded shape using the mirror symmetry, and add to these regions as shown in Figure 3.4, 3.5, and 3.11.
- Re-compute shape measurements such as area, diameters, and perimeter using RLC lines from re-captured shape of an object. Then, re-compute shape parameters based on measurements.
- Apply to a classifier as proposed in (Cho & Choi, 2004).

From the above discussions, we described how to reconstruct and estimate the partially occluded shape and region of an object and how to find the best matching class of partially occluded objects after the estimation.

4. Experimental Results

In the sections, we evaluate and describe the results of partial object recognition by the proposed method. We have selected 190 partially occluded objects of images from electronic catalogs on the Internet as well as manipulated images. We assume that occluded objects have more than 50% visibility of objects, and images of catalogs contain partially occluded objects. The objects are categorized by semantic meanings such as cup and plate. In addition, a proposed approach and experiments are limited to cups and plates since we use roughly-rounded or elliptical objects. More precisely, the database contains 32 objects from different viewpoints and images of 97 objects comprising image plane rotations and scale changes.

In sample images, we have extracted image features of partially occluded objects such as shape and texture. We experimented with shape reconstruction based on the contour of

objects using symmetry properties. We assumed that inputs are not correctly classified and have occlusion.

We experimented with samples such as plates and cups to reconstruct the occluded shape of objects as shown in Figure 4.1 and 4.2. In Figure 4.2, it is correctly classified after the reconstruction with an occlusion about 30%. On the other hand, Figure 4.1 is not correctly classified after the reconstruction since the width of plate is too narrow. This experiment shows that our method heavily relies on shape of objects.

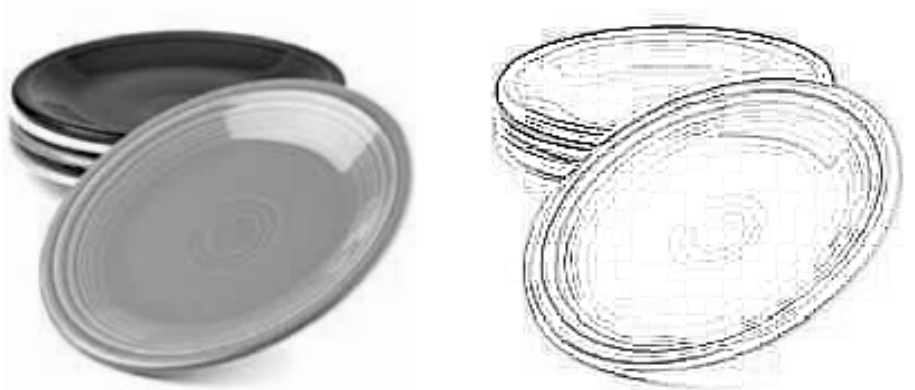


Fig. 4.1 Example of the occlusion with a plate.

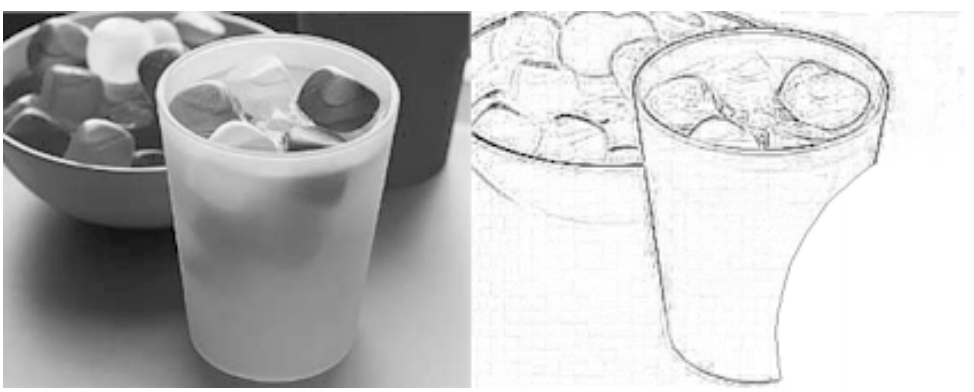


Fig. 4.2 Example of the manipulated occlusion with a Cup.

We performed an experiment for the relationships between visible portion of objects and recognition rates. In order to evaluate the visibility of objects, we used manipulated images of cups and plates. Figure 4.3 shows the pattern of object recognition in the presence of partial occlusion of objects and the results obtained by the symmetric recognition. A visible portion of approximately 67% is sufficient for the recognition of objects based on the contour.

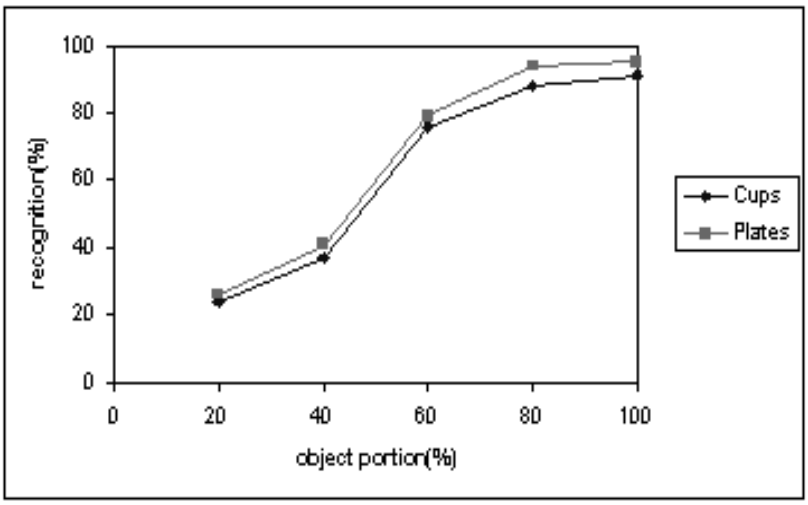


Fig. 4.3 Object recognition in the presence of the occlusion of objects based on the contour.

There are many efforts in object recognition for dealing with occlusion. The visible portion of objects required to recognize occluded objects are shown in Table 4.1. Table 4.1 shows a simple comparison between a proposed method and other existing methods. The probabilistic method based on local measurements requires small portions of objects to recognize the whole objects, but it required extensive training processes to recognize occluded objects. A proposed method show good visibility of partial object recognition and do not need extensive training processes.

Methods	**Visibility**	**Training processes**
Appearance matching techniques using adaptive masks	90%	not required
Probabilistic technique using Chi-square	72%	required
Probabilistic technique using local measurements	34%	required
Contour-based approach using symmetry	67%	not required

Table 4.1 The visibility of object recognition in the presence of partial occlusion.

In order to measure the influence of occlusion and compare its impact on the recognition performance of the different methods, we performed an experiment as follows. Figure 4.4 summarizes the recognition results for different visible object portions. For each test object,

we varied the visible object portion from 20% to 100% and recorded the recognition results using Chi-square divergence and a proposed method.

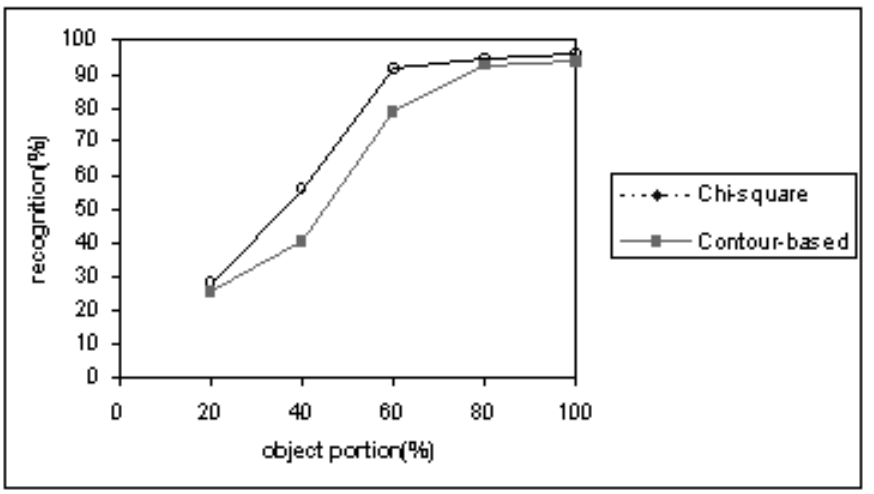


Fig. 4.4 Experimental results with occlusion.

The results show that a proposed method clearly obtains better results than Chi-square divergence. Using only 60% of the object area, almost 80% of the objects are still recognized. This confirms that a proposed method is capable of reliable recognition in the presence of occlusion.

Methods	**Occlusion**	**Scale changes**	**Object Pose**	**Rotation**
(Bischof & Leonardis, 1998)	Yes	Yes	No	No
(Edwards & Murase, 1997)	Yes	Yes	No	Yes(limited)
(Ohba & Ikeuchi,1997)	Yes	No	Yes	No
(Rao, 1997)	Yes	No	Yes	No
(Jacobs & Basri, 1997)	Yes	No	Yes	No
(Krumm, 1997)	Yes	No	No	No
Contour-based Method	Yes	Yes	Yes(limited)	Yes

Table 4.2 Summary of Object Recognition Methods for dealing with Occlusion.

Table 4.2 summarizes the various object recognition methods. The table indicates whether the methods can handle occlusion, rotation, pose, and changes in the size of objects in the database. Unlike the other methods, a proposed method can handle scale change, object pose, and rotated objects with occlusion, even though a proposed method has minor limitations of object poses.

5. Conclusion

In this paper, we have discussed how to estimate parameters and to reconstruct the occluded shape of partial elliptical objects in image databases. In order to reconstruct occluded shapes, we used mirror symmetry, which provides powerful method for the partial object recognition. Unlike the existing methods, a proposed method tried to reconstruct occluded shapes and regions within objects, since most objects in a domain have symmetrical figures. However, we have limitations in the shape of objects and the occluded region of objects. For example, if a pan has an occlusion in handle, it cannot correctly reconstruct and be recognized.

Another minor limitation of a proposed method is that it is sensitive to the pose of an object. For example, if we cannot see an ellipse due to the object's pose, we cannot recognize the object. After estimation, we have applied inputs, which include estimated parameters, to the existing classification trees, to get to the best matching class.

All experiments are performed based on the classifier in earlier work. In experiments, the results show that the recognition of the occluded object is properly reconstructed, estimated, and classified, even though we have limited to the size of samples. In addition, we have experienced the power of the symmetry through experiments.

6. References

Adam, N.; Cho, J. & Gangopadhyay, A. (2000). Feature Extraction for Content-based Image search in Electronic Commerce. *Proceedings of the MIS/OA International Conference,* pp. 513-517, June 2000.

Bischof, H. & Leonardis, A. (1998). Robust Recognition of Scaled Eigenimages through a Hierachical Approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 664-670, June 1998.

Blum, H. & Nagel, R. N. (1978). Shape Description using Weighted Symmetric Axis Features. *Pattern Recognition,* Vol. 10, No. 3, (May 1978) 167-180, ISSN:0031-3203.

Boshra, M. & Bhanu, B. (2000). Predicting Performance of Object Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 22, No. 9, (September 2000) 956-969, ISSN:0162-8828.

Carmichael, O. & Hebert, M. (2004). Shape-Based Recognition of Wiry Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 26, No. 12, (December 2004) 1537–1552, ISSN:0162-8828.

Chang, P. & Krumm, J. (1999). Object Recognition with Color Cooccurrence Histograms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 504-510, June 1999.

Cho, J.; Kim, H. & Choi, J. (2004). Object Classification based on the Probabilities of Pre-Assigned Intervals. *Proceedings of the Conference on Information and Knowledge Engineering,* pp. 49-55, June 2004.

David, P. & DeMenthon, D. (2005). Object Recognition in High Clutter Images Using Line Features. *Proceedings of the IEEE International Conference of Computer Vision*, pp. 1581- 1588, October 2005.

Edwards, J. & Murase, H. (1997). Appearance Matching of Occluded Objects using Coarse-to-fine Adaptive Masks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 533-539, June 1997.

Ho, C. & Chan, L. (1995). A Fast Ellipse/Circle Detector using Geometric Symmetry. *Pattern Recognition,* Vol. 28, No. 1, (January 1995) 117-124, ISSN:0031-3203.

Jacobs, D. W. & Basri, R. (1997). 3D to 2D Recognition with Regions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 547-553, June 1997.

Jones, G. & Bhanu, B. (1999). Recognition of Articulated and Occluded Objects. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 7, (July 1999) 603–613, ISSN:0162-8828.

Krumm, J. (1997). Object Detection with Vector Quantized Binary Features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 179-185, June 1997.

Leonardis, A. & Bishof, H. (1996). Dealing with Occlusions in the Eigenspace Approach. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 453-458, June 1996.

Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. *Proceedings of the International Conference on Computer Vision*, pp. 1150-1157, September 1999.

Mikolajczyk, K.; Zisserman, A. & Schmid, C. (2003). Shape Recognition with Edge-Based Features. *Proceedings of the British Machine Vision Conference*, 2, pp. 779-788, September 2003.

Ohba, K. & Ikeuchi, K. (1997). Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects. *IEEE Transaction of Pattern Analysis and Machine Intelligence,* Vol. 19, No. 9, (September 1997) 1043-1048, ISBN:0162-8828.

Rajpal, N.; Chaudhury, S. & Banerjee, S. (1999). Recognition of Partially Occluded Objects using Neural Network based Indexing. *Pattern Recognition*, Vol. 32, No. 10, (October 1999) 1737-1749, ISSN:0031-3203.

Rao, R. (1997). Dynamic Appearance-based Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 540-546, June 1997.

Russ, J. C. (1998). *The Image Processing Handbook*, 3rd, CRC Press, Inc., Boca Raton, FL.

Schiele, B. & Pentland, A. (1999). Probabilistic Object Recognition and Localization. *Proceedings of the the International Conference on Computer Vision,* pp. 177-182, September 1999.

Wu, W. & Wang, M. J. (1993). Elliptical Object Detection by using its Geometrical Properties. *Pattern Recognition*, Vol. 26, No. 10, (October 1993) 1499-1509, ISSN:0031-3203.

Williams, L. R. (1997). Topological Reconstruction of a Smooth Manifold-Solid from its Occluding Contour. *International Journal of Computer Vision,* Vol. 23, No. 1, (May 1997) 93-108, ISSN:0920-5691.

Williams, L. & Hanson, A. (1996). Perceptual Completion of Occluded Surfaces. *Computer Vision and Image Understanding*, Vol. 64, No. 1, (July 1996) 1–20, ISSN: 1077-3142.

24

Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics

Peng-Yeng Yin
Department of Information Management, National Chi Nan University
Taiwan

1. Introduction

Representation of digital planar curves is an important step prior to many image analysis tasks, such as object recognition, image matching, target tracking, etc. Polygonal approximation is an important technique to digital curve representation since the main information of curves is preserved at the corner points, it is desired to approximate a digital curve by an appropriate polygon to reduce the memory storage and the processing time for subsequent analyses. The design of a polygonal approximation algorithm not only impacts on the compression ratio of the data volume but also affects the accuracy of the subsequent image analysis tasks. There are several possible criteria with which the polygonal approximation can be performed, one of the most broadly used can be described as "given a digital curve and an error tolerance, the algorithm approximates the curve with a polygon by taking a subset of the points on the curve as the vertices such that the number of vertices is minimized and the approximation error between the curve and the corresponding polygon is no more than the error tolerance." (Yin, 2006)

An exact method to the polygonal approximation problem is impractical due to the intensive computations involved. An attempt using the dynamic programming technique had been made (Dunham, 1986), however, it required a worst-case complexity of $O(N^4)$ where N is the number of data points. Early solutions to reduce the amount of computations rely on local search heuristics, namely the sequential scan-along approaches (Wall & Danielsson, 1984; Ray & Ray, 1993), split-and-merge approaches (Ansari & Delp, 1991; Ray & Ray, 1995), and dominant point detection approaches (Teh & Chin, 1989; Zhu & Chirlian, 1995). However, the quality of the approximation result depends upon the initial condition where the heuristics take place and the metric used to measure the curvature.

Metaheuristics are alternatives to solve complex combinatorial optimization problems. Fred Glover first coined the term *metaheuristic* as a strategy that guides another heuristic to search beyond the local optimality such that the search will not get trapped in local optima. Metaheuristics combine two components, an exploration strategy and an exploitation heuristic, in a framework. The exploration strategy searches for new regions, and once it finds a good region the exploitation heuristic further intensifies the search for this area. In this context, metaheuristics encompass several well-known approaches such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), scatter search (SS), ant colony optimisation (ACO), particle swarm optimisation (PSO), just to name a few. Most of the

central metaheuristics have been applied to the polygonal approximation problems and attained promising results. Instead of describing all the methods, this chapter will focus on the more recently proposed metaheuristics, ACO and PSO, and give their comparative evaluations.

The remainder of this chapter is organized as follows. Section 2 presents the formulation of the polygonal approximation problem. Section 3 renders the details of the ACO- and the PSO-based methods. In Section 4, we present the experimental results and discussions. Finally, a conclusion is given in Section 5.

2. Problem Formulation

Given a digital curve represented by a set of N points, S = $\{x_0, x_1, ..., x_{N-1}\}$ where $x_{(i+1) \bmod N}$ is considered as the succeeding point of x_i. We define arc $\widehat{x_i x_j}$ as the collection of those points between x_i and x_j, and chord $\overline{x_i x_j}$ as the line segment connecting x_i and x_j. If we approximate $\widehat{x_i x_j}$ by $\overline{x_i x_j}$, the incurred approximation error, denoted by $e(\widehat{x_i x_j}, \overline{x_i x_j})$, can be measured by any distance norm; for here, the L_2 norm, i.e., the sum of squared perpendicular distance from every data point on $\widehat{x_i x_j}$ to $\overline{x_i x_j}$, is adopted. Thus a polygon with the vertex set T = $\{x_{p_0}, x_{p_1}, \ldots, x_{p_{M-1}}\}$, where T ⊂ S and 3 ≤ M ≤ N, can approximate the given curve with a total error E = $\sum_{i=0}^{M-1} e\left(\widehat{x_{p_i} x_{p_{(i+1) \bmod M}}}, \overline{x_{p_i} x_{p_{(i+1) \bmod M}}}\right)$, and our aim is to construct a polygon with the minimal vertex set and the incurred approximation error is less than the pre-specified tolerance. Formally, the polygonal approximation problem can be formulated as follows.

$$\arg\min_{T \subset S} |T| \qquad \text{subject to } 3 \le |T| \le \text{N and E} \le \varepsilon, \tag{1}$$

where $|T|$ denotes the cardinality of T and ε is the pre-specified error tolerance.

3. Polygonal Approximation Using Metaheuristics

Metaheuristics have shown many successful applications in diverse domains and the effectiveness and the malleability of metaheuristics are proven to be significantly better than most of the traditional local search heuristics. Metaheuristics are attractive to researchers because of their common features: natural metaphor, adaptivity, parallelism, easy implementation, and high quality result. In the following we illustrate the polygonal approximation application using two state-of-the-art metaheuristics: ant colony optimization (ACO) and particle swarm optimization (PSO).

3.1 ACO-based method

The basic framework of ant colony optimization (ACO) was first introduced in Dorigo's Ph.D. dissertation (Dorigo, 1992). Since then many ACO applications have been investigated such as the travelling salesman problem (Dorigo & Gambardella, 1997), quadratic assignment problem (Maniezzo et al., 1994), and combined heat and power economic dispatch problem (Song et al., 1999). The ACO is inspired by the research on the real ant

behavior. Ethologists observed that ants are able to construct the shortest feasible path from their colony to the feeding source by the use of pheromone trails. An ant leaves some quantities of pheromone on the ground and marks the path by a trail of this substance. The next ant then senses the pheromone laid on different paths and chooses one with a probability proportional to the amount of pheromone on it. The ant traverses the chosen path and leaves its own pheromone. This is an autocatalytic (positive feedback) process which favors the path along which more ants previously traversed. To apply the ACO to circumvent the problem, we need to define the path space and the pheromone field that play central roles in the algorithm (Yin, 2003).

3.1.1 Graph representation

Ideally, we can construct a graph G = <S, E*>, where S is the set of data points on the given curve and E* is the ideal edge set that has the desired property that any closed circuit through E* which originates and ends at the same node represents a feasible solution to the problem, i.e., the polygon consisting of the edges and nodes along the closed circuit should approximate the curve with E ≤ ε. However, it is impossible to generate E* in practice. An alternative is to generate a pseudo-ideal edge set $\hat{E}$, such that, E*$\subseteq \hat{E}$. For the constructed circuits which violate E ≤ ε, we can decrease the intensity of pheromone trails on the circuits to make them less attractive. $\hat{E}$ is constructed as follows. First, an empty edge set is created, i.e., $\hat{E} = \varnothing$. For every node $x_i \in S$, we examine each of the remaining nodes, $x_j \in S$, in clockwise order. The directed edge $\overrightarrow{x_i x_j}$ is added to $\hat{E}$ if the approximation error between the arc $\widehat{x_i x_j}$ and the line segment $\overline{x_i x_j}$ is no more than ε. The reason for using a directed edge is to avoid the ants walking backward. Now, the problem of polygonal approximation is equivalent to finding the shortest closed circuit on the directed graph G = <S, $\hat{E}$> such that E ≤ ε.

For the convenience of presentation, we define some notations as follows. Let the closed circuit completed by the *k*th ant be denoted $tour_k$, the number of nodes visited in $tour_k$ be $|tour_k|$, and the approximation error between the original curve S and the approximating polygon corresponding to $tour_k$ be $E(S, tour_k)$.

3.1.2 Starting node selection

Each ant chooses a starting node in the graph and sequentially constructs a closed path to finish its tour during each iteration. We establish a selection table for the starting node which is a linear array of N entries denoted by T_i, i = 1, 2, …, N. Initially, we let each T_i = 1. The probability with which the *i*th node is chosen as a starting node, denoted $Select_i$, is estimated as the entry value T_i divided by the sum of all entry values, $Select_i = T_i \Big/ \sum_{j=1}^{N} T_j$.

The ties with respect to $Select_i$ are broken randomly. Apparently, at the beginning of the first cycle, every node has equal probability of being chosen as a starting node since $Select_i = 1/N$. We then update the entry value of the selection table at the end of each cycle.

Let the set of ants which start with the ith node at the current cycle be Ant_Start$_i$, and the size of Ant_Start$_i$ be $|Ant_Start_i|$. We update entry T$_i$ based on a trade-off between the average quality of current solutions constructed by those ants in Ant_Start$_i$ and the value of Select$_i$ derived from older cycles. Thus, we let

$$T_i \leftarrow \begin{cases} \dfrac{(1-r)}{|Ant_Start_i|} \sum_{j \in Ant_Start_i} \dfrac{1}{|tour_j|} + r\, Select_i, & \text{if the } i\text{th node was chosen as a starting node at current cycle} \\ T_i, & \text{otherwise,} \end{cases} \tag{2}$$

where $r \in (0,1)$ is the parameter which controls the relative contribution of each component.

3.1.3 Node transition rule

The node transition rule is a probabilistic one determined by the pheromone intensity τ_{ij} and the visibility value η_{ij} of the corresponding edge. In the proposed method, τ_{ij} is equally initialized to $1/N$ (actually, any small constant positive value will suffice), and is gradually updated at the end of each cycle according to the average quality of the solutions that contain this edge. On the other hand, the value of η_{ij} is determined by a greedy heuristic which encourages the ants to walk to the farthest accessible node in order to construct the longest possible line segment in a hope that an approximating polygon with fewer vertices is obtained eventually. This can be accomplished by setting $\eta_{ij} = |\widehat{x_i x_j}|$, where $|\widehat{x_i x_j}|$ is the number of points on $\widehat{x_i x_j}$. The value of η_{ij} is fixed during all the cycles since it considers local information only.

We now define the transition probability from node i to node j through directed edge $\overrightarrow{x_i x_j}$ as

$$p_{ij} = \frac{(\tau_{ij})^{\alpha} (\eta_{ij})^{\beta}}{\sum_{\substack{\forall\, \overrightarrow{x_i x_h} \\ \text{from } x_i}} (\tau_{ih})^{\alpha} (\eta_{ih})^{\beta}}. \tag{3}$$

Also, the ties with respect to p_{ij} are broken randomly.

3.1.4 Pheromone Updating Rule

The intensity of pheromone trails of an edge is updated at the end of each cycle by the average quality of the solutions that traverse along this edge. In particular, the pheromone intensity at directed edge $\overrightarrow{x_i x_j}$ is updated by

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \max(\sum_{k=1}^{m} \Delta\tau_{ij}^{k}, 0), \tag{4}$$

where $\rho \in (0,1)$ is the persistence rate of previous pheromone trails, and $\Delta\tau_{ij}^{k}$ is the quantity of new trails left by the kth ant and it is computed by

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1}{|tour_k|}, & \text{if } \overrightarrow{x_i x_j} \in tour_k \\ & \text{and } E(S, tour_k) \le \varepsilon; \\ -\dfrac{E(S, tour_k)}{\varepsilon N}, & \text{if } \overrightarrow{x_i x_j} \in tour_k \\ & \text{and } E(S, tour_k) > \varepsilon; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, more quantities of pheromone trails will be laid at the edges along which most ants have constructed shorter feasible tours. On the other hand, in the worst case, the edges will receive no positive rewards because either no ants walked through them or most passing ants constructed infeasible tours. As such, the proposed rule can guide the ants to explore better tours corresponding to high quality solutions.

3.2 PSO-based method

Particle swarm optimization (PSO) is a new metaheuristic developed in 1995 (Kennedy & Eberhart, 1995). It has exhibited effectiveness and malleability in many applications, such as evolving weights and structure for artificial neural networks (Eberhart & Shi, 1998), manufacture end milling (Tandon, 2000), and reactive power and voltage control (Yoshida et al., 1999). The development of PSO is inspired by the observation on the behaviors of bird flocking. A large number of birds flock synchronously, change direction suddenly, and scatter and regroup together. Each individual, called a particle, benefits from the experience of its own and that of the other members of the swarm during the search for food. The PSO models the social dynamics of flocks of birds and serves as an optimizer for nonlinear continuous functions. In order to deal with combinatorial optimization, the discrete version of PSO has also been introduced (Kennedy & Eberhart, 1997). However, in our experiments this discrete version does not show effective result for polygonal approximation problem. We conjecture that the deterioration is due to the linear combination of reference solutions which is often adopted in solving continuous function optimization. Thus, we add genetic features to enhance the search ability in combinatorial optimisation using the discrete PSO (Yin, 2006).

3.2.1 Particle representation and fitness evaluation

Since particles of the PSO correspond to candidate solutions of the underlying problem, we use the particle to represent the approximating polygon by a binary vector. For the *i*th particle, the corresponding representation is

$$P_i = \left(p_{i0}, p_{i1}, \ldots, p_{i(N-1)}\right) \quad \text{subject to } \sum_{j=0}^{N-1} p_{ij} \geq 3 \text{ and } p_{ij} \in \{0, 1\}, \tag{6}$$

where $p_{ij} = 1$ if x_j is one of the vertices chosen to represent the polygon, and $p_{ij} = 0$ otherwise. Thus, the particle representation indicates which data points constitute the vertex set T of the polygon and $\sum_{j=0}^{N-1} p_{ij} = |T|$.

The fitness of the particle is evaluated in two ways. If the approximation error entailed by a candidate polygon exceeds the specified error tolerance, i.e., $E > \varepsilon$, the fitness of the corresponding particle will be assigned a negative value to express the infeasibility degree of this candidate solution, else the particle fitness is set to the inverse of the sum of particle bit values to assess the solution quality in terms of the number of vertices. More precisely, the fitness of particle P_i is determined by

$$fitenss(P_i) = \begin{cases} -E/\varepsilon N & \text{if } E > \varepsilon, \\ 1/\sum_{j=0}^{N-1} p_{ij} & \text{otherwise.} \end{cases} \tag{7}$$

Therefore, there are two optimization goals in our setting. The first one is to move the particle from infeasible solution space to feasible regions, and the second one is to fly the particle to a new position which may result in a polygon with fewer vertices, i.e., with better merit in problem objective. The two optimization goals are pursued simultaneously since the PSO evolves with a swarm of particles and each of which may invoke different fitness evaluation depending on the entailed approximation error.

3.2.2 Genetic operations

PSO is a population-based search paradigm using a swarm of particles, it is natural to compare PSO with GA which is another population-based search algorithm and is well-known to the community. In PSO, each particle flies to a better position which is a randomized weighted sum of vectors based on its personal best (*pbest*) and the global best (*gbest*) positions, while in GA the quality of individual chromosome is improved by using two principal genetic operations: *selection* and *reproduction*. The selection operation picks the good individuals for survival to mimic the natural selection of the fittest and the reproduction operation provides a mechanism to exchange and recombine the information (building blocks) among good-quality individuals. The feature of genetic selection has been added to PSO for solving continuous function optimization problems (Angeline, 1998; Shigenori et al., 2003) and the experimental results demonstrated substantial improvement over the original version. In this chapter, we further devise the scheme for conducting the genetic reproduction with the discrete PSO.

Since the particle vector adjustment formulae are in fact a linear combination of critical vectors with quasi-random coefficients, the newly explored parameter values are bounded between experienced vectors to some extent. This is perhaps a desired property for continuous function value optimization problems, however, it hinders the solution exploration for discrete combinatorial optimization. For the latter one, the building blocks of

good quality solutions are segments of specific ordering or partial selections of elements, and the optimal solution may be obtained through recombination of those segments instead of a weighted sum of those values. Hence, we propose a new particle adjustment rule with genetic recombination for the *j*th bit of particle *i* as follows.

$$p_{ij} = w(0, w_1)rand(p_{ij}) + w(w_1, w_2)rand(pbest_{ij}) + w(w_2, 1)rand(gbest_j), \tag{8}$$

where $0 < w_1 < w_2 < 1$ and $w(\bullet)$ and $rand(\bullet)$ are the threshold function and the probabilistic bit flipping function, respectively, and they are defined as follows.

$$w(a,b) = \begin{cases} 1 & \text{if } a \le q_1 < b, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where $q_1 \in U(0, 1)$ is a randomly drawn real number. Therefore, only one of the three terms on the right hand side of Eq. (8) will remain depending on the value of q_1.

$$rand(y) = \begin{cases} (y+1) \text{ modulo } 2 & \text{if } q_2 \le t, \\ y & \text{otherwise.} \end{cases} \tag{10}$$

Thus, $rand(y)$ mutates the binary bit y with a small probability t (q_2 is another random number drawn from $U(0, 1)$). To relate the new particle adjustment rule to genetic reproduction, we analyze Eq. (8) in two aspects. *First*, the particle P_i derives its every single bit from either one of p_{ij}, $pbest_{ij}$, or $gbest_j$, this operation corresponds to a 3-way uniform crossover among P_i, $pbest_i$, and $gbest$, such that the particle can exchange building blocks (segments of ordering or partial selections of elements) with personal and global experiences. *Second*, each bit attained in this way will be flipped with a small probability, analogous to the binary mutation performed in genetic algorithms. As such, the genetic reproduction, in particular, the crossover and mutation, have been added to the discrete PSO, and this new version is very likely more suitable to solve combinatorial optimization problems than the original one.

3.3 Hybrid strategy

Metaheuristics combine two elements, *exploration* and *exploitation*, in a framework. The exploration strategy searches for new regions, and once it finds a good region the exploitation heuristic further intensifies the search for this area. However, since the two strategies are usually inter-wound in the algorithm, the search is conducted to other regions before it finds the local optima. Many researchers have suggested to employ a hybrid strategy which embeds a local optimizer such as hill-climbing in between the iterations of the metaheuristics to enhance the searching ability. In the light of this, we propose to embed a local heuristic into the ACO- and the PSO-based approaches. To save the computational efforts, the local heuristic is only applied to the best candidate solution observed so far at each iteration.

The local heuristic, named the segment-adjusting-and-merging, takes into account the problem-specific knowledge that the approximation error may be further reduced if the positions of the vertices of the polygon are appropriately adjusted, and that the number of vertices is decreased if we merge two adjacent segments under the constraint that the

resulting new polygon still satisfies the error tolerance. The two solution-improving processes are performed repeatedly until the number of vertices cannot be further decreased.

4. Experimental Results and Discussions

In this section, we present the computational results and evaluate the performance of the algorithms. The platform of the experiments is a PC with a 1.8 GHz CPU and 192 MB RAM. The algorithms are coded in C++. A number of benchmark curves borrowed from relevant literature are used for testing.

4.1 Benchmark curves

Three synthesized benchmark curves (see Fig. 1) and two real image curves (see Figs. 2-3) which are broadly used in the literature to evaluate various algorithms for polygonal approximation are included in our experiments for testing. As such the readers can easily compare the proposed algorithms with existing works. Fig. 1(a) is a leaf curve with 120 points, Fig. 1(f) is a chromosome curve with 60 points, Fig. 1(k) is a semi-circle curve with 102 points, Fig. 2(a) is a plane contour image with 682 edge points, and Fig. 3(a) is a fish contour image with 700 edge points.

4.2 Competing metaheuristics

In addition to evaluating the ACO-based and the PSO-based algorithms presented in Section 3, we compare the results with those obtained using two other major metaheuristics: GA (Yin, 1999) and TS (Yin, 2000). The GA-based approach used the same solution representation scheme as that of the PSO-based mthod (see Eq. (6)). It applied a fitness function as k - $\sum_{j=0}^{N-1} p_{ij}$ - max(E-ε, 0) where k is a constant. Besides using the traditional genetic operators (selection, crossover, mutation), a learning strategy is employed to improve the best chromosome observed so far at each iteration. The TS-based approach also followed Eq. (6) to generate its solution configuration. Three kinds of moves are defined: vertex-addition, vertex-deletion, and vertex-adjustment. As such the bounded neighborhood space is well defined. The tabu moves are enforced in order to prevent the current solution configuration getting into a subregion already visited. However, appropriate aspiration criteria are applied to resume a tabu move when it results in a better solution status than the ones observed so far.

4.3 Comparative performances

All of these metaheuristics have been proved to significantly outperform traditional local heuristics in solving the polygonal approximation problem (Yin, 2003; Yin, 2006). We thus focus our comparison among these metaheuristics only. The experiments on the three synthesized curves using the competing metaheuristics are shown in Table 1. As these metaheuristics are stochastic and each separate run of the same program may yield a different result, we report the average number of vertices (M) on the finally obtained polygon and the average consumed times in seconds (t) over 10 independent runs. The standard deviation (σ_M) of M is calculated for measuring the stability of the metaheuristics. It is evident from Tables 1 that the ACO- and the PSO-based approaches have better

performance than those of the GA-based and the TS-based approaches in terms of minimizing the value of M. This is due to the fact that the ACO- and the PSO-based methods further intensify the search in the neighborhood of the best solution observed so far using the hybrid strategy. All of the four competing metaheuristics have small values of σ_M, this means that these methods are all malleable against various curves with different properties. As for the computational times, all of these methods can derive quality results very quickly because the number of data points on the curves is small.

Fig. 1 shows the visualization of the finally obtained approximating polygons with their specified error tolerance (ε) and the number of yielded vertices (M) using various metaheuristics. It is seen that GA and TS yield worse approximating polygons with redundant vertices while ACO and PSO produce the least number of vertices but still preserving the main corner information.

		GA		TS		ACO		PSO	
	ε	M (σ_M)	*t*	M (σ_M)	*t*	M (σ_M)	*t*	M (σ_M)	*t*
Leaf (N=120)	150	15.6 (0.6)	0.4	10.6 (0.5)	0.1	11.0 (0.0)	0.9	10.7 (0.5)	0.4
	100	16.3 (0.5)	0.3	13.7 (0.6)	0.1	12.6 (0.2)	0.8	12.4 (0.5)	0.3
	90	17.3 (0.5)	0.3	14.6 (0.5)	0.1	12.8 (0.3)	0.9	13.0 (0.0)	0.3
	30	20.5 (0.6)	0.3	20.1 (0.5)	0.1	16.6 (0.4)	0.9	16.6 (0.5)	0.3
	15	23.8 (0.6)	0.3	23.1 (0.5)	0.1	19.7 (0.3)	0.9	20.0 (0.0)	0.2
Chromosom (N=60)	30	7.3 (0.4)	0.2	6.7 (0.4)	0.1	6.0 (0.0)	0.4	6.0 (0.0)	0.2
	20	9.0 (0.6)	0.2	8.0 (0.3)	0.1	7.6 (0.3)	0.5	7.6 (0.7)	0.2
	10	10.2 (0.4)	0.2	11.0 (0.4)	0.1	10.0 (0.3)	0.5	10.5 (0.5)	0.1
	8	12.2 (0.5)	0.2	12.2 (0.5)	0.1	11.0 (0.4)	0.5	11.0 (0.0)	0.1
	6	15.2 (0.6)	0.2	14.4 (0.5)	0.1	12.2 (0.3)	0.5	12.4 (0.7)	0.1
Semicircle (N=102)	60	13.2 (0.4)	0.3	11.0 (0.4)	0.1	10.0 (0.0)	0.8	10.0 (0.0)	0.3
	30	13.9 (0.7)	0.3	13.6 (0.5)	0.1	12.0 (0.0)	0.8	12.1 (0.3)	0.3
	25	16.8 (0.7)	0.3	14.9 (0.6)	0.1	13.0 (0.0)	0.7	13.2 (0.4)	0.3
	20	19.2 (0.6)	0.3	16.2 (0.6)	0.1	15.8 (0.4)	0.7	14.6 (0.7)	0.2
	15	23.0 (0.9)	0.3	18.3 (0.7)	0.1	16.8 (0.4)	0.7	15.8 (1.2)	0.2

Table 1. The comparative results on synthesized curves using competing metaheuristics

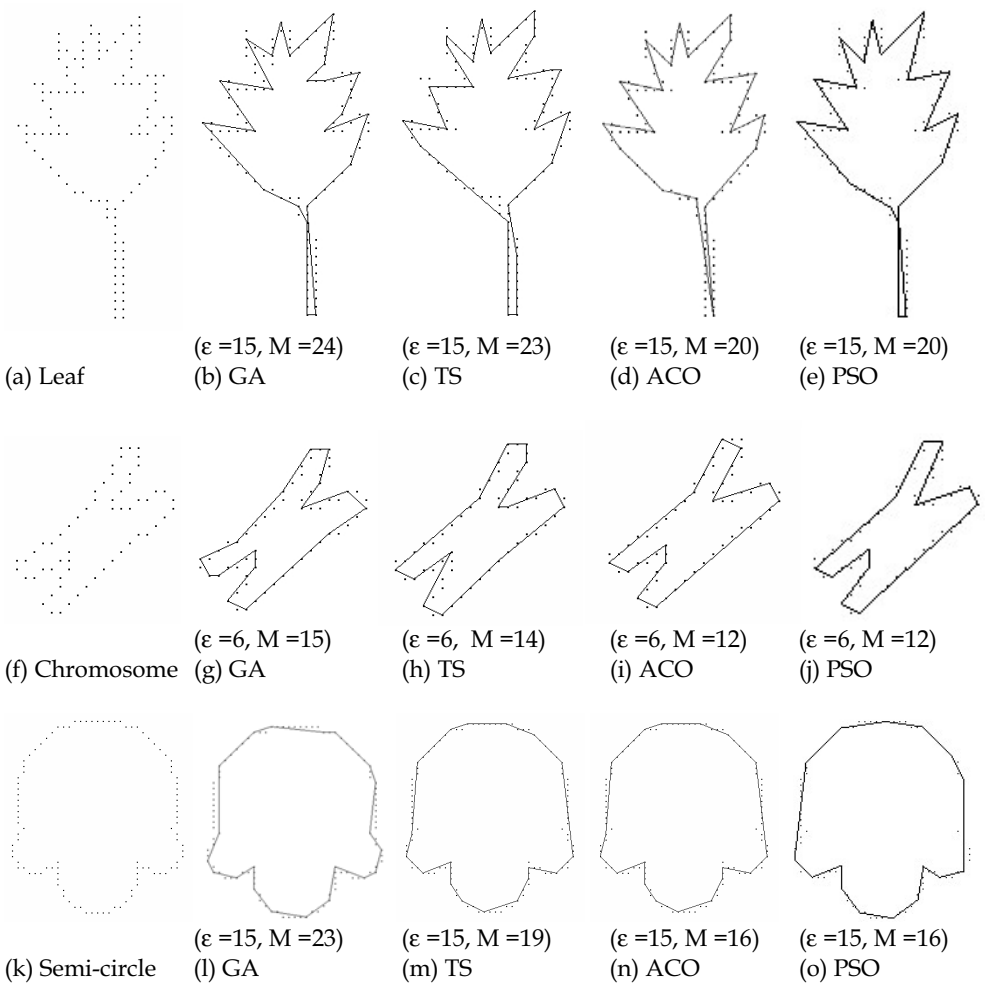


Fig. 1. Finally obtained approximating polygons on the synthesized curves with their specified error tolerance (ε) and the number of yielded vertices (M) using various metaheuristics

To demonstrate the feasibility of the metaheuristics for real-world applications, two real images containing a symbol of a plane and a fish, respectively, are further experimented with. The two images are binarized and the contour edge points are extracted by detecting the black-white transitions (see Figs. 2(a) and 3(a)). By specifying various values of error tolerance, the comparative performances obtained using the competing metaheuristics are summarized in Table 2. It is observed that the performance of the GA- and the TS-based methods deteriorates in the two real applications as the error tolerance decreases where the numbers of polygon vertices are significantly greater than those obtained by the ACO- and

the PSO-based approaches. However, the TS-based approach is the fastest one because it only uses one seed solution to conduct the search path while the others are population-based searching methods.

Figs. 2-3 show the finally obtained approximating polygons with their specified error tolerance (ε) and the number of yielded vertices (M) using various metaheuristics. Similary, the ACO- and the PSO-based methods econimcally preserve the main corner information on the curve while the GA- and the TS-based methods may use multiple vertices to approximate some corners in a small region.

To justify the reason behind the performance difference observed, we disable the application of the hybrid strategy in the ACO- and the PSO-based approaches and reperform the experiments again. We found that the new results obtained by the ACO- and the PSO-based methods without hybrid strategy become comparable with that obtained by the GA- and the TS-based methods. Therefore, the problem-specific local heuristics such as the segment-adjusting-and-merging are the key-reason that results in the performance differences among these metaheuristics. It is worth further studying other appropriate problem-specific local heuristics, e.g., the scan-along search, split-and-merge process, and dominant-point detection, to be hybridized with these metaheuristics. Note that the learning strategy employed by the GA-based approach is a general strategy that may be not as efficient as the problem-specific heuristics in some complex problems but it is useful when the probelm-specific heuristics are not easy to design.

		GA		TS		ACO		PSO	
	ε	M (σ_M)	t	M (σ_M)	t	M (σ_M)	t	M (σ_M)	t
	3000	14.2 (0.8)	2.5	13.0 (0.3)	0.4	12.1 (0.4)	5.0	12.3 (0.5)	6.4
	2000	15.1 (0.9)	2.4	14.4 (0.6)	0.4	13.0 (0.2)	4.7	13.0 (0.0)	6.1
Plane	1000	17.4 (0.6)	2.3	16.7 (0.5)	0.4	14.0 (0.6)	4.8	15.3 (0.8)	5.6
(N=682)	500	21.3 (0.8)	2.2	19.6 (0.6)	0.4	17.8 (0.5)	4.5	17.4 (0.5)	5.3
	100	33.8 (0.9)	2.4	31.3 (0.5)	0.4	28.1 (0.7)	4.6	24.0 (0.6)	4.5
	4000	16.5 (0.5)	2.3	14.0 (0.3)	0.5	12.2 (0.4)	5.7	15.8 (0.4)	5.5
	3000	17.4 (0.6)	2.2	16.0 (0.3)	0.5	14.6 (0.3)	5.9	16.9 (0.3)	5.1
Fish	2000	22.1 (1.0)	2.2	21.2 (0.4)	0.4	17.1 (0.5)	5.5	18.6 (0.9)	4.9
(N=700)	1000	32.4 (0.9)	2.3	29.1 (1.0)	0.5	26.8 (0.7)	5.6	25.3 (0.5)	4.0
	500	37.0 (1.1)	2.4	35.9 (1.2)	0.4	34.8 (0.7)	5.6	32.8 (0.6)	3.4

Table 2. The comparative results on real image curves using competing metaheuristics

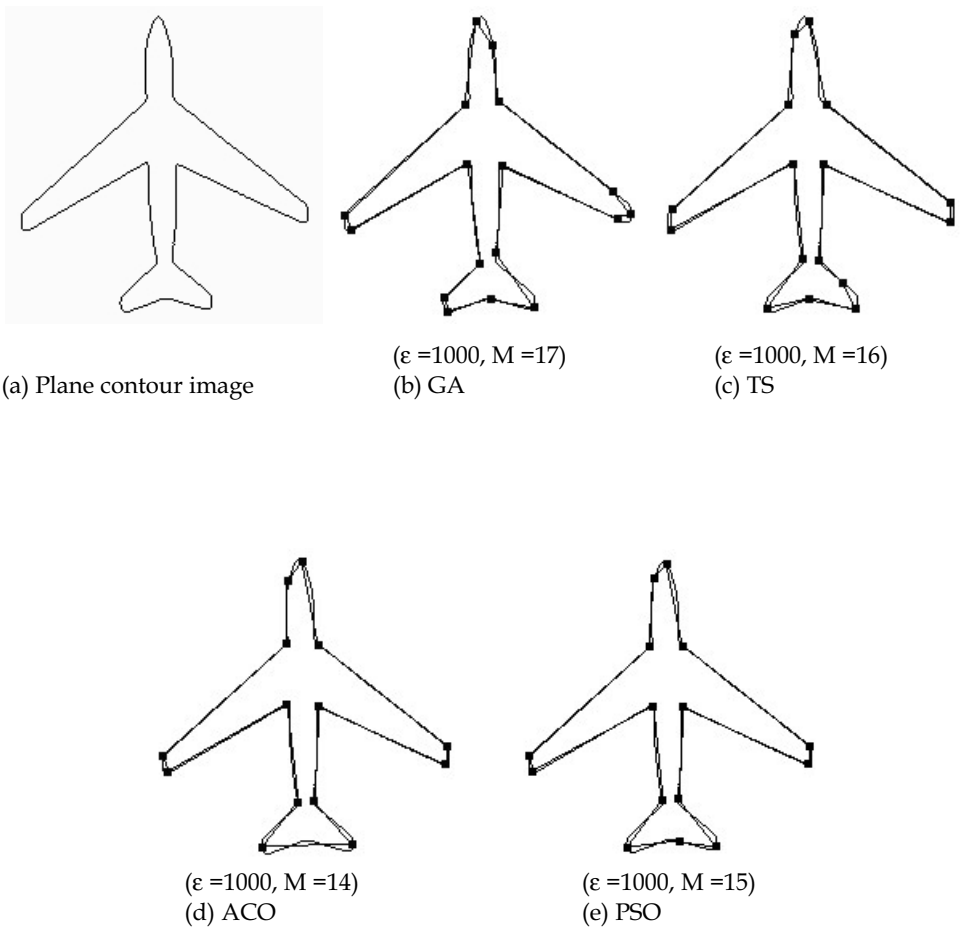
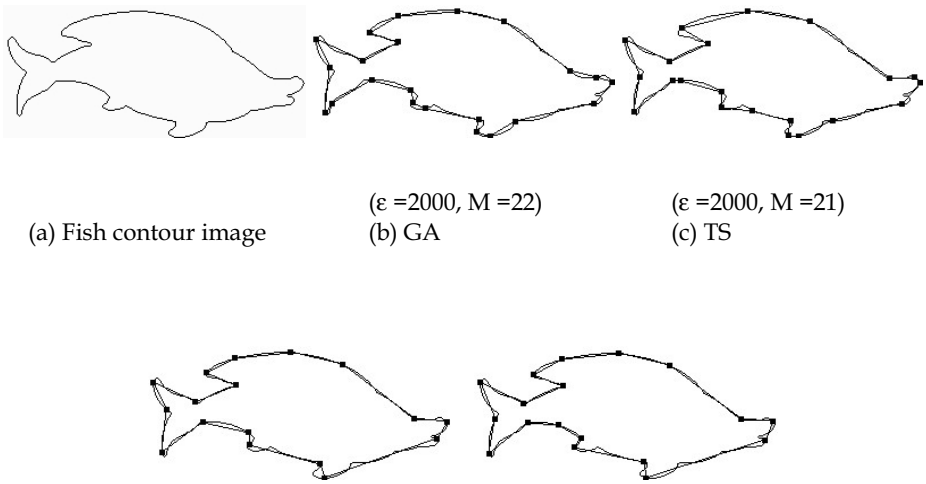


Fig. 2. Finally obtained approximating polygons one the plane image with their specified error tolerance (ε) and the number of yielded vertices (M) using various metaheuristics



(a) Fish contour image

(ε =2000, M =22)
(b) GA

(ε =2000, M =21)
(c) TS

(ε =2000, M =17)
(d) ACO

(ε =2000, M =18)
(e) PSO

Fig. 3. Finally obtained approximating polygons on the fish image with their specified error tolerance (ε) and the number of yielded vertices (M) using various metaheuristics

5. Conclusion

In this chapter, we investigate the polygonal approximation problem which is fundamental to many image analysis tasks. Traditional problem-specific heuristics are not suitable to be applied alone because the quality of the obtained result depends on the initial setting of the algorithms and the properties of the curves. On the other hand, metaheuristic approaches can produce stable approximation quality for various kinds of curves. We have illustrated the implementations based on two newly developed metaheuristics, namely the ACO and the PSO. To circumvent the underlying problem, specific features have been introduced such as the ACO graph representation, PSO genetic operators, penalty functions, and the hybrid strategy. Experimental results on several benchmark curves have manifested that these new features can improve the performance of metaheuristics in solving the polygonal approximation problem.

6. References

Angeline, P. (1998). Using selection to improve particle swarm optimization. *Proceedings IEEE Int'l. Conf. on Evolutionary Computation,* pp. 84-89

Ansari, N. & Delp, E. J. (1991). On detection dominant points. *Pattern Recognition*, Vol. 24, pp. 441-450

Dorigo, M. (1992). *Optimization, learning, and natural algorithms*. Ph.D. Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy

Dorigo, M. & Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transaction on Evolutionary Computation*, Vol. 1, pp. 53-66

Dunham, J. G. (1986). Optimum uniform piecewise linear approximation of planar curves, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 67-75

Eberhart, R. C. & Shi, Y. (1998). Evolving artificial neural networks, *Proceedings Int'l. Conf. on Neural Networks and Brain,* PL5-PL13

Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization, *Proceedings IEEE Int'l. Conf. on Neural Networks, IV,* , pp. 1942-1948

Maniezzo, V.; Colorni, A. & Dorigo, M. (1994). The ant system applied to the quadratic assignment problem. Universite Libre de Bruxelles, Belgium, Technical Report IRIDIA/94-28

Ray, B. K. & Ray, K. S. (1993). Determination of optimal polygon from digital curve using L1 norm. *Pattern Recognition*, Vol. 26, pp. 505-509

Ray, B. K. & Ray, K. S. (1995). A new split-and-merge technique for polygonal approximation of chain coded curves. *Pattern Recognition Letters*, Vol. 16, pp. 161-169

Shigenori, N.; Takamu, G.; Toshiku, Y. & Yoshikazu, F. (2003). A hybrid particle swarm optimization for distribution state estimation. *IEEE Transaction on Power Systems*, Vol. 18, pp. 60-68

Song, Y. H.; Chou, C. S. & Stonham, T. J. (1999). Combined heat and power economic dispatch by improved ant colony search algorithm. *Electric Power Systems Research*, Vol. 52, pp. 115-121

Tandon, V. (2000). Closing the gap between CAD/CAM and optimized CNC end milling, Master thesis, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis

Teh, C. H. & Chin, R. T. (1989). On the detection of dominant points on digital curves, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 859-872

Wall, K. & Danielsson, P. E. (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, Vol. 28, pp. 220-227

Yin, P.Y. (1999). Genetic algorithms for polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 13, pp. 1-22

Yin, P.Y. (2000). A tabu search approach to the polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, pp. 243-255

Yin, P.Y. (2003). Ant colony search algorithm for optimal polygonal approximation of plane curves. *Pattern Recognition*, Vol. 36, pp. 1783-1797

Yin, P.Y. (2006). Genetic particle swarm optimization for polygonal approximation of digital curves. *Pattern Recognition and Image Analysis*, Vol. 16, No. 2, pp. 223-233

Yoshida, H.; Kawata, K.; Fukuyama, Y. & Nakanishi, Y. (1999). A particle swarm optimization for reactive power and voltage control considering voltage stability, *Proceedings Int'l. Conf. on Intelligent System Application to Power Systems,* , pp. 117-121

Zhu, P. & Chirlian, P. M. (1995). On critical point detection of digital shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 737-748

25

Pseudogradient Estimation of Digital Images Interframe Geometrical Deformations

A.G. Tashlinskii
Ulyanovsk State Technical University
Russia

1. Introduction

Nowadays the systems of information extraction that include spatial apertures of signal sensors are widely used in robotics, for the remote exploration of Earth, in medicine, geology and in other fields. Such sensors generate dynamic arrays of data having the proper feature which is in their space-time correlation and due to which they can be represented in the form of multidimensional images (Gonzalez & Woods, 2002). When producing algorithmic software for the processing of such images it is necessary to take into account the dynamics of the scene to be observed, distortions caused by signal propagation environment, spatial movements of signal sensors and imperfection of their construction. The influence of the mentioned factors can be described through mathematical models of space-time deformations of multidimensional grids with the specified images.

The estimation of varying parameters of image spatial deformations is required not only in robotics applications, but also to solve a wide range of other problems, for example, for automated search of a fragment on the image, navigational tracking of mobile object course in the conditions of limited visibility, combination of multiregion images at remote explorations of Earth, in medical research. A lot of scientific calls for papers are devoted to different problems of image sequence space-time deformation parameters estimation (the bibliography is given, for example, in (Tashlinskii, 2000)). This chapter is devoted to one of the directions of solving this type of problems, where pseudogradient estimation of image interframe geometrical deformations (IIGDs) is considered.

Let us assume that the model of IIGDs is defined to an accuracy of a parameters vector $\overline{\alpha}$ and the estimation quality criterion is formulated in terms of some functional $\mathrm{J}(\overline{\alpha})$ minimization showing expected losses. However, it is impossible to find optimal parameters $\overline{\alpha}^*$ in the mentioned sense in view of incompleteness of description of the images to be observed. In this case we can estimate the parameters $\overline{\alpha}$ on the basis of a given realization Z analysis of the image to be observed by means of some adaptation procedure which minimizes $\mathrm{J}(\overline{\alpha}) = \mathrm{J}(\overline{\alpha}, Z)$ for the given realization Z. However, it is reasonable to avoid this intermediate state of the research and estimate $\overline{\alpha}$ directly on values $\mathrm{J}(\hat{\overline{\alpha}}, Z)$ (Polyak & Tsypkin, 1984):

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \nabla \mathrm{J}(\hat{\overline{\alpha}}_{t-1}, Z), \tag{1}$$

where $\hat{\overline{\alpha}}_t$ – next after $\hat{\overline{\alpha}}_{t-1}$ approximation of the minimum point of $J(\hat{\overline{\alpha}}, Z)$; $\mathbf{\Lambda}_t$ – gain matrix (positively defined matrix determining a value of the estimates change at the t-th iteration); $\nabla J(\hat{\overline{\alpha}}_{t-1}, Z)$ – gradient of the functional $J(\hat{\overline{\alpha}}_{t-1}, Z)$. The necessity of multiple cumbersome calculations hinders the procedure (1) application in the image processing. It is possible to significantly reduce computational expenses due to the usage of contraction $\nabla J(\hat{\overline{\alpha}}_{t-1}, Z_t)$ instead of $\nabla J(\hat{\overline{\alpha}}_{t-1}, Z)$ at some part Z_t of realization Z at each iteration choosing, for example, Z_t in the form of a sliding window. For relatively large-sized images, the analysis of approaches (Tashlinskii, 2000; Minkina et al., 2007) to the synthesis of algorithms of IIGDs estimation in real time showed that it is expedient to seek a decision, satisfying the requirements of simplicity, rapid convergence and efficiency in various real situations, among recurrent non-identification algorithms. The pseudogradient algorithms (PGAs) constitute the most representative class of such algorithms.

The conception of the pseudogradient was introduced in work (Polyak & Tsypkin, 1973). A unified approach to the analysis and synthesis of various procedures of the stochastic minimization has been developed on the basis of it. For the given problem to be solved the pseudogradient $\overline{\beta}_t$ may be represented as any random vector in the parameter space depending on a function of losses and estimates $\hat{\overline{\alpha}}_{t-1}$ at the t-th iteration if the following condition is satisfied

$$\left[\nabla J(\hat{\overline{\alpha}}_{t-1}, Z)\right]^T M\{\overline{\beta}_t\} \geq 0, \tag{2}$$

where T – sign of transposition; $M\{\cdot\}$ – symbol of the mathematical expectation. In the geometrical interpretation the vector $\overline{\beta}_t$ is the pseudogradient if it makes, on average, an acute angle with the exact value of the functional $J(\hat{\overline{\alpha}}, Z)$ gradient. The class of PGAs includes algorithms of stochastic approximation, random search and many others.

The following procedure is used in PGA (Tsypkin, 1995)

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \overline{\beta}_t, \tag{3}$$

where $\overline{\alpha}$ – vector of the parameters to be estimated; $t = \overline{1, T}$ – iteration number; $\hat{\overline{\alpha}}_0$ – initial approximation of the parameters vector; T – number of iterations. The algorithm is considered to be pseudogradient if $\overline{\beta}_t$ is the pseudogradient at each its iteration. In this case the iterations are, on average, performed in the direction of reduction of $J(\overline{\alpha})$ and sequence $\hat{\overline{\alpha}}_1, \hat{\overline{\alpha}}_2, \ldots, \hat{\overline{\alpha}}_t, \ldots$ converges to the optimal parameters when satisfying relatively weak conditions (Polyak, 1976).

If realizations α_t, $t = 1, 2, \ldots$, of the parameter α to be estimated are observed as, for example, in the problems of image brightness prediction, then we can choose $\beta_t = \hat{\alpha}_t - \alpha_t$ as the pseudogradient, where the estimate $\hat{\alpha}_t$ is found on realization Z or on a part of realization Z_t. In problems of image processing the quality functional $J(\overline{\alpha}, Z)$ is often expressed through the mathematical expectation of some function $f(\overline{\alpha}, Z)$:

$$\mathrm{J}(\overline{\alpha}, Z) = \mathrm{M}\{(f(\overline{\alpha}, Z))\}. \tag{4}$$

In particular, it can be mean square of error of some value χ:

$$f(\overline{\alpha}, Z) = (\hat{\chi}(\overline{\alpha}, Z) - \chi)^2 = \Delta^2(\overline{\alpha}, Z),$$

where χ – the exact value and $\hat{\chi}(\overline{\alpha}, Z)$ – its estimate. In this case the condition of the pseudogradientness is met if differentiation under the symbol of the mathematical expectation in (4) is possible.

We should also mention that the procedure (3) does not require compulsory finding $\mathrm{J}(\hat{\overline{\alpha}}_{t-1}, Z_t)$ or $\nabla \mathrm{J}(\hat{\overline{\alpha}}_{t-1}, Z_t)$, i.e. $\mathrm{J}(\overline{\alpha})$ can be non-observable. It is necessary to meet only the condition of the pseudogradientness. At the non-observable realization an auxiliary observable quality functional $\mathrm{Q}(\overline{\alpha})$ can be introduced and a noisy value $\mathrm{Q}(\overline{\alpha})$ can be chosen as $\overline{\beta}_t$, whose point of extremum (not necessary the point of minimum) is obtained at the same optimal parameters. Later on, this chosen functional characterizing the estimation quality will be called the goal function. For example, when estimating the mathematical expectation of random value X the following value can be selected as the goal function

$$\mathrm{Q}(\alpha, X) = M\{((X - \alpha)^2)\},$$

then, in the simplest case $\beta_t = x_t - \hat{\alpha}_{t-1}$, where x_t – value X at the t-th iteration. When estimating the correlation coefficient between the centered values X and Y the goal function can be represented as

$$\mathrm{Q}(\alpha, X, Y) = M\{(\alpha X - Y)^2\},$$

then, $\beta_t = (\hat{\alpha}_{t-1} x_t - y_t)$, where x_t and y_t – realizations of X and Y at the t-th iteration.

The problem of IIGDs estimation considered in this chapter is related to the second type of problems, where it is necessary to use the auxiliary quality functional.

Let us note one more important property of the pseudogradient procedures that consists in that, $\overline{\beta}_t$ assumes dependence on estimation values $\hat{\overline{\alpha}}_p$, $p < t$ in the preceding samples and rows of the image that enables to run image processing in the order of some sweep. The last property is very important at practical realization of the algorithms.

Thus, to synthesize fast PGAs of parameters estimation $\overline{\alpha}$, it is necessary to find a relatively easily calculated pseudogradient of the given goal function of the estimation quality. In the next part, several possible ways of computational expense reduction when finding the goal function pseudogradient are considered.

2. Choice of pseudogradient

When synthesizing PGA the important stages are in the choice of a goal function and a rule of finding its pseudogradient. Let us consider some approaches to solve these problems.

Let the studied frames $\mathbf{Z}^{(1)} = \{z_{\bar{j}}^{(1)} : \bar{j} \in \Omega_{\bar{j}}\}$ and $\mathbf{Z}^{(2)} = \{z_{\bar{j}}^{(2)} : \bar{j} \in \Omega_{\bar{j}}\}$ of images specified on a regular samples grid $\Omega_{\bar{j}} = \{\bar{j} = (j_1, ..., j_n) : j_k = \overline{1, N_k}\}$ represent additive mixture of the informational pattern $\mathbf{X} = \{x_{\bar{j}}\}$ and a pattern $\Theta = \{\theta_{\bar{j}}\}$ of an independent noise:

$$\mathbf{Z}^{(1)} = \mathbf{X}^{(1)} + \mathbf{\Theta}^{(1)}, \;\; \mathbf{Z}^{(2)} = \mathbf{X}^{(2)} + \mathbf{\Theta}^{(2)}, \tag{5}$$

where $\overline{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_m)^T$ – vector of unknown geometrical transformation parameters of the image $\mathbf{X}^{(1)} = \mathbf{X}$ into the image $\mathbf{X}^{(2)} = \mathbf{X}(\bar{j}, \overline{\alpha})$, for example, rotation, shift in some direction, scale change etc. In doing so, $x_{\bar{j}}$, $\theta_{\bar{j}}^{(1)}$ and $\theta_{\bar{j}}^{(2)}$ are homogeneous and have Gaussian distribution with zero mean and known covariance functions $R_{x\bar{j}\bar{l}} = M\{x_{\bar{j}} x_{\bar{l}}\}$; $R_{\theta\bar{j}.\bar{l}} = \sigma_\theta^2 \delta_{\bar{j},\bar{l}}$, where $\delta_{\bar{j},\bar{l}} = \begin{cases} 1, & if \;\; \bar{j} = \bar{l}; \\ 0, & if \;\; \bar{j} \neq \bar{l}. \end{cases}$ – Kronecker symbol; $\bar{j}, \bar{l} \in \Omega$.

Under the assumed constraints the goal function for the gradient estimation of the parameters vector $\overline{\alpha}$ can be written using the conditions of the optimal estimation obtained by means of the maximum likelihood method in work (Vasiliev & Tashlinskii, 1998). In particular, it is shown that if the image $\mathbf{Z}^{(1)}$ is noisy, then the maximization of the likelihood function is almost the same as minimization of the quadratic form. Then, for the gradient of the goal function we obtain

$$\nabla J(\overline{\alpha}, Z) = \nabla\left[\left(\dot{\mathbf{Z}}_{\bar{j}}^{(2)} - \dot{\hat{\mathbf{X}}}(\bar{j}, \overline{\alpha})\right)^T \dot{\mathbf{V}}_z^{-1}\left(\dot{\mathbf{Z}}_{\bar{l}}^{(2)} - \dot{\hat{\mathbf{X}}}(\bar{l}, \overline{\alpha})\right)\right], \tag{6}$$

where $\mathbf{V}_z$ – covariance matrix of the conditional distribution $w\left(\{z_{\bar{j}}^{(2)}\} \middle| \mathbf{Z}^{(1)}, \overline{\alpha}\right)$, $\hat{\mathbf{X}}(\bar{j}, \overline{\alpha})$ – prediction found on the basis of observations $\mathbf{Z}^{(1)}$, which is the best estimate in the sense of estimation error variance minimum of a deformed image . The point above the matrices denotes their lexicographic representation. In the same work it is shown that in many cases the product $\dot{\hat{\mathbf{X}}}^T(\overline{\alpha})\dot{\mathbf{V}}_z^{-1}\dot{\hat{\mathbf{X}}}(\overline{\alpha})$ can be considered to be independent of the deformation parameters $\overline{\alpha}$. Then the gradient of the goal function is determined by the relation

$$\nabla J(\overline{\alpha}, Z) = -\nabla\left[\dot{\hat{\mathbf{X}}}(\overline{\alpha})\dot{\mathbf{V}}_z^{-1}\dot{\mathbf{Z}}^{(2)}\right]. \tag{7}$$

We should note that in the last case to find the optimal estimates of the parameters $\overline{\alpha}^*$ the maximization of the goal function is carried out. It requires performing of recurrent algorithm iterations not in the direction of the antigradient, but in the direction of the gradient which corresponds to minus in (7).

It is obvious, the expressions (6) and (7) can not be realized in systems of continuous image processing, because it requires great computational expenses. However, their simplification enables to obtain various realizable pseudogradients of the goal function. Let us consider some possible ways of such simplification. If we assume that the image insignificantly varies

from frame to frame (i.e. $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ are noisy realizations of the images $\mathbf{X}$ and $\mathbf{X}(\bar{j},\bar{\alpha})$), then there is no need to calculate the unwieldy covariance matrix $\mathbf{V}_z$ of the conditional distribution $w\left(\left\{z_{\bar{j}}^{(2)}\right\}\middle|\,\mathbf{Z}^{(1)},\bar{\alpha}\right)$, because in this case $\mathbf{V}_z \approx \sigma_\theta^2 \delta_{\bar{j},\bar{l}}$, where σ_θ^2 – variance of additive noise according to the model of observations (5). However, in this case calculation of the optimal prediction $\hat{x}\left(\bar{j},\bar{\alpha}\right)$ requires matrix operations, which lead to very large computational expenses for large-sized images. We can obtain their reduction by substituting the optimal prediction of values of deformed frame for prediction at limited local region of image. We can achieve even more calculations reduction using various interpolations for the prediction. When performing the interpolations at the current iteration of the algorithm the estimates $\hat{\bar{\alpha}}$ obtained at the preceding iteration are employed (Minkina et al., 2007). Then, to find the pseudogradient at the t-th iteration of the algorithm it is enough to use a local sample $Z_t = \left\{z_{\bar{j},t}^{(2)}, \tilde{z}_{\bar{j},t}^{(1)}\right\}$ of samples, where $z_{\bar{j},t}^{(2)}$ – samples of the deformed image $\mathbf{Z}^{(2)}$ contained in a local sample at the t-th iteration and $\tilde{z}_{\bar{j},t}^{(1)} = \tilde{z}^{(1)}\left(\bar{j}_t, \hat{\bar{\alpha}}_{t-1}\right)$ – brightness values from continuous image $\tilde{\mathbf{Z}}^{(1)}$ obtained from $\mathbf{Z}^{(1)}$ through the chosen interpolation; $\bar{j}_t \in \Omega_{\bar{j},t} \in \Omega_{\bar{j}}$ – sample coordinates $z_{\bar{j},t}^{(2)}$ ($\Omega_{\bar{j},t}$ – plan of a local sample). The number of samples $\left\{z_{\bar{j},t}^{(2)}\right\}$ in Z_t will be called the local sample size and denoted with μ. Under these assumptions the pseudogradients obtained on the basis of relations (6) and (7) will become

$$\bar{\beta}_t = \sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}_{\bar{j},t}^{(1)}}{\partial \bar{\alpha}} \Delta_{\bar{j},t} \Bigg|_{\bar{\alpha}=\hat{\bar{\alpha}}_{t-1}}, \tag{8}$$

$$\bar{\beta}_t = -\sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}_{\bar{j},t}^{(1)}}{\partial \bar{\alpha}} z_{\bar{j}t}^{(2)} \Bigg|_{\bar{\alpha}=\hat{\bar{\alpha}}_{t-1}}, \tag{9}$$

where $\Delta_{\bar{j},t} = \tilde{z}_{\bar{j},t}^{(1)} - z_{\bar{j},t}^{(2)}$.

We should note that the pseudogradient (8) is used for solving the problem of interframe difference mean square minimization. In this case it will be the goal function

$$\mathrm{J}\left(\bar{\alpha},\mathbf{Z}^{(1)},\mathbf{Z}^{(2)}\right) = \frac{1}{M}\sum_{\bar{j}\in\Omega}\left(z_{\bar{j}}^{(2)} - \tilde{z}_{\bar{j}}^{(1)}\right)^2,$$

where M – number of samples in the frame $\mathbf{Z}^{(2)}$.

The pseudogradient (9) corresponds to the problem of interframe correlation sample coefficient maximization:

$$\mathrm{J}\left(\bar{\alpha},\mathbf{Z}^{(1)},\mathbf{Z}^{(2)}\right) = \frac{\sum_{\bar{j}\in\Omega}\left(z_{\bar{j}}^{(2)} - z_m^{(2)}\right)\left(\tilde{z}_{\bar{j}}^{(1)} - \tilde{x}_m^{(1)}\right)}{M\hat{\sigma}_{z1}\hat{\sigma}_{z2}},$$

where: $\hat{\sigma}_{z1}^2 = \frac{1}{M-1}\sum_{\bar{j}\in\Omega}\left(z_{\bar{j}}^{(2)} - z_m^{(2)}\right)^2$ and $\hat{\sigma}_{z2}^2 = \frac{1}{M-1}\sum_{\bar{j}\in\Omega}\left(\tilde{z}_{\bar{j}}^{(1)} - \tilde{x}_m^{(1)}\right)^2$ – variance estimates of the images $\mathbf{Z}^{(2)}$ and $\tilde{\mathbf{Z}}^{(1)}$; $z_{\bar{j}m}^{(2)} = \frac{1}{M}\sum_{\bar{j}\in\Omega} z_{\bar{j}}^{(2)}$; $\hat{x}_m^{(1)} = \frac{1}{M}\sum_{\bar{j}\in\Omega}\tilde{z}_{\bar{j}}^{(1)}$; $\bar{j}\in\Omega_{\bar{j}}$.

Thus, in practical problems of IIGDs estimation the basic goal functions can be the interframe difference mean square and the interframe correlation sample coefficient. We should note that the pseudogradient (9) in contrast to (8) is invariant to the total variability of samples brightness of the image $\mathbf{Z}^{(2)}$. The choice of interframe difference mean square as the goal function is expedient in absence of multiplicative distortions and noncentered interference in the observable image models.

The vector $\bar{\beta}_t = \bar{\phi}(\nabla Q(\bar{\alpha}_{t-1}, Z_t))$ can be chosen as a pseudogradient, where $\bar{\phi}(\cdot)$ – vector function of the same dimensionality as ∇Q. For example, the function $\bar{\phi}(\cdot)$ can be linear transformation with the positively determined matrix. At that, if errors with respect to the true gradient have symmetric distributions with reference to zero, then the condition of the pseudogradientness (2) holds for any odd function $\bar{\phi}(\cdot)$. In particular, very simple and at the same time well converging algorithms of the parameters estimation are obtained when choosing the following sign function as $\bar{\phi}(\cdot)$ (Korn & Korn, 1968)

$$\bar{\beta}_t = \mathrm{sgn}\left(\nabla Q(\hat{\bar{\alpha}}_{t-1}, Z_t)\right), \tag{10}$$

where $\mathrm{sgn}(\nabla Q) = (\mathrm{sgn}(\nabla Q_1), \ldots, \mathrm{sgn}(\nabla Q_m))^T$. When using the peseudogradient (10) and the diagonal gain matrix in PGA (3) the i-th component of the vector $\hat{\bar{\alpha}}_t$ is different from the corresponding component of the vector $\hat{\bar{\alpha}}_{t-1}$ by $\pm\lambda_{i,t}$, where $\pm\lambda_{i,t}$ – corresponding diagonal element of the gain matrix $\mathbf{\Lambda}_t$; $i = \overline{1, m}$. At that, PGA iterations are carried out at finite and a priori known number of directions of the space of the parameters to be estimated. If each component of the error (10) in relation to the true gradient takes positive and negative values with equal probabilities, then the pseudogradientness condition (2) is met. Let us note the algorithms that use the pseudogradients of type (10) have wide application in various problems requiring IIGDs estimation in the conditions of complex noise assemblage.

3. Pseofogradient algorithms for interframe geometrical deformations parameters estimation

3.1 Algorithms at given set of geometrical deformations model parameters

If a parameters set $\bar{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^T$ of possible IIGDs is known, then at chosen goal function the problem amounts to estimation of their values that are constant on the images $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$. For example, if for (3) the interframe difference mean square is chosen as a goal function and its pseudogradient is given by relations (8) and (10), then to estimate $\bar{\alpha}$ we accordingly obtain the following algorithms:

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \left(\sum_{\bar{j} \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}^{(1)}_{\bar{j},t}}{\partial \overline{\alpha}} \Delta_{\bar{j},t} \right)_{\overline{\alpha} = \hat{\overline{\alpha}}_{t-1}} ; \tag{11}$$

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \operatorname{sgn} \left(\sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}^{(1)}_{\bar{j},t}}{\partial \overline{\alpha}} \Delta_{\bar{j},t} \right)_{\overline{\alpha} = \hat{\overline{\alpha}}_{t-1}} . \tag{12}$$

Experimental study show it is expedient to extend the algorithms set (11)–(12) by adding two more ones

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \left(\sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}^{(1)}_{\bar{j},t}}{\partial \overline{\alpha}} \operatorname{sgn} \Delta_{\bar{j},t} \right)_{\overline{\alpha} = \hat{\overline{\alpha}}_{t-1}} ; \tag{13}$$

$$\hat{\overline{\alpha}}_t = \hat{\overline{\alpha}}_{t-1} - \mathbf{\Lambda}_t \left(\sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \Delta_{\bar{j},t} \operatorname{sgn} \frac{\partial \tilde{z}^{(1)}_{\bar{j},t}}{\partial \overline{\alpha}} \right)_{\overline{\alpha} = \hat{\overline{\alpha}}_{t-1}} . \tag{14}$$

In the algorithm (11) all the components of estimation increment vector depend on interframe differences $\Delta_{\bar{j},t}$, $\bar{j}_t \in \Omega_{\bar{j},t} \in \Omega_{\bar{j}}$. It determines higher estimation convergence speed compared to other given algorithms. However, at finite number of iterations the precision of (11) is often lower because we can not always attain little $\Delta_{\bar{j},t}$ and then, the estimates variation steps can be too large. In the algorithm (13) only signs of $\Delta_{\bar{j},t}$ are used. It is preferable to apply it when we want to avoid excessive influence of modulo large values $\Delta_{\bar{j},t}$, for example, in presence of infrequent but intensive impulse interference on image. The algorithm (12) is even more immune to interference, but it may not operate well in the neighborhood of zero values $\Delta_{\bar{j},t}$. If high accuracy of estimation is attained at some iteration, then the next step can be taken in the direction backward from optimal values of the parameters. To eliminate this disadvantage we can employ a sign function with expanded zero:

$$\operatorname{sgn}_\varepsilon(x) = \begin{cases} -1, & x < -\varepsilon, \\ 0, & |x| \le \varepsilon, \\ 1, & x > \varepsilon, \end{cases} \quad \varepsilon > 0 .$$

In the algorithm (14) the increase of convergence speed at large values $\Delta_{\bar{j},t}$ combines with immunity to derivatives estimation errors. As a result this algorithm is more resistant to interference than the algorithm (11).

When choosing the interframe correlation sample coefficient as a goal function the properties of the corresponding algorithms are close to the properties of the algorithms (11)–(14). The advantage in this case is in high immunity to additive noise and close to linear brightness distortions. Among disadvantages are larger computational expenses (determined by large size of the local sample) and also high sensibility to local extremums of the goal function.

It is necessary to note that the convergence speed of the estimates, formed by PGAs, is higher if the sequence of local samples, which is the basis for parameters estimation, is not correlated. To reduce correlation of the observations sequence it is expedient to choose random coordinates of samples of the local sample.

The algorithms (11)–(14) demonstrated high efficiency when estimating interframe deformations of simulated and real images. In particular, for two-dimensional images of size 64x64 pixels formed by means of the wave model (Krasheninnikov, 2003) shifted by several steps of sample grid and turned at any angle, the shift was estimated with error variance of about $2 \cdot 10^{-4}$ steps of the sample grid and rotation – $5 \cdot 10^{-5}$ radians. Let us give the results of analytical calculation of the probability $P(\Delta)$ of parallel shift estimate $\hat{h}_1$ error spillover of the given interval $\Delta = [-a\ a]$. The calculation was carried out on the basis of the accuracy analysis method of PGAs estimates at finite number of iterations (Tashlinskii & Tikhonov, 2001) for the PGA (12) and the following parameters: the images with a Gaussian brightness distribution and the autocorrelation function with correlation radius equal to 5; the signal variance-to-noise variance ratio $g = 100$; local sample size $\mu = 10$; initial error of the shift is $\bar{h}_o = (5, 4)^T$; $a = 1.0$, 0.3 и 0.1 (here, by correlation radius we imply the distance in steps of the sample grid when the autocorrelation function of the image is equal to 0.5). The value of estimate shift increment in one case was chosen to be constant $\lambda_{1,t} = \lambda_{2,t} = \lambda = const = 0.1$ and in the other – falling off according to the rule $\lambda_{1,t} = \lambda_{2,t} = \mathrm{var} = \dfrac{1}{(10 + 0.01t)}$. The plots of the probability $P(\Delta)$ are shown in Fig. 1. It follows from the analysis of the plots that at constant λ the balance between tendency of the estimate to the true value and error, caused by λ, comes at a certain iteration. The further increase of iterations number does not lead to estimates improvement. It enables to find the minimum number of iterations that is necessary to achieve the highest possible accuracy of parameter estimation.

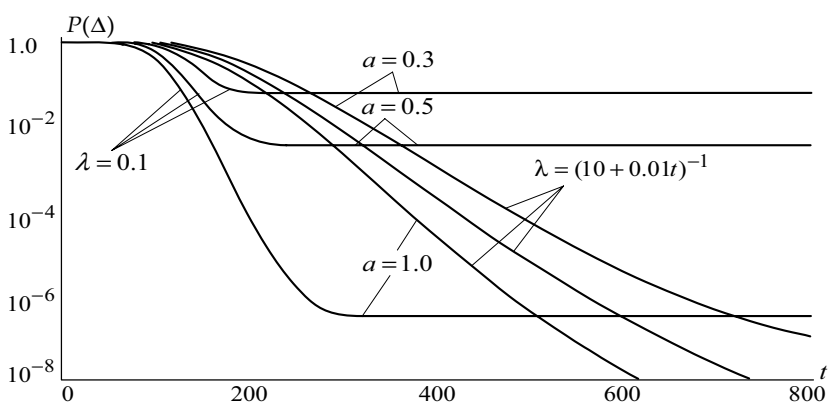


Fig. 1. Probability of estimate error spillover of the confidence interval

For the above-mentioned characteristics of images and PGA in Fig. 2 the shift h_1 estimate error probability distribution change at both constant (Fig. 2,a) and varying (Fig. 2,b) value

of the shift estimate increment is shown. For clearness the distributions are presented only for 12 iterations from the range of 10 to 100. The interframe difference mean square at local sample size $\mu = 4$ was used as a goal function. The analysis of the plots shows that at constant shift increment step the process of probability distributions forming stabilizes after about 500th iteration. At varying shift increment step the process of probability distribution forming does not have an equilibrium state and the estimate variance theoretically permanently decreases.

Let us note that at known set of IIGD parameters the algorithms (11)–(14) have shown a good performance at automated search of local fragments on images.

3.2 Algorithms at unknown set of geometrical deformations model parameters

If the form of IIGD is not given then we can specify a certain sample grid deformations model

$$\overline{\alpha}(\bar{j}) = \left(\bar{j} + \overline{h}_{\bar{j}}\right) = (j_1 + h_1, j_2 + h_2, \ldots\ j_n + h_n)\,, \tag{15}$$

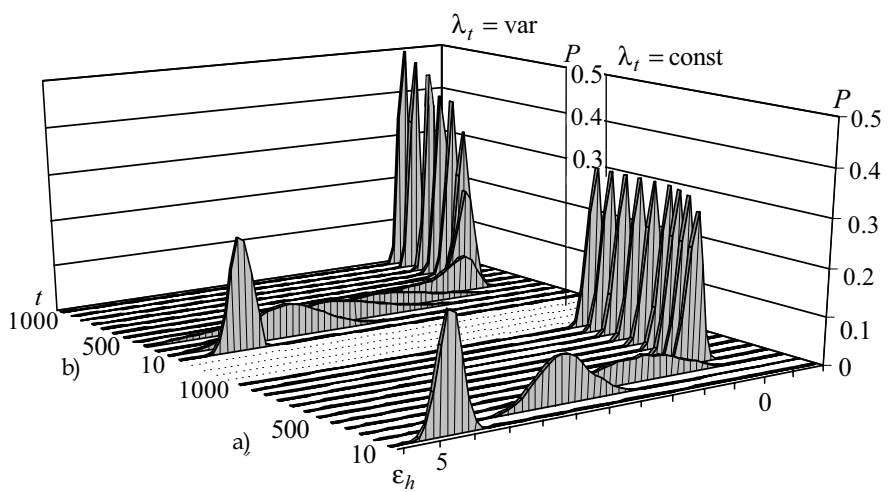


Fig. 2. Probability distributions of the shift estimate error at constant and varying estimate increment steps

considering its parameters to be varying, where $j_k \in \bar{j} \in \Omega_{\bar{j}}$, $j_k = \overline{1, N_k}$, $k = \overline{1, n}$; $\Omega_{\bar{j}}$ – n-dimensional rectangular grid. Then, the algorithm (3) can be written in the form

$$\hat{\overline{h}}_t = \hat{\overline{h}}_{t-1} - \mathbf{\Lambda}_t \overline{\beta}_t\left(Z_t, \bar{j}_t, \hat{\overline{h}}_{t-1}\right) \tag{16}$$

In this case to ensure variability of the estimates the components of the gain matrix $\mathbf{\Lambda}_t$ in (16) have to be bounded below. Then, assuming $\mathbf{\Lambda}_t = \mathbf{\Lambda}$ and choosing the interframe difference mean square as a goal function, we come to the algorithm

$$\hat{\bar{\alpha}}_t = \hat{\bar{\alpha}}_{t-1} - \mathbf{\Lambda}_t \left(\sum_{\bar{j}_t \in \Omega_{\bar{j},t}} \frac{\partial \tilde{z}^{(1)}\left(\bar{j}_t, \bar{h}\right)}{\partial \bar{h}} \left(\tilde{z}^{(1)}\left(\bar{j}_t, \bar{h}\right) - z^{(2)}_{\bar{j},t} \right) \right)_{\bar{h} = \hat{\bar{h}}_{t-1}} . \quad (17)$$

In work (Tashlinskii, 2000) it is shown that if point-to-point correlation is within the limits $0.8 \div 0.99$ and signal variance-to-noise variance ratio is more than 50, then for many problems of IIGDs estimation it is enough to choose $\mu = 1$ in (17). In spite of the simplicity the algorithm (17) is rather effective at high correlation of the shifts $\bar{h}_{\bar{j}}$ in the direction of image scanning and relatively minor interframe brightness distortions. However, only one-dimensional filtering of deformation parameters that does not take into account interrow and interframe correlation is carried out in it. The simplest way to take into account this correlation can be in refinement of the estimates, obtained at one pass through the images. For that we can perform repeated passes at lesser values λ in the backward and other directions (along columns, diagonals) during which the obtained estimates are corrected.

The algorithms of the type (17) have shown a good performance at automatic combination of image fragments that have reciprocal spatial and amplitude deformations (the problem of image «pasting»). This problem often occurs when forming a unified image from a sequence of frames, obtained from a mobile object, that have small common regions on the adjacent frames. When solving the mentioned problem it is required as a rule to ensure continuity of spatial and brightness characteristics on the resulting image. The considered algorithms enable to do it. To illustrate it in Fig.3 an example of «pasting» of images is presented, where a) and b) – are the images of size 100×160 elements to be connected, having parallel shift $\bar{h} = (-0.5, -0.2)^T$, rotation angle $\varphi = 0.5^o$ and scale coefficient $k = 1.2$; c) – the result of «pasting» before spatial correction using the obtained estimates; d) – the result of «pasting» after correction. The estimates accounted for $\hat{\bar{h}} = (-0.676, -0.13)$, $\hat{\varphi} = 0.509^o$, $\hat{k} = 1.18$.

a) b) c) d)

Fig. 3. An example of automated «pasting» of images

Basically, the model (15) enables to define any IIGDs. However, if at the chosen order of the images pass the shifts $\bar{h}_{\bar{j}}$ change rapidly then their estimation by means of PGA (17) is difficult. It is due to the fact that when variation speed of the shifts increases in the direction of the image pass it is necessary to increase steps $\left(\mathbf{\Lambda}_t \bar{\beta}_t\right)$ of PGA (17). The last, in its turn, leads to estimation error increase. In this situation we can not improve the estimates even by repeated passes. The mentioned contradiction can be solved due to the usage at each following pass information about estimates, obtained at the preceding passes. Let us

consider the algorithm that forms the deformation matrix $\mathbf{H}_l$ of size $N_1 \times N_2 \times ... \times N_n$ (where n – image dimensionality) as an example of such an approach. This matrix contains shifts estimates $\hat{\bar{h}}_{\bar{j}}^{(l)}$ of all image pixels, corresponding to sample grid nodes of the frame $\mathbf{Z}^{(1)}$ after the l-th pass. The method of estimates forming can be various, for example, it can be determined by available conceptions regarding physical nature of geometrical deformations. Assuming, that all elements of the matrix $\mathbf{H}_{l-1}$ at the $(l-1)$-th pass have been determined we can write

$$\mathbf{H}_l = \left\| \hat{\bar{h}}_{\bar{j}}^{(l)} \right\| = f\left(\hat{\bar{h}}_{\bar{j}}^{(l-1)}, \left\{ \hat{\bar{\alpha}}_t^{(l)} \right\} \right),$$

$$\hat{\bar{\alpha}}_t^{(l)} = \hat{\bar{\alpha}}_{t-1}^{(l)} - \mathbf{\Lambda}_{lt} \bar{\beta}\left(z_{\bar{j},t}^{(2)}, \tilde{z}^{(1)}\left(\bar{j}_t + h_{\bar{j},t}^{(l-1)} \right), \hat{\bar{\alpha}}_{t-1}^{(l)} \right),$$

where $\hat{\bar{\alpha}}_t^{(l)} = \left(\hat{\alpha}_{1,t}^{(l)}, \hat{\alpha}_{2,t}^{(l)}, ..., \hat{\alpha}_{m,t}^{(l)} \right)^T$; $\bar{j}_t \in \Omega_t^{(l)} \in \Omega$; $\mathbf{\Lambda}_{l,t} = \left\| \begin{matrix} \lambda_{1,t}^{(l)} & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & \lambda_{n,t}^{(l)} \end{matrix} \right\|$; $z_{\bar{j},t}^{(2)}$, $\tilde{z}^{(1)}\left(\bar{j}_t + h_{\bar{j}t}^{(l-1)} \right) \in Z_t^{(l)}$; $Z_t^{(l)}$ – local sample of the goal function for the pseudogradient estimation at the t-th iteration of the l-th pass of the algorithm; $l = \overline{1,L}$ - pass number; L – given number of passes. Choosing various goal functions and pseudogradients we can obtain different algorithms. For example, if we choose the pseudogradient (10) for the interframe difference mean square and assume $\lambda_{it}^{(l)} = \lambda_l$; $\hat{\bar{h}}_{\bar{j}}^{(l)} = \hat{\bar{h}}_{\bar{j}}^{(l-1)} + \hat{\bar{\alpha}}_{\bar{j}}^{(l)}$, where $\hat{\bar{\alpha}}_{\bar{j}}^{(l)}$ – current estimate $\hat{\bar{\alpha}}^{(l)}$ at the point $\bar{j}$ then we obtain

$$\mathbf{H}_l = \left\| \hat{\bar{h}}_{\bar{j}}^{(l)} \right\|, \ \hat{\bar{h}}_{\bar{j}}^{(l)} = \hat{\bar{h}}_{\bar{j}}^{(l-1)} + \hat{\bar{\alpha}}_{\bar{j}}^{(l)}, \ l = \overline{1,L},$$

$$\hat{\bar{\alpha}}_t^{(l)} = \hat{\bar{\alpha}}_{t-1}^{(l)} - \lambda_l \operatorname{sgn}\left(\sum_{\bar{j}_t \in \Omega_{\bar{j},t}^{(l)}} \frac{\tilde{z}^{(1)}\left(\bar{j}_t + \hat{\bar{h}}_{\bar{j},t}^{(l-1)} + \hat{\bar{\alpha}}_{t-1}^{(l)} \right)}{\partial \bar{\alpha}} \left(\tilde{z}^{(1)}\left(\bar{j}_t + \hat{\bar{h}}_{\bar{j},t}^{(l-1)} + \hat{\bar{\alpha}}_{t-1}^{(l)} \right) - z_{\bar{j},t}^{(2)} \right) \Bigg|_{\bar{\alpha} = \hat{\bar{\alpha}}_{t-1}^{(l)}} \right). \quad (18)$$

If IIGDs with known parameters set (for example, the common for all image shift, rotation etc.) are present along with deformations of unknown type, then values of these parameters can be estimated, specified and taken into account when forming elements of matrix $\mathbf{H}_l$ at each algorithm pass.

Another advantage of the algorithms of the type (18) is that they enable to estimate IIGDs that do not satisfy the continuity requirement.

An example of such estimation is shown in Fig. 4, where a) and b) – images of size 256×256 elements having reciprocal shifts ($h_1 = 1.5$, $h_2 = 3.5$), besides in the lower image the continuity of geometrical deformations is violated (5 rows are missing) and the fragment is

absent; c) – result of parameter h_1 estimation at $L = 40$ and $\mu = 1$. The sudden change corresponding to the break of the parameter h_1 is well visible. Besides in the region, corresponding to the absent fragment, the estimates have significantly differing statistical properties and due to which these regions can be easily identified.

One of the disadvantages of PGA at IIGD parameters estimation is in a relatively minor definition domain of parameters, where effective convergence of estimates is ensured (not large operating range). The size of this region is determined by sample correlation that can appear in the local sample Z_t. The situation is also complicated by the fact that in real images samples of reference and deformed images taken rather far from each other are almost non-correlated. At operating with real images another serious disadvantage of PGAs is in the possibility of the estimates to converge to points of false extremums of the goal function in the parameter space.

3.3 Algorithms with adaptive forming of local sample size

In view of random character of images and noise, the estimate of the goal function is not unimodal and besides the global extremum it also contains false local extremums. The local extremums appear because of correlatedness of separate extensive objects on the image and are exposed if a portion of samples of the local sample appears into these regions, i.e. they are caused by limited size of the local sample. As local sample increases or changes the probability of this effect appearance sharply decreases. As a result it is reasonable to verify on the goal function local extremums attributes at each iteration of estimation and if any, to increase sample size or change it. Here, the sample size μ becomes an adaptive value.

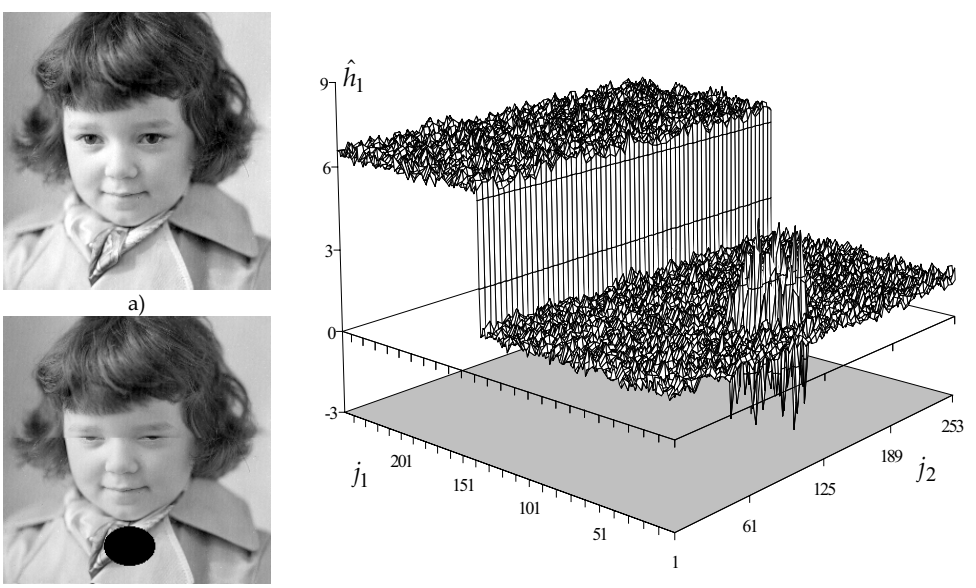


Fig. 4. An example of geometrical deformations estimation that does not satisfy the continuity requirement

Let us consider only one example of construction of IIGD parameter estimation PGA, where μ is adjusted automatically during the procedure performing at each iteration. A current iteration of parameter estimation is carried out when a certain condition is met. If at minimal μ for the current iteration the condition is not met, then μ increases step-by-step until the condition is fulfilled. So for the local sample, formed at the given iteration, its size is to be minimum to meet the condition of the iteration realization. For definiteness, we assume that the PGA (3) with the pseudogradient (10) and the diagonal gain matrix $\mathbf{\Lambda}_t$ is used. Then, the estimates for the i -th parameter are formed according to the following rule:

$$\hat{\alpha}_{i,t+1} = \hat{\alpha}_{i,t} \mp \lambda_{t+1} \operatorname{sgn}\left(\beta_{i,t+1}\left(\hat{\bar{\alpha}}_t, Z_t\right)\right), \quad t = \overline{1,T}, \quad i = \overline{1,m},$$

where the signs «–» and «+» correspond to the problems of minimization and maximization of the goal function.

Given a certain initial sample size $\mu_{t\min}$, whose minimum value at interframe correlation sample coefficient maximization must be not less than 2 and at the interframe difference mean square minimization – is equal to 1. To find numerical values in the conditions of the iteration realization we use the goal function estimates obtained at the corresponding μ. Let us denote the goal function estimate with $q_t(\mu_k)$ which is calculated at the t -th iteration on the samples $z_{\bar{j},t}^{(2)}$ and $\tilde{z}_{\bar{j},t}^{(1)}$ of the local sample of size μ_k and as $q_t^{\pm}(\mu_k)$ – goal function estimate at the t -th iteration at the same μ_k calculated on samples $z_{\bar{j},t}^{(2)}$ and $\tilde{z}_{\pm}^{(1)}(\bar{j}_t, \hat{\alpha}_{1,t-1}, \ldots \hat{\alpha}_{i,t-1} \pm \Delta_{\alpha,i}, \ldots \hat{\alpha}_{m,t-1})$, i.e. when a certain increment $\Delta_{\alpha i} > 0$ is specified $\bar{j}_t \in \mathbf{\Omega}_t \in \Omega$ for the parameter α_i estimate.

Let us fulfill the following condition of the iteration realization: the iteration of finding the current estimate $\hat{\alpha}_{i,t+1}$ is not performed and $\mu_{t\min}$ increases by 1 (a new pair of samples $z_{\bar{j},t}^{(2)}$ and $\tilde{z}_{\bar{j},t}^{(1)}$ is added to the local sample) in two cases:

– if at the current t -th iteration the estimate $q_t(\mu_{t\min})$ for the local sample size $\mu_{t\min}$ is «better» than both the values $q_t^{+}(\mu_{t\min})$ and $q_t^{-}(\mu_{t\min})$;

– if at the current t -th iteration the estimate $q_t(\mu_{t\min})$ for the local sample size $\mu_{t\min}$ is «worse» than the values $q_t^{+}(\mu_{t\min})$ and $q_t^{-}(\mu_{t\min})$ but at that $q_t^{+}(\mu_{t\min}) = q_t^{-}(\mu_{t\min})$.

After that the sample size increases by one ($\mu_{t\min}$ +1) and the above-mentioned conditions are verified again. If one of them is fulfilled, then μ increases by one again and so on right up to a certain value $\mu_{\max}$. If $\mu_{\max}$ is attained, the following iteration of the parameter α_i estimation is performed. If at the current μ the conditions are not met, then the next $(t+1)$ -th iteration of the estimate $\hat{\alpha}_{i,t+1}$ forming for the parameter α_i is carried out.

In particular, when maximizing the goal function we can write the procedure of parameter α_i estimation in the following form

$$\hat{\alpha}_{i,t+1} = \hat{\alpha}_{i,t} - \lambda_{t+1}\beta_{i,t+1}(\mu_t);$$
$$\mu_t = \begin{cases} \mu_k + 1, \ if \ \left(q_t(\mu_k) < q_t^+(\mu_k) \wedge q_t^-(\mu_k)\right) \vee \left(q_t(\mu_k) > q_t^+(\mu_k) = q_t^-(\mu_k)\right), \\ \mu_{\max}, \ if \ \mu_k = \mu_{\max}, \\ \mu_k, \quad in\ the\ other\ case; \end{cases} \tag{19}$$
$$\beta_{i,t+1} = \begin{cases} 1, \ if \ \ q_t^+(\mu_t) < q_t^-(\mu_t) \wedge q_t(\mu_t) > \min\left(q_t^+(\mu_t), q_t^-(\mu_t)\right); \\ 0, \ if \ \ q_t^+(\mu_t) = q_t^-(\mu_t); \\ -1, \ if \ \ q_t^+(\mu_t) > q_t^-(\mu_t) \wedge q_t(\mu_t) > \min\left(q_t^+(\mu_t), q_t^-(\mu_t)\right). \end{cases}$$

Let us note that as t increases the value $\mu_{t\min}$ varies according to a certain prescribed rule that is defined by the problem to be solved, in particular, in the simplest case $\mu_{t\min} = const$. In Fig. 5 experimental results obtained for the algorithm (19) realization are presented. In the experiment a real image of optical range with correlation radius equal to 5 steps on the sample grid was used. A parameter to be estimated was the parallel shift $\overline{h} = (10, 0.5)^T$. The shifted image was additionally noised by an independent centered Gaussian noise. The dependencies of μ_t as a function of the number of iterations, averaged on 50 realizations, are shown in Fig. 5,a. Here, the dependence 1 corresponds to signal variance-to-noise variance ratio g =100 and the dependence 2 – g =50. It is seen that for great errors of the estimate the sample size is small (for g =100 at t =10 it is equal to about 2 and at t =500 – to about 2.3) and it increases monotonously on average as the number of iterations increases (attaining about 6 at g =100 and t =2000). In Fig. 5,b plots of the estimation error ε_h versus the number of iterations are presented, where curve 1 corresponds to the results, obtained for the adaptive forming of μ_t, and curve 2 – at constant $\mu = \mu_m$, where $\mu_m = \sum_{k=1}^{2000} \mu_k$ – average sample size for t varying from 1 to 2000. The results are averaged on 250 realizations. It is obvious at small number of iterations there is a loss in estimation accuracy (at the 100th iteration it is about 5 per cent). It can be explained by high speed of the algorithm convergence with constant μ at the initial stage of estimation (due to a greater average of μ). However, at equal computational expenses (to the 2000th iteration) a gain in accuracy of about 2.4 times as large is guaranteed.

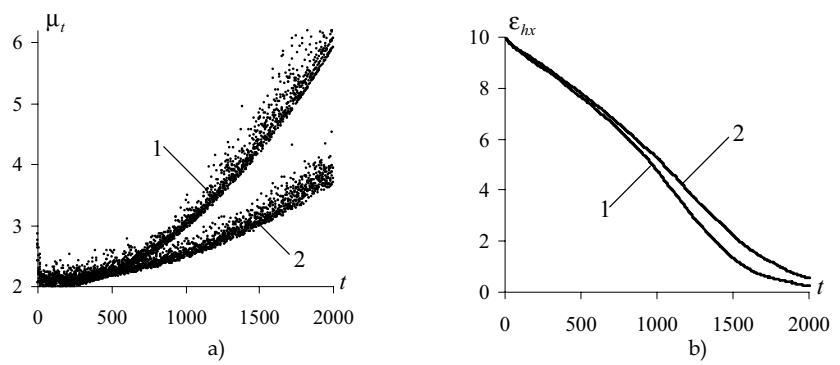


Fig. 5. An example of dependence of local sample size and estimate error versus the number of iterations

Thus, due to the fact that the proposed PGA with adaptive size of local sample facilitates the estimates vector recovery from local extremums of the goal function, it enables to significantly increase parameter estimate convergence speed in comparison with PGA with constant sample size at equal computational expenses. One more example of construction of PGA with adjustable local sample size is presented in work (Tashlinskii, 2003).
Let us note that if the problem of IIGD parameter estimation is a part of the problem of identification with a decision rule based on goal function values, then in order to achieve the required confidence probability of identification it may require large sample size μ_I that is not justified in the process of the estimates $\hat{\bar{\alpha}}$ convergence. In this case it is expedient to use adaptive adjustment of sample size and we can choose $\mu_{max} = \mu_I$ as its maximum. Then the attainment of μ_{max} will simultaneously mean the identification problem solution. An example of such a problem can be search of a fragment location on the reference image where a criterion of correspondence is in excess of a certain correlation sample coefficient value between the fragment and the reference image.

4. Structural optimization of pseudogradient algorithms

In practical problems of IIGDs estimation by means of pseudogradient procedures the required accuracy of parameter $\bar{\alpha}$ estimates is not obtained in the whole domain $\Omega_{\bar{\alpha}}$ of possible values $\bar{\alpha}$, but only in a certain subdomain bounded by an operating range of the procedures. This leads to the necessity of decomposing $\Omega_{\bar{\alpha}}$ into N subdomains $\Omega_{\bar{\alpha}}^{(i)}\left(\hat{\bar{\alpha}}_{0,i}\right)$, $i = \overline{1,N}$ corresponding to the operating range of the employed procedures where $\hat{\bar{\alpha}}_{0,i}$ – an initial approximation of the parameters for the procedure operating in the i-th subdomain. Let the procedure, operating in the subdomain which contains the sought vector $\bar{\alpha}_v$ of parameters, be called a V-procedure (from veritas - true) and the corresponding subdomain – V-subdomain. Subdomains that do not include $\bar{\alpha}_v$ are called P-subdomains (from pseudo – false) and the corresponding procedures – P-procedures. As the result of all these procedures operation N vectors $\hat{\bar{\alpha}}_i$ of IIGD parameter estimates are formed and the problem of determination of a V-subdomain among these estimates with required accuracy where the goal function attains its extremum arises.

4.1 The principle of pseudogradient procedure set control

In the problems of IIGDs estimation the number of subdomains can run up to dozens of thousands. Thus, bringing all the procedures operating in subdomains to the number of iterations that ensures the necessary accuracy of estimation requires great computational expenses. At such an approach the choice of the V-subdomain requires additional calculations. To reduce computational expenses the following principle of structural optimization can be used (pseudogradient procedures set control). At each step of the algorithm the priority of the current iteration realization is given to the procedure, having the least value of a certain penalty function ψ characterizing the level of the priority (Tashlinskii, 2006). Here, by «algorithm step» we imply a set of operations that includes performing of the current iteration by the procedure with the least penalty function, finding

a new value of the penalty function and obtaining a procedure with the least penalty function.

A characteristic property of such an approach is in the necessity to compare the «penalties» of the procedures which have performed different number of iterations. Studies have shown that when minimizing the goal function the following penalty functions satisfies such requirements

$$\psi_t^{(i)} = \sum_{k=1}^{t}\left(q_k^{(i)} - q_{\inf}\right),\ i = \overline{1,N},$$

where $q_k^{(i)}$ – goal function estimate at the k-th iteration; $q_{\inf} \le \inf\left\{q_k^{(i)}\right\}$ – value which is less than the lower bound of the possible estimates set of the goal function. If the goal function is to be maximized then

$$\psi_t^{(i)} = \sum_{k=1}^{t}\left(q_{\sup} - q_k^{(i)}\right),\ i = \overline{1,N},$$

where $q_{\sup} \ge \sup\left\{q_k^{(i)}\right\}$.

In the process of parameters estimates convergence to the optimal values the probabilistic properties of the goal function estimates are changing, which leads to the change of the probabilistic properties of the penalty function ψ. So when studying properties of ψ it is necessary to know its probability distribution density $w_t(\psi)$ at each iteration of estimation. At that, $w_t(\psi)$ depends on the local sample Z_t i.e. a rate of the correspondence (similarity) of the sets $\left\{\tilde{z}_{\hat{j},t}^{(1)}\right\}$ and $\left\{z_{\hat{j},t}^{(2)}\right\}$ involved in the local sample. It is expedient to use correlation sample coefficient ρ as a value characterizing this correspondence. For isotropic images ρ is a one-dimensional characteristic for any number of parameters to be estimated which simplifies calculations. Then for probability distributions of the penalty function increment $\Delta\psi$ at the t-th iteration we can write

$$w_t(\Delta\psi) = \int_{-1}^{1} w_t(\Delta\psi|\rho)w(\rho)\,d\rho, \tag{20}$$

where $w_t(\Delta\psi|\rho)$ – conditional density of increment; $w(\rho)$ – probability distribution density of the correlation coefficient. Let us note that for V-procedures $w_t(\Delta\psi|\rho)$ depends on the iteration number because ρ increases as the estimates vector $\hat{\bar{\alpha}}$ converges to the optimal values.

Without loss of generality, we can assume that ψ takes only positive values. Then, for calculation of the distribution density of ψ at the t-th iteration we can obtain the recurrent expression

$$w_t(\psi) = \int_0^{\infty}\int_{-1}^{1} w_{t-1}(\psi - \Delta\psi_t)w_t(\Delta\psi\,|\,\rho)w(\rho)\,d\Delta\psi\,d\rho. \tag{21}$$

For the P-procedures $w_t(\Delta\psi|\rho)$ does not depend on the iteration number. Then,

$$w_t(\psi) = \int_0^{\infty} w_{t-1}(\psi - \Delta\psi_t)w(\Delta\psi)\,d\Delta\psi, \tag{22}$$

where $w(\Delta\psi) = w_t(\Delta\psi|\rho = 0)$.

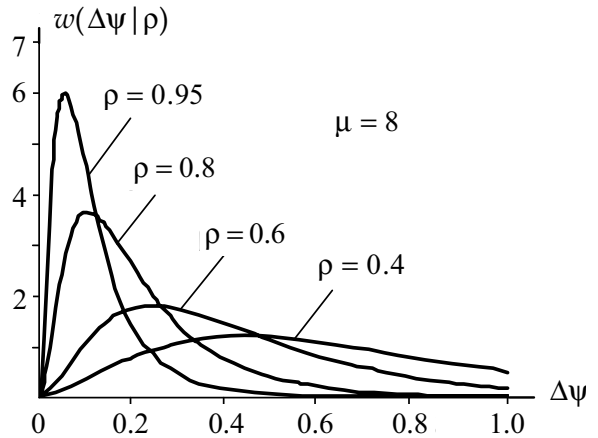
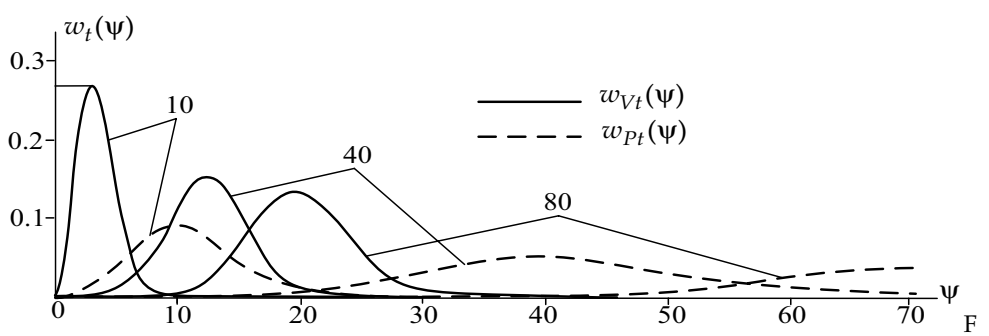


Fig. 6. Conditional distribution density of the penalty function increment

The expressions (20)–(22) enable to easily find the penalty function distribution density for the goal functions obtained in the second part. As an example, in Fig. 6 curves $w(\Delta\psi|\rho)$ of the increment $\Delta\psi$ of the interframe difference mean square at ρ =0.4, 0.6, 0.8, 0.95 are presented. Other parameters of calculation were the following: the images $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are Gaussian with correlation radius equal to 5 steps of the sample grid; the signal/noise ratio $g = 100$; the local sample size μ =4. In Fig. 7 the probability distribution densities of the interframe correlation sample coefficient for V-procedure ($w_{Vt}(\psi)$) and P-procedure ($w_{Pt}(\psi)$) at the number of iterations 10, 40 and 80 and μ = 7 are presented. From the plot it is seen that the area of intersection between $w_{Vt}(\psi)$ and $w_{Pt}(\psi)$ decreases sharply as the number of iterations increases, which facilitates reliable separation of the procedures of P- and V-type.



ig. 7. Penalty function distribution density at the interframe correlation sample coefficient

4.2 Testing of the hypothesis about goal function extremum absence in the parameter definition domain

If the existence of V-subdomains in the IIGD definitional domain to be studied is not known a priori, then the problem of testing of the hypothesis that there is no V-subdomain among N studied subdomains appears. Let us consider the possibility of construction of a simple criterion for testing of such a hypothesis. In doing so, we use the circumstance that when using the principle of the structural optimization, the basic feature of the V-subdomain is in the number of iterations performed by the procedure that operates in it. If among the procedures to be studied there is a procedure corresponding to the V-subdomain, then the number of iterations performed by it is, as a rule, more than the number of iterations performed by the procedures, which operate in P-subdomains. So the V-procedure attains the given number of iterations on average faster than P-procedure. In absence of a V-subdomain in the domain to be studied all the procedures carry out, on average, equal number of iterations and the leading procedure attains the threshold number of iterations for a larger number of algorithm steps. Thus, the total number M of iterations performed by all procedures, the number of steps of the algorithm before the leading procedure attains a certain threshold value T_M can be chosen as a numerical value of the criterion of acceptance of the hypothesis about absence of the V-subdomains. In doing so, the statistical criterion of the hypothesis testing becomes very simple: if during $M = M_c$ steps of the algorithm none of procedures has reached the T_M -th iteration then there is no subdomain of V-type among the analyzed subdomains. The choice of values T_M and M_c enables to obtain the necessary error probabilities of the first and the second kind.

The value T_M in a number of cases can be defined by the problem of the investigation following the hypothesis testing, for example, in the problem of fragment search on a large image – by the accuracy of its location parameters determining. In this case it is expedient to find the number of the algorithm steps M_{c1} or M_{c2} ensuring the required error probability of the first or the second kind basing on the preassigned value T_M . Let us note that a certain value of the error probability of the second kind corresponds to each value M_{c1} assuring a given error probability of the first kind and vice versa. We can specify only one probability whereas the second will be determined through the given probability value and the algorithm parameters. Simultaneous meeting of both the conditions (Here,, one of them will be limited) can be obtained by a choice of a value T_M . If T_M and the algorithms parameters are given, then we can find the required number of steps M_{c1} for the error probability of the first kind and M_{c2} – for the error probability of the second kind and use the biggest of them in the algorithm as the threshold value. In the process of the algorithm realization, depending on the number of iterations t_L performed by the leading procedure and the total number M of algorithm steps, several various situations are possible that characterize veracity of the criterion of testing of the hypothesis about absence of a V-subdomain in the domain to be studied. Possible variants are presented in Tab.

t_L	M	Conclusion
$t_L < T_M$	$M < M_{c1} \wedge M < M_{c2}$	The error probabilities of the first and the second kind exceed the given values $P^{(1)}$ and $P^{(2)}$.
$t_L < T_M$	$M = \min\{M_{c1}, M_{c2}\}$	If $M_{c1} < M_{c2}$ then the error probability of the first does not exceed the value $P^{(1)}$; if $M_{c1} > M_{c2}$, then the error probability of the second does not exceed the value $P^{(2)}$.
$t_L < T_M$	$M = \max\{M_{c1}, M_{c2}\}$	The error probabilities of the first and the second kind do not exceed the given values $P^{(1)}$ and $P^{(2)}$.
$t_L \geq T_M$	$M < M_{c1} \wedge M < M_{c2}$	The error probabilities of the first kind exceeds the given values $P^{(1)}$ and $P^{(2)}$.

Table. Veracity of the criterion of testing of the hypothesis about absence of the V-subdomains in the parameters domain to be studied

Let us consider the veracity of the proposed criterion. First let us find the number of algorithm steps M_{c2} guaranteeing the hypothesis testing with the specified error probability of the second kind $P^{(2)}$. This error may appear when two conditions are simultaneously satisfied:

– the V-procedure has not reached the T_M-th iteration (we will consider the probability of this condition meeting to be $P_V^{(2)}$);

– none of P-procedures has reached the T_M-th iteration iteration (we will consider the probability of this condition meeting to be $P_P^{(2)}$).

Then assuming $P_V^{(2)}$ and $P_P^{(2)}$ to be independent we obtain

$$P^{(2)} = P_V^{(2)} P_P^{(2)} .$$

To find the probabilities $P_V^{(2)}$ and $P_P^{(2)}$ it is necessary to know the discrete distributions of the number of iterations for V- and P-procedures at the given number M of algorithm steps. Let us denote the discrete distribution of the number of iterations for the V-procedure as P_{Vt}, $t = \overline{1, T_M}$, where P_{Vt} – probability of the event that the V-procedure has performed t iterations at M algorithm steps. Then, the probabilities of the event that the V-procedure will not reach the T_M-th iteration are

$$P_V^{(2)} = \sum_{t=1}^{T_M - 1} P_{Vt} .$$

Accordingly, let us denote the discrete distribution of the number of iterations for the P-procedure as P_{Pt}, $t=\overline{1,T_M}$, where P_{Pt} – probability of the event that the P-procedure has performed t iterations at M algorithm steps. Here, the probability of the event that none of the P-procedures reaches the T_M -th iteration is

$$P_P^{(2)} = \left(\sum_{t=1}^{T_M-1} P_{Pt} \right)^{N-1} .$$

Let us note that P_{Vt} and P_{Pt} depend on the total number M of algorithm steps accordingly various M correspond to different distribution density $w_{T_M}(\psi)$ of the penalty function

If the number of algorithm steps M and the probability $P^{(2)}$ are given, then we can find the corresponding number of iterations T_M from the following considerations. On average, the V-procedure fulfills $\sum_{t=1}^{T_M-1} tP_{Vt}$ iterations at M algorithm steps and each of P-procedures - $\sum_{t=1}^{T_M-1} tP_{Pt}$ iterations. Then, we can obtain T_M from the condition

$$\sum_{t=1}^{T_M-1} tP_{Pt} + (N-1) \sum_{t=1}^{T_M-1} tP_{Pt} = M . \tag{23}$$

The condition (23) can be simplified if we take into account the fact that the V-procedure in comparison with the V-procedure guarantees significantly high convergence speed and will fulfill the number of iterations close to T_M at M steps. Then,

$$T_M + (N-1) \sum_{t=1}^{T_M-1} tP_{Pt} \approx M .$$

Having determined T_M it is easy to find probabilities P_{Vt} and P_{Pt}:

$$P_{Pt} = \int_0^{\infty} w_{T_M}(\psi)(1 - F_{Pt}(\psi))d\psi - \sum_{i=1}^{t-1} P_{Pi} ,$$

$$P_{Vt} = \int_0^{\infty} w_{TM}(\psi)(1 - F_{Vt}(\psi))d\psi - \sum_{i=1}^{t-1} P_{Vi} ,$$

where $F_{Pt}(\psi) = \int_0^{\psi} w_{Pt}(x)\mathrm{d}x$ и $F_{Vt}(\psi) = \int_0^{\psi} w_{Vt}(x)\mathrm{d}x$ – distribution functions of ψ at the t -th iteration for P- and V-procedures; $w_{T_M}(\psi)$ – penalty function distribution density at the T_M -th iteration defined by the procedures of both V-type and P-type. At one V-procedure

$$w_{T_M}(\psi) = w_{VT_M}(\psi)(1 - F_{PT_M}(\psi))^{N-1} + w_{PT_M}^{(N-1)}(\psi)(1 - F_{VT_M}(\psi)), \tag{24}$$

where $w_{PT_M}^{(N-1)}(\psi)$ – distribution density of the minimum of $(N-1)$ penalty functions values of P-procedures at the T_M -th iteration.

Thus,

$$P^{(2)} = \sum_{t=1}^{T_M-1} P_{Vt}\left(\sum_{i=1}^{T_M-1} P_{Pi}\right)^{N-1} = \left(\int_0^\infty w_{T_M}(\psi)\left(1-F_{V(T_M-1)}(\psi)\right)d\psi\right)\left(\int_0^\infty w_{T_M}(\psi)\left(1-F_{P(T_M-1)}(\psi)\right)d\psi\right)^{N-1}. \quad (25)$$

Similarly we can find the error probability of the second type $P^{(1)}$ (the probability of the event that at least one of N P-procedures performs not less than T_M iterations at M algorithm steps):

$$P^{(1)} = 1-\left(\sum_{i=1}^{T_M-1} P_{Pi}\right)^N.$$

In this case V-procedures are absent so $w_{T_M}(\psi) = w^{(N)}_{PT_M}(\psi)$ – probability distribution of minimum of N penalty function values of P-procedures. Then

$$P^{(1)} = 1-\left(\sum_{t=1}^{T_M-1} P^{(1)}_{Pt}\right)^N = 1-\int_0^\infty w^{(N)}_{PT_M}(\psi)\left(1-F_{P(T_M-1)}(\psi)\right)d\psi. \quad (26)$$

4.3 The probability of goal function extremum subdomain erroneous choice

One of the most important characteristics of the considered structural optimizations of PGAs is the probability P_{er} of the V-subdomain erroneous choice. First, let us consider the simplest case, when the domain $\Omega_{\bar{\alpha}}$ is decomposed into only two subdomains and one of them is of P-type and the other is of V-type. A subdomain of P-type will be chosen if the procedure operating in the P-subdomain is the first to reach the final T-th iteration, i.e. when the condition $\psi_{PT} < \psi_V$ is met, where ψ_{PT} – penalty function value for the P-procedure at the T-th iteration, ψ_V – penalty function value for the V-procedure that can theoretically perform from 1 to $(T-1)$ iterations. Let us assume we know a priori the value ψ_s of the penalty function which is considered to be a criterion of the V-subdomain choice. Then, to have a error of the V-subdomain choice it is necessary the simultaneous performing of two events: the penalty function of the V-procedure has exceeded the value ψ_s and of the P-procedure has not exceeded the value ψ_s. The probabilities of these events are determined by expressions

$$\int_{\psi_s}^{\infty} w_{VT}(\psi)\mathrm{d}\psi = 1-F_{VT}(\psi) \text{ and } \int_0^{\psi_s} w_{PT}(\psi)\mathrm{d}\psi = F_{PT}(\psi),$$

where $F_{VT}(\psi) = \int_0^{\psi} w_{VT}(x)\mathrm{d}x$, $F_{PT}(\psi) = \int_0^{\psi} w_{PT}(x)\mathrm{d}x$ – distribution function of ψ at the T-th iteration for procedures of V- and P-type correspondingly. Assuming the independence of the mentioned events the probability of the erroneous choice amounts to:

$$P_{er} = (1-F_{VT}(\psi))F_{PT}(\psi).$$

However, the value ψ_s is unknown a priory. Then in presence of the V-subdomain in the parameter definition domain the conditional probability $P_{er|V}$ of the choice of the P-procedure instead of the V-procedure is the probability of the event that $\psi_{VT} > \psi_{PT}$ at the T-th iteration:

$$P_{er|V} = \int_0^{\infty} w_{VT}(\psi) \int_0^{\psi} w_{PT}(x)\mathrm{d}x\mathrm{d}\psi = \int_0^{\infty} w_{PT}(\psi)(1 - F_{VT}(\psi))\mathrm{d}\psi .$$

Let us note that $w_{VT}(\psi)$ and $w_{PT}(\psi)$ depend on the distribution density $w(\hat{\bar{\alpha}}_0)$ of the initial approximation $\hat{\bar{\alpha}}_0$ of vector of the parameters to be estimated and in this sense they are considered to be estimated the conditional probabilities too. Below for definiteness, we assume that the initial approximation $\hat{\bar{\alpha}}_0$ for the V-procedure gives the worst estimate in the operating range and the sought probability $P_{er|V}$ corresponds to the upper bound of the error probability when choosing a V-subdomain.

It is easy to show the probability $P_{er|V}$ for the case of presence of one V-subdomain and $(N-1)$ P-subdomain in the parameter definitional domain is equal to

$$P_{er|V} = \int_0^{\infty} w_{VT}(\psi)\left(1 - \left(1 - F_{PT}(\psi)\right)^{N-1}\right)\mathrm{d}\psi . \tag{27}$$

Here, the penalty function values of the procedures operating in P-subdomains are considered to be independent. Let us note that assumed constraint on independence of local samples from different P-procedures is not rigid, because in real images the samples in the domains corresponding to P-procedures are almost noncorrelated.

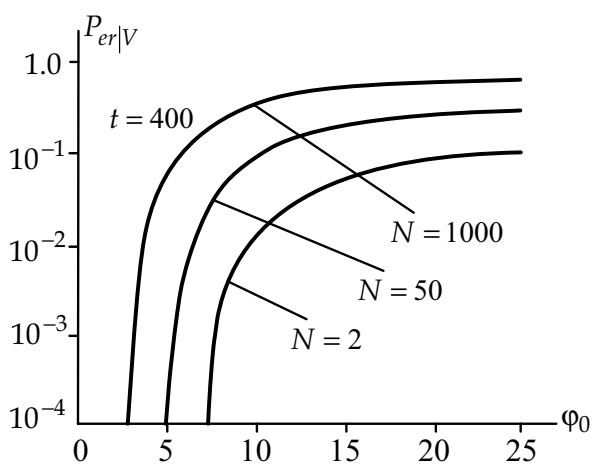


Fig. 8. The probability of omission of the sought fragment on the studied image

As an example, in Fig 8. graphs of the probability $P_{er|V}$ of omission of the sought fragment on the image to be studied are given. The IIGD parameters include parallel shift $\bar{h} = (h_1, h_2)^T$ of the sought and standard fragments and rotation angle φ between them. The calculation was carried out for the case of usage of the pseudogradient (10) and diagonal gain matrix with diagonal elements $\lambda_{ht} = 0.1/(1+0.1t)$ and $\lambda_{\varphi t} = 0.1/(1+0.01t)$ for $\bar{h}$ and φ, respectively. The situations when image size required the partitioning of the parameter definitional domain $\Omega_{\bar{\alpha}}$ into 2, 50 and 1000 subdomains have been studied. The presented

plots correspond to the 400th iteration of the estimation. Here, the initial approximation φ_0 of the rotation angle varied from 25 to 0 degrees and the initial approximation h_0 of the parallel shift was fixed and equal to 5 sample grid steps. The results have been obtained for the image parameters corresponding to the previous examples. On the basis of the graphs it is possible to find, for example, the maximum rotation angle at which the required probability $P_{er|V}$ is guaranteed. As φ_0 decreases the probability $P_{er|V}$ increases too, for example, if $\varphi_0 = 10^o$ then at $N = 50$ the probability $P_{er|V} \le 10^{-2}$ and at $N = 1000$ – $P_{er|V} \le 10^{-3}$.

Let us note that if the operating range of procedures exceeds the subdomains $\Omega_{\bar{\alpha}}^{(i)}\left(\hat{\bar{\alpha}}_{0i}\right)$, then several V-procedures can correspond to one extremum of the goal function which increases the error probability $P_{er|V}$ when doing V-subdomain search. It can be explained by the fact that in this situation the error condition at the V-subdomain choice is in a lesser than that of all V-procedures penalty function value of the P-procedure, having the largest number of iterations. To find the probability $P_{er|V}$ in this case we can use the expression (27). However, in doing so, the probability distribution $w_{VT}(\psi)$ at that has the sense of probability distribution of the penalty function minimum value among all V-procedures.

It is also of interest also to study the situation when the goal function has several extremums as it is, for example, in the problem of search of several similar objects locations on an image. In this case derivation of the corresponding relations enabling to find the probability of omission of one or several subdomains of extremum position does not cause theoretical difficulties.

4.4 Computational expense analysis

Probability distributions of the number of iterations, performed by thr V-procedure and P-procedures when attaining the threshold T-th iteration by one of the procedures, contain information necessary for analysis of average computational expenses.

Assume it is known a priori that there is one subdomain of V-type among N subdomains. Suppose also that a penalty function value of the leading procedure at the T-th iteration to be known and equaled to ψ_T. Then, the conditional probability of the event that the value ψ_{P1} of the penalty function of the P-procedure exceeds ψ_T after first iteration is

$$P_P(1|\psi_T) = P(\psi_{P1} > \psi_T) = \int_{\psi_T}^{\infty} w_{P1}(\psi)d\psi .$$

Here, on average $(N-1)P_P(1|\psi_T)$ procedures of $(N-1)$ P-procedures perform only one iteration. The probability $P_P(t|\psi_T)$ of the event that the excess of the value ψ_T occurs directly after the t-th iteration is equal to

$$P_P(t|\psi_T) = \int_{\psi_T}^{\infty} w_{Pt}(\psi)d\psi - \int_{\psi_T}^{\infty} w_{P(t-1)}(\psi)d\psi = \int_{\psi_T}^{\infty} w_{Pt}(\psi)d\psi - \sum_{i=1}^{t-1} P_P(i\,|\psi_T) . \qquad (27)$$

Here, on average $(N-1)P_P(t|\psi_T)$ procedures fulfill t iterations. The relation (27) at $t = \overline{1, T-1}$ corresponds to the conditional discrete distribution of the number of iterations

performed by the P-procedure. We can also obtain a similar conditional probability distribution $P_V(t|\psi_T)$, $t=\overline{1,T-1}$ for the V-procedure. The total average number of the procedures which have performed t iteration constitutes

$$N_m(t)=P_V(t|\psi_T)+(N-1)P_P(t|\psi_T).$$

To find unconditional probability distribution $\{P_t\}$, $t=\overline{1,T}$ it is necessary to take into account the fact that a value ψ_T is a random one. Here, the V-procedure will be the leading one with the probability $P(\min\psi_{PTi}>\psi_T)$, $i=\overline{1,N-1}$, where $\min\psi_{PTi}$ – minimum of penalty function values for P-procedures. If $\psi_{VT}>\psi_T$ then one of P-procedures will be the leading one with the probability $P(\psi_{VT}>\psi_T)=1-F_{VT}(\psi)$. Then, for the probability distribution $w(\psi_T)$ of the minimum penalty function value ψ_T for all procedures

$$w(\psi_T)=w_{VT}(\psi)\,P(\min\psi_{PTi}>\psi_T)+w_{PT}(\psi)\,P(\psi_{VT}>\psi_T). \tag{28}$$

On the assumption of noncorrelation of the local samples of P-procedures

$$P(\min\psi_{PTi}>\psi_T)=(1-F_{PT}(\psi))^{N-1}.$$

Taking into account the relations (27) и (28) for the unconditional distribution $\{P_t\}$ of the number of iterations we can write

$$P_t=\int_0^\infty w_T(\psi)\int_\psi^\infty\left(w_{Pt}(x)-w_{P(t-1)}(x)\right)dxd\psi,\ t=\overline{1,T-1}, \tag{29}$$

where

$$w_T(\psi)=w_{VT}(\psi)(1-F_{PT}(\psi))^{N-1}+w_{PT}^{(N-1)}(\psi)(1-F_{VT}(\psi)); \tag{30}$$

$w_{PT}^{(N-1)}(\psi)$ – probability distribution of the penalty function minimum value for P-procedures at the T-th iteration.

Let us note if we need calculation of all the components of the distribution (29) then to reduce computational expenses it is convenient to use the following recurrent relation

$$P_t=\int_0^\infty w_T(\psi)(1-F_{Pt})d\psi-\sum_{i=1}^{t-1}P_{Pi}, \tag{31}$$

and for calculation of the probability $P_{t-l,t}$ of finding of the number of iterations in the range from $(t-l)$ to t, $l=\overline{2,t-1}$ the expression

$$P_{t-l,t}=\int_0^\infty w_T(\psi)\int_\psi^\infty\left(w_{Pt}(x)-w_{P(t-l)}(x)\right)dxd\psi.$$

In absence of the goal function extremum in the parameter definitional domain the expression (30) is simplified $w_T(\psi)=w_{PT}^{(N)}(\psi)$.

Then for the distribution $\{P_t\}$ we obtain

$$P_t = \int_0^{\infty} w_{PT}^{(N)}(\psi)(1-F_{Pt})d\psi - \sum_i^{t-1} P_{Pi}\ ,\ t=\overline{1,T-1}\ . \tag{32}$$

In Fig. 9 examples of the probability distribution of number of iterations for the problem of fragment search on the image for the situations of presence (Fig. 9,a) and absence (Fig. 9,b) of the sought fragment on the image are presented. The results have been obtained using the relations (31) and (32) for the case of attainment of the 100th iteration by the leading procedure. The fragment shift with reference to the subdomain center was equal to 5 sample grid steps, interframe correlation sample coefficient was used as the goal function, the pseudogradient (10) and diagonal gain matrix with elements $\lambda_{ht} = 0.05/(1+0.04t)$, $g=100$ were applied in procedures. For the given example in absence of the sought fragment on the image (in absence of the goal function extremum in the parameter definitional domain) the total number of all the procedures is about 2.3 times as large.

In the diagrams the experimental results obtained at the same parameters on the simulated Gaussian images, when the image domain was decomposed into 900 subdomains averaged on 200 realizations are also shown (with circles). A good correspondence of the theoretical and the experimental results can be seen. The performed simulation showed that at $T>200$ the distribution $\{P_t\}$ is normalized which enables to use Gaussian approximation.

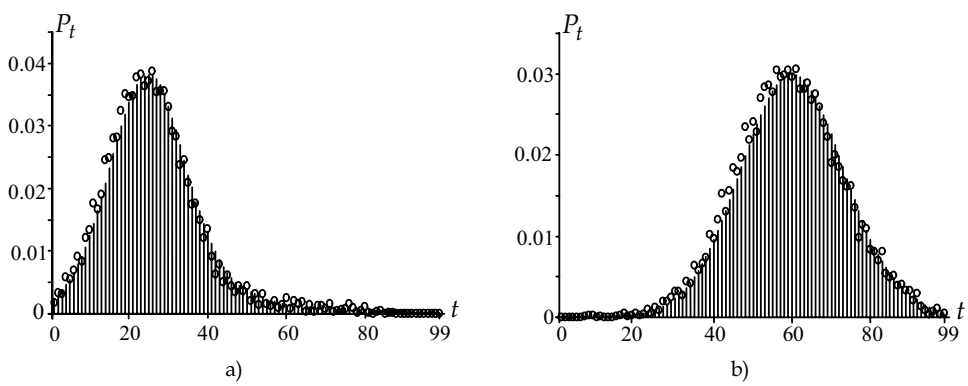


Fig. 9. Probability discrete distributions of the number of iterations

When using structural optimization the computational expenses depend on both the type of pseudogradient procedures to be used and computational resources due to which the algorithm is realized. Let us consider only approximate relations primarily characterizing the total number of iterations performed by the procedures to attain the required result. Assume the time necessary to perform one step of the PGA to be the sum of the time τ_I, spent on performing of one iteration by the procedure, and the time τ_ψ of finding the procedure that has the least value of the penalty function (for example, by arranging the penalty function values of all the procedures in ascending order and by choosing the procedure leading in this arranged sequence).

The computational expenses $E^{(1)}$ on testing of the hypothesis about absence of the V-subdomain at the given error $P^{(1)}$ of the first type are the simplest to find. In this case the

threshold value of the statistical criterion of hypothesis acceptance is the number M_{c1} of iterations performed by all procedures. Accordingly,

$$E^{(1)} = M_{c1}\left(\tau_I + \tau_\psi\right).$$

If the hypothesis is rejected then the number of iterations T_M is chosen as a rule basing on the required probability of V-subdomain omission. Then for the mathematical expectation of the total computational expenses E_T we can write

$$E_T = \left(\tau_I + \tau_\psi\right)\left(T + (N-1)\sum_{t=1}^{T-1} tP_t\right),$$

where P_t – probability of the event that the procedure has performed t iterations; N – total number of iterations. If all procedures have attained T iterations then the computational expenses constitute at least

$$E_{TA} = \tau_I T N .$$

Accordingly the gain in computational expenses when using structural optimization in comparison with the case when all procedures are brought to the threshold number of iterations is determined by the relation

$$G = \frac{E_{TA}}{E_T} = \frac{\dfrac{\tau_I}{\tau_I + \tau_\psi}}{\dfrac{1}{N} + \left(\dfrac{1}{T} - \dfrac{1}{NT}\right)\sum\limits_{t=1}^{T-1} t\, P_t}. \tag{33}$$

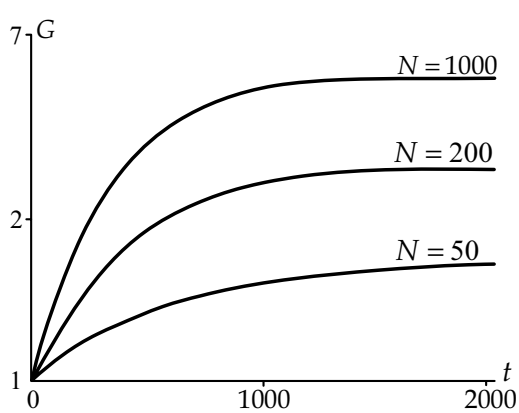
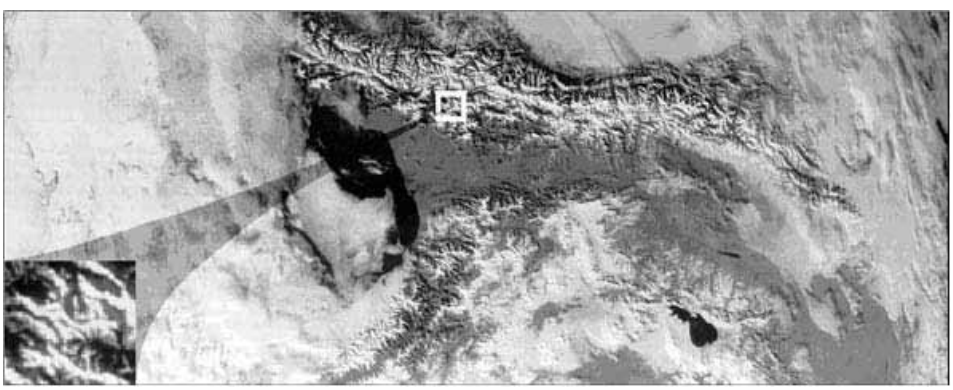


Fig. 10. Gain in computational expenses when using structural optimization

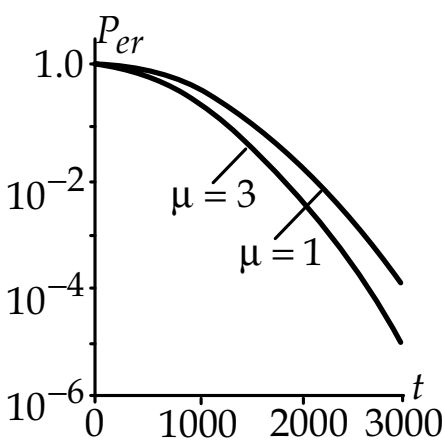
As an example, diagrams of the gain G in computational expenses at $N = 50$, 200 and 1000 obtained by means of the relation (33) for the problem of image fragment search are

presented if Fig. 10. Here, we assumed that $\tau_\psi = 0.1\,\tau_I$. It is seen that the gain depends on the number N of subdomains of the parameter definitional domain. Thus, at t =100 for $N = 50$ the gain $G = 1.6$, for $N = 200$ – $G = 2.5$, for $N = 1000$ – $G = 5.3$.

Structural optimization of PGAs was used, for example, to solve the problem of small fragments location search on large-sized images with reference to a given standard fragment. Here, the sought and the standard fragments had reciprocal rotation angle, different scale and amplitude distortions. In Fig. 11,a) an example of image of 3048×1608 elements and the sought fragment of 48×48 elements is given. The number of subdomains of image decomposition has constituted $N = 31200$ and computational expense reduction in comparison with the traditional approach – more than 15 times as large. In Fig. 11,b graphs of dependence of the probability P_{er} of fragment omission versus the number of iterations at the local sample size $\mu = 1$ and 3 (under the condition that the sought fragment does exist on the image) are shown.



a)



b)

Fig. 11. Image, fragment and error probability of fragment omission

5. Conclusion

The considered PGAs can be directly used in various applied problems of image processing. The algorithms of this class can be applied to image processing in the conditions of a priori uncertainty, they assume small computational expenses and do not require the preliminary estimation of the parameters of the image to be studied. The estimates formed through them are immune to impulse interference and converge to optimal values under rather weak conditions. At an unknown set of the parameters of geometrical deformations model PGAs enable to estimate shifts of each node of image sample grid. At a given IIGD model the processing of the image samples can be performed in an arbitrary order, for example, in order of scanning with decimation that is determined by the hardware speed, which facilitates obtaining a tradeoff between image entering rate and the speed of the available hardware. The mentioned properties make them attractive for usage in real time systems.

Unfortunately a limited size of this manuscript does not make it possible to consider some important aspects of this lead of investigations, in particular, the analysis of probabilistic properties and computational expenses at PGAs structural optimization for the situation when the goal function has several extremums. Let us note two more such aspects for further study in the form of the problem definition.

A disadvantage of the PGAs when performing the processing of real images is in the presence of local extremums of the goal function estimate characterizing the estimation quality which significantly reduces convergence speed or even may lead to its failure at some realizations in the process of estimate convergence. Besides, algorithms of this class have a comparably small range of operating.

At that the estimate convergence character and computational expenses in many respects depend on the image samples local sample size used on various iterations of estimation. Thus the development and study of the methods of a priori and a posteriori optimization of size and plan of the sample used to obtain the goal function pseudogradient is considered to be a vital problem. One of the trends of a posteriori optimization is planned in the part 3.3 in this work. Of works concerned with a priori optimization we can highlight (Samojlov, 2006; Minkina et al., 2005).

Modern information systems are characterized by increasing rate of the entering data. It gives rise to the vitality of pseudogradient procedures optimization on criteria of computational expense minimum and iterations number minimum at limitations on computational expenses. We should note that many scientists addressed to the study of precision potentiality of the pseudogradient procedures, in particular, (Albert & Gardner, 1967; Benveniste et al., 1990). Asymptotic rate of convergence of the estimates to be formed has been profoundly studied in works (Chung, 1954; Sacks, 1958), in works (Goodwin & Payne, 1977; Soderstrom, 1981) and others the conditions of asymptotic normality of various pseudogradient procedures have been found, the works (Kushner & Clark, 1978; Tsypkin & Polyak, 1974) are devoted to estimation of asymptotic rate of convergence. However, for practical application of these procedures the investigation of their precision potentiality at a finite number of iterations is of significant importance. Unfortunately, at present this issue has been studied insufficiently. It is due to the fact that at a finite number of iterations an analysis of interframe deformations parameter estimates probabilistic properties is complicated by a large number of factors whose effect cannot be ignored. These are the nature of probability densities and autocorrelation functions of images and interfering noise, the kind of goal function determining the quality of estimation, the parameters of the

procedures and the number of iterations. Besides, when estimating the image parameters we have to deal with complex assemblage of hindering factors such as time and spatial inhomogeneity of characteristics of the desired signals and noise, inhomogeneity of sensitivity and the faults in sensors, pulse interference, etc. These above-mentioned factors are of random nature, so when describing real images both parametric and non-parametric a priori uncertainty nearly always takes place. One of the techniques of analysis of probabilistic characteristics of the estimates, formed by PGAs at a finite number of iterations has been proposed in works (Tashlinskii & Tikhonov, 2001; Tashlinskii, 2004) and was used when developing the chapters 3.1, 4.3 and 4.4 of the present manuscript. However, this lead requires further serious investigations.

6. References

Albert, A. & Gardner, L. (1967). *Stochastic approximation and nonlinear regression*, MIT-Press, Cambridge, Massachusetts

Benveniste, A.; Metivier, M. & Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, Berlin

Chung, K. L. (1954). On a stochastic approximation method. *The Annals of Mathematical Statistics*, Vol. 25, No. 3, pp. 463–483

Gonzalez, R. C. & Woods, R. E. (2002). *Digital Image Processing*. Prentice Hall, New Jersey, ISBN 0-201-18075-8

Goodwin, G. C. & Payne, R.L. (1977). *Dynamic system identification: Experimental design and data analysis*. Academic Press, New York

Korn, G. A. & Korn, T. M. (1968). *Mathematical handbook for scientists and engineers*. McGraw-Hill Book Company, New York, San Francisco, Toronto, London, Sydney

Krasheninikov, V. R. (2003). *Osnovi teorii obrabotki izobrazhenii [The Fundamentals of image processing theory]*, UlGTU, ISBN 5-89146-434-9, Uljanovsk [in Russian]

Kushner, H.J. & Clark, D.S. (1978). *Stochastic approximation methods for constrained and unconstrained systems.* Springer-Verlag, New York

Minkina, G. L. & Samojlov, M. U. (2005). Choice of Values Characterizing Estimate Convergence for Pseudogradient Estimation of Image Interframe Deformation Parameters, *Vestnik UlGTU [Herald of the UlSTU]*, No. 4, pp. 32–37, ISSN 1674-7016 [in Russian]

Minkina, G. L.; Samojlov, M. U. & Tashlinskii, A. G. (2007). Choice of The Objective Function for Pseudogradient Measurement of Image Parameters. *Pattern Recognition and Image Analysis*, Vol. 17, No. 1, pp. 136–139, ISSN 1054-6618

Polyak, B. T. & Tsypkin, Ya. Z. (1973). Pseudogradient Algorithms of Adaptation and Education. *Avtomatika i telemechanika [Automation and Telemechanics]*, No. 3, pp. 45–68, ISSN 0005-2310 [in Russian]

Polyak, B. T. (1976). Convergence and Convergence Speed of Iterative Stochastic Algorithms: a General Case. *Avtomatika i telemechanika [Automation and Telemechanics]*, No 2, pp. 83–94, ISSN 0005-2310 [in Russian]

Polyak, B. T. & Tsypkin, Ya. Z. (1984). Criterion Algorithms of Stochastic Optimization. *Avtomatika i telemechanika [Automation and Telemechanics]*, No. 6, pp. 95–104, ISSN 0005-2310 [in Russian]

Sacks, J. (1958). Asymptotic distribution of stochastic approximation. *The Annals of Mathematical Statistics*, Vol. 29, No. 2, pp. 373–405

Samojlov, M. U. (2006). Procedure Optimization of Image Interfarme Geometrical Deformation Parameter Pseudogradient Estimation. *Radiolocation, Navigation, Connection: Proc. of The XII Inter. Sci.-Tech. Conf.*, Vol. 1, pp. 162–167, ISBN 5-9900094-8-8, Voronezh, March, 2006, Sakvoee, Voronezh

Soderstrom, T. (1981). *On a method for model selection in system identification*. Automatica, Vol. 13, No. 2, pp. 387–388

Tashlinskii, A. G. (2000). *Otsenivanie parametrov prostranstvennih deformatsii posledovatelnostei izobrazhenii [Image Sequence Spatial Deformation Parameters Estimation],* UlGTU, ISBN 5-89146-204-4, Uljanovsk [in Russian]

Tashlinskii, A. G. & Tikhonov, V. O. (2001). The Method of Multidimensional Process Parameter Pseudogradient Measuring Error Analysis. *Izvestija Vuzov: Radioelektronika, [Proceedings of institutes of higher education: Radio Electronics]*, Vol. 44, No. 9, pp. 75–80, ISSN 0021-3470 [in Russian]

Tashlinskii, A. G. (2002). Pseudogradient estimation of image sequence spatial deformations. *A Publication of The International Association of Science and Technology for Development – IASTED*, pp. 382–385, ACTA Press, ISBN 0-889860342-3, Anaheim, Calgary, Zurich

Tashlinskii, Alexandr (2003). Computational Expenditure Reduction in Pseudo-Gradient Image Parameter Estimation. *Computational Science*, Vol. 2658, No. 2, pp. 456-462, ISSN 0302-9743

Tashlinskii, A. G. (2004). Probabilistic Modeling of Image Parameters Pseudogradient Measuring Process. *7-th International Conference PRIA-7-2004*, Vol. II, pp. 402–403, ISBN 5-7629-0631-0, St. Peterburg, October, 2004, Nauka/Interperiodica, St. Peterburg

Tashlinskii, A. G. (2006). Structural Optimization of Pseudogradient Algorithms for Estimating Image Parameters. *Pattern Recognition and Image Analysis*, Vol. 16, No. 2, pp. 218–222, ISSN 1054-6618

Tsypkin, Ya. Z. & Polyak, B. T. (1974). Attainable Accuracy of Adaptation Algorithms. *Dokladi AN SSSR [Proceedings of USSR Academy of Sciences]*, Vol. 218, No. 3, pp. 532–535, ISSN 0002-3264 [in Russian]

Tsypkin, Ya. Z. (1995). *Informatsionnaya teoriya identifikatsii [Information theory of Identification]*, Nauka/ Fizmatlit, ISBN 5-02-015071-1, Moskow [in Russian]

Vasiliev, K. K. & Tashlinskii, A. G. (1998). Estimation of Deformation Parameters of Multidimensional Images to Be Observed on The Background of Interference. *Pattern Recognition and Image Analysis: New Information Technologies, Proceedings of the IV Inter. Conf.*, Vol. 1, pp. 261–264, Novosibirsk, June, 1998, SO RAN, Novosibirsk, ISBN 5-89896-189-5

26

Anisotropic Filtering Techniques applied to Fingerprints

Shlomo Greenberg and Daniel Kogan
Ben-Gurion University of the Negev
Israel

1. Introduction

Noise filtering of images is basically a smoothing process, and it is a subject that has been addressed for many years. The idea of adaptive smoothing is being investigated a long time and many different approaches have been proposed over the years.
Mastin (1985) reported superior performance of nonlinear such as medina filtering over linear techniques applied for adaptive image smoothing. Zucker et al. (1977) proposed to perform adaptive smoothing using weighted mask, which is computed by the difference between the value of the center point and its neighbors. Wang et al. (1981) applies a weighting scheme that averages values within a sliding window and changes the weights according to local differential. Instead of basic averaging Davis (Davis & Rozenfeld, 1978) performs iterated local noise cleaning by K-Nearest Neighbour averaging. The main disadvantage of these methods is their difficulty to ensure convergence.
Blake (Blake & Zisserman, 1987) proposed a smoothing process, which reconstructs a noisy signal in a piecewise continuous manner by employing weak continuity constraints. Although the convergence behavior was well studied, the computational complexity is extremely high. An anisotropic diffusion scheme was presented by Perona & Malik (1990). They suggested to employ a heat equation in anisotropic medium for edge enhancement. This is done by selectively smoothing regions with low gradient. Another approach, called Forward-and-Backward diffusion, is presented by Smolka et al. (2003) and emphasizes regions with high gradient which are not caused by noise. Almansa (Almansa & Lindeberg, 2000) and Weickert (2001) have used diffusion techniques, which are based on a multi-scale analysis called scale-space representation, and applied an iterative process for local features estimation. Diffusion methods tend to distort sloping edges, while iterative methods slow down the filtering process in images with considerable amount of noise.
Steerable filters are a class of filters, in which a filter of arbitrary orientation is synthesized as a linear combination of a set of "basis filters" (Freeman & Adelson, 1991). Steerable filters are used in many image-processing tasks and specifically in image enhancement. Steerable-scalable kernels roughly shaped like Gabor functions have the advantage that they can be specified and computed easily (Perona, 1992). However, those filters usually approximate the orientation with low resolution, since they are usually based on angular frequency sampling, and a huge number of basis filters are required in order to approximate orientation steerability with high resolution (Yu et al., 2001). Another kind of structure-

adaptive anisotropic filtering technique has been proposed by Yang et al. (1996). Instead of using local gradients as a means of controlling the anisotropism of filters, it uses both a local intensity orientation and an anisotropic measure to control the shape of the filter. Although the filters proposed by Yang and Almansa are both structure-adaptive anisotropic filters, they are still significantly different mostly by the fact that Almansa's filter is iterative while Yang's is not.

We propose to improve the structure-adaptive anisotropic filter (Yang et al., 1996) in the space domain. We suggest changing the filter's kernel from a circle to an ellipse with the form, size and direction depending on image local anisotropic features. The essential idea of the improved filter is to apply a median filter for pixels bounded by an anisotropic elliptical kernel. We propose to use a non-linear filter kernel function rather than a linear, which produces less blurring during image filtering. The non-linear function is implemented in the form of the median filter. Moreover, instead of using Yang's derivatives-based method for estimation the oriented pattern direction we have used Donahue's (Donahue & Rokhlin, 1993) method, which is more robust to noise. It uses a gradient type local operator and least squares minimization to control the noise.

Fingerprints are today the biometric features most widely used for personal identification. The uniqueness of a fingerprint is determined by the local ridge characteristics and their relationships. Most of today's automatic systems used for fingerprint comparison are based on minutiae matching, which represents local discontinuities in a fingerprint image. An automatic fingerprint image matching process, which enables a personal identification, strongly depends on comparison of the minutiae points of interest MPOI and their relationships. Reliable automatic extraction of these MPOI is a critical step in fingerprint classification. The performance of minutiae extraction algorithm relies heavily on the quality of the fingerprint images (Hong et al., 1996). The ridge structures in poor-quality fingerprint images are not always well defined and, hence, cannot be correctly detected. In order to ensure robust performance of minutiae extraction algorithm an enhancement algorithm that improves the clarity of the ridge structures is necessary (Hong et al., 1996). Enhancement of ridge structures essentially involves some filtering operation.

We propose to modify the structure-adaptive anisotropic filter in the frequency domain by converting it from a lowpass filter into a band-pass one. In this work we show that the modified structure-adaptive anisotropic filter can be effectively applied to applications, such as fingerprint image enhancement, in which the oriented patterns in local neighborhood form a sinusoidal-shaped plane wave with a welldefined frequency and orientation i.e., ridges and valleys in a fingerprint image. Adjustment of the modified filter to fingerprints is made resulting in a unique structure-adaptive anisotropic filter. The performance of the unique structure-adaptive anisotropic filter is compared to that of some other filters in the framework of minutiae detection process.

2. The structure-adaptive anisotropic filter

The structure-adaptive anisotropic filter, which has been proposed by Yang, uses a local intensity orientation and an anisotropic measure of level contours to control the shape and extent of the filter kernel. The filter kernel applied at each point x_0 is defined as follows (Yang et al., 1996):

$$k(x_0,x)=\rho(x-x_0)\exp\left\{-\left[\frac{((x-x_0)\cdot n)^2}{\sigma_1^2(x_0)}+\frac{((x-x_0)\cdot n_\perp)^2}{\sigma_2^2(x_0)}\right]\right\} \tag{1}$$

where n and $n_\perp$ are mutually normal unit vectors, and n is parallel with the local oriented pattern direction. The shape of the kernel is controlled through $\sigma_1^2(x_0)$ and $\sigma_2^2(x_0)$, ρ satisfy the condition $\rho(x)=1$ when $|x|<r$, and r is the maximum support radius.

Direction estimation of an oriented pattern is based on the fact that the power spectrum of such a pattern lies along a line through the origin in the Fourier domain, and the direction of the line is perpendicular to the dominant spatial orientation of the pattern. The evaluation of the Fourier transform, however, is not necessary for actual calculations. Through a simple relationship between the local orientation direction and the matrix eigenvectors, the estimation of oriented pattern direction $\theta(x)$ [the direction of vector n in (1)] can be made (Yang et al., 1996) as follows:

$$\theta(x)=\frac{1}{2}\tan^{-1}\left\{\frac{\iint_\Omega 2\cdot\left(\frac{\partial f}{\partial x_1}\right)\left(\frac{\partial f}{\partial x_2}\right)dx_1dx_2}{\iint_\Omega\left(\frac{\partial f}{\partial x_1}\right)^2-\left(\frac{\partial f}{\partial x_2}\right)^2dx_1dx_2}\right\}+\frac{\pi}{2} \tag{2}$$

where Ω is a local neighborhood $x=(x_1,x_2)$.

The space constants $\sigma_1^2(x_0)$ and $\sigma_2^2(x_0)$ are controlled through the corner detector $c(x_0)$ and by the measurement of anisotropism $g(x_0)$ as follows (Yang et al., 1996):

$$\sigma_1(x_0)=\frac{r}{1+c(x_0)/\beta} \tag{3}$$

$$\sigma_2(x_0)=(1-g(x_0))\frac{r}{1+c(x_0)/\beta} \tag{4}$$

where β is a normalization factor that controls how faithfully the corners and junctions can be preserved during the filtering process.

The anisotropic measure gives an indication of how strong a pattern is oriented and is defined as follows (Yang et al., 1996):

$$g(x)=\frac{\left\{\iint_{\Omega}\left(\frac{\partial f}{\partial x_1}\right)^2-\left(\frac{\partial f}{\partial x_2}\right)^2 dx_1dx_2\right\}^2+\left\{\iint_{\Omega}2\cdot\left(\frac{\partial f}{\partial x_1}\right)\left(\frac{\partial f}{\partial x_2}\right) dx_1dx_2\right\}^2}{\left\{\iint_{\Omega}\left(\frac{\partial f}{\partial x_1}\right)^2+\left(\frac{\partial f}{\partial x_2}\right)^2 dx_1dx_2\right\}^2} \tag{5}$$

which can be calculated directly from the original data $f(x_1, x_2)$ and its partial derivatives. The anisotropic measure can also provide a convenient way of finding corner and junction points within a given image. Yang suggests using both the measure of anisotropism and a gradient strength for an estimation of corner strength in a following way (Yang et al., 1996):

$$c(x)=(1-g(x))|\nabla f(x)|^2 \tag{6}$$

3. Improved structure-adaptive anisotropic filter

In this section we propose some improvements to the structure-adaptive anisotropic filter (Yang et al., 1996) in the space domain. As previously mentioned the essential idea of the improvement is to apply the median filter for pixels bounded by an anisotropic elliptical kernel.

The structure-adaptive anisotropic filter suffers from the following problems: corner strength measure $c(x)$ is highly influenced by noise. This results in a wrong estimation of space constants $\sigma_1^2(x_0)$ and $\sigma_2^2(x_0)$, which control the shape of the filter kernel. Derivatives-based approach for oriented pattern direction estimation fails to produce correct estimates for noisy images. The normalization factor β controls how faithfully the corners and junctions can be preserved during the filtering process. Therefore, it is a critical factor and the choise of β significantly affects the filter performance. Despite the fact that the structure-adaptive filter is directional and adjusts the shape of the kernel according to image anisotropic local features, the filter causes to unnecessary blurring in processed image due the linearity of its filtering function. The structure-adaptive anisotropic filter operates on a pixels neighborhood of a constant size and moreover, the size is not depending on local features of input image. We suggest a solution to the above-mentioned problems and propose the improved structure-adaptive anisotropic filter, which combines non-linear filtering function, a more robust to noise technique for oriented pattern direction estimation and elliptical kernel with its form, size and direction depending on image local anisotropic features.

Instead of using a constant kernel the size must be changed to embody image local anisotropic features, namely, the anisotropic measure $g(x)$ and the corner strength $c(x)$. This can be achieved by defining an elliptical kernel (Fig. 1) with its principal axis and direction changing according to image local anisotropic features.

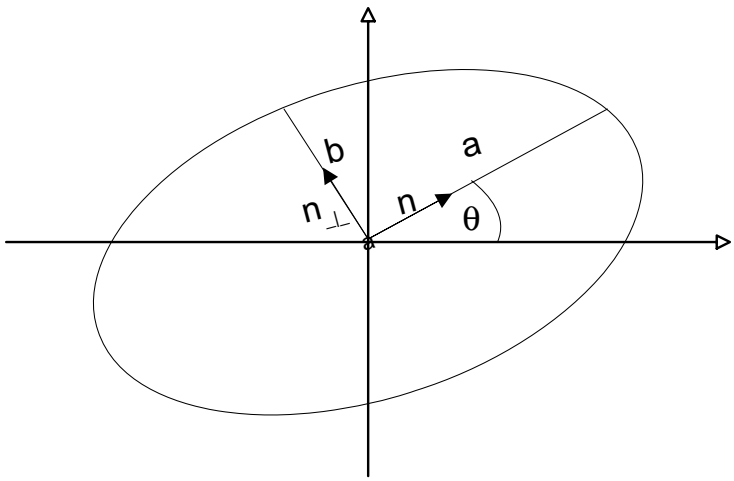


Fig. 1. Controlling the shape and direction of an elliptic kernel through principal axes a, b and direction θ, which are controlled through image local anisotropic features.

The elliptical kernel's main axis $a(x_0)$ must be minimal in regions where there is a high number of corners (edges, corners and etc) and maximal in regions with no corners (smooth places). The transition of $a(x_0)$ from smooth regions to regions that include corners should be performed in an exponential manner in order to prevent smoothing corners. The kernel's shape must be circular in regions with low values of anisotropic measure ($g \to 0$). In regions with high anisotropic measure ($g \to 1$) the shape obtains a highly oriented elliptical form with its main axis runs in parallel to direction of local oriented pattern as can be seen in Fig. 2. To meet these requirements we propose to define the principal axes of the elliptical kernel as follows:

$$a(x_0) = r \cdot \exp\left(\frac{-c(x_0)}{\beta}\right) \tag{7}$$

$$b(x_0) = a(x_0) \cdot \left(1 - g(x_0) + \varepsilon\right) \tag{8}$$

where r is the maximal support radius, ε is the minimal kernel width and β is a normalization factor that will be defined later.

Comparing the behavior of Yang's space parameter $\sigma_1(x_0)$ (3) with the new proposed $a(x_0)$ parameter (7) emphasis the improvement of the new filter. Fig. 3 demonstrates the behavior of controlling the main axis of the filter kernel for both Yang's and the proposed filter. This figure shows that adopting the proposed filter results with better behavior relative to the corner strength $c(x)$. For high values of corner strength the elliptical kernel results with smaller values for the main axis $a(x_0)$ in comparison to Yang's filter, and only few pixels

are collected by the kernel. Therefore, it suggests that the improved filter better preserves corners and edges in the input image.

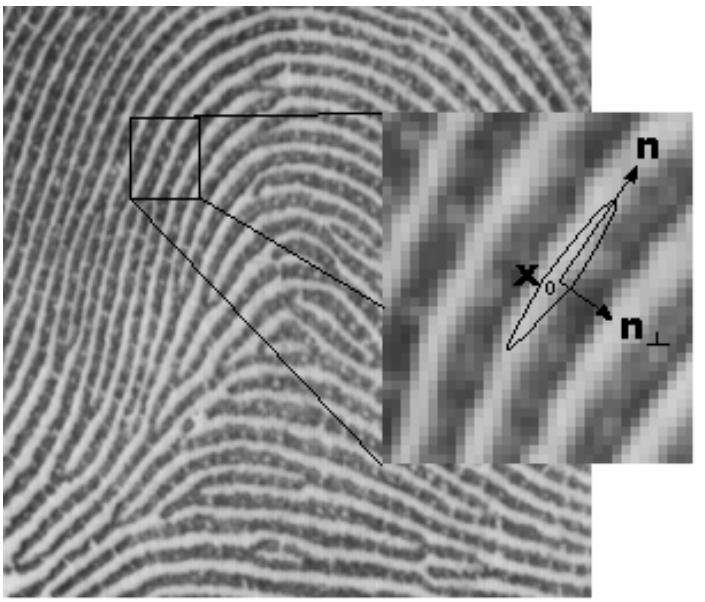


Fig. 2. Controlling the shape and direction of the kernel in the space domain.

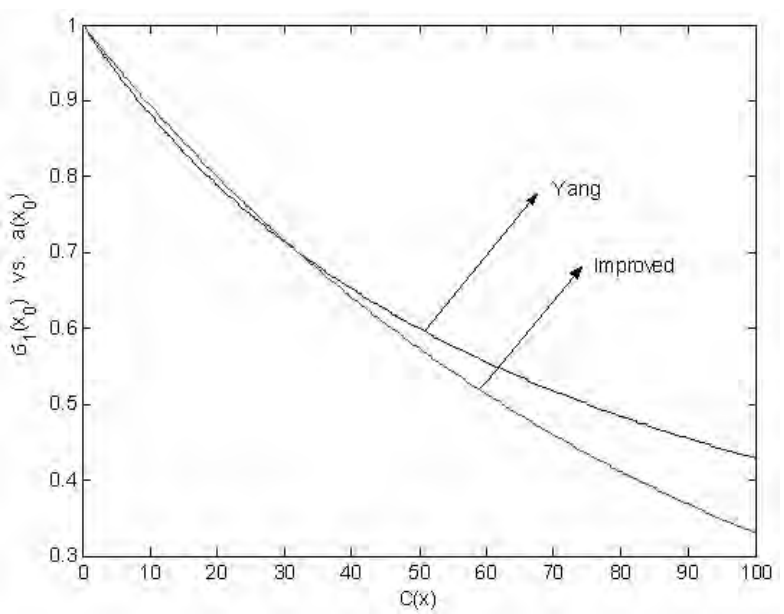


Fig. 3. Controlling the main axis of the filter kernel for Yang's and the improved filter.

Correct estimation of oriented patterns direction is of high importance for effective performance of directional filters such as structure-adaptive anisotropic filter. The technique, which was proposed by Yang, is based on the fact that the power spectrum of an oriented pattern lies along a line through the origin in the Fourier domain, and the direction of the line is perpendicular to the dominant spatial orientation of the pattern. This technique fails to produce correct estimates for noisy images. The proposed filter adopts the method, proposed by Donahue & Rokhlin (1993), which uses a gradient-type operator and least-squares minimization to control the noise. The estimation of the oriented pattern direction is extracted from each $2x2$ pixels neighborhood, which is then averaged over a local window of 5x5 pixel size.

We propose to use a non-linear filter kernel function rather than a linear one that produces less blurring during image filtering. We have implemented the non-linear function in the form of a median filter that is applied within neighborhoods of pixels bounded by an elliptical kernel. The obtained is the improved structure-adaptive anisotropic filter, which is expressed mathematically as follows:

$$k(x_0,x)=median\{f(x),\ \ x\in \mathrm{P}(x-x_0)\} \tag{9}$$

where $\mathrm{P}(x-x_0)$ is the elliptical kernel centered at pixel x_0, oriented at angle θ (defined by mutually normal unit vectors n and $n_\perp$) and is defined as follows:

$$\frac{((x_1-x_{01})\cdot n)^2}{a(x_0)^2}+\frac{((x_2-x_{02})\cdot n_\perp)^2}{b(x_0)^2}\le 1 \tag{10}$$

Substituting (7) and (8) into (10) and incorporating the obtained equation into (9) results in a final expression of improved structure-adaptive anisotropic filter:

$$k(x_0,x)=median\left\{f(x),\ \ x\in\left(\frac{((x_1-x_{01})\cdot n)^2}{\left(r\cdot\exp\left\{\frac{-c(x_0)}{\beta}\right\}\right)^2}+\frac{((x_2-x_{02})\cdot n_\perp)^2}{\left(r\cdot\exp\left\{\frac{-c(x_0)}{\beta}\right\}\cdot(1-g(x_0)+\varepsilon)\right)^2}\le 1\right)\right\} \tag{11}$$

Normalization factor β used in the structure-adaptive anisotropic filter (Yang (1996)) is set to 75 percent of the maximal value of $c(x)$. Appropriate choice of β involves a trade off between effective smoothing of areas with no corners (higher β) and preserving most of the corners in the image (setting lower β). Fig. 4 shows the PSNR of a reconstructed image using different β values. Fig. 4 suggests using higher values of β to obtain higher PSNR

values for the reconstructed image. However, Fig. 5 which demonstrates the reconstructed image, shows that the image is over smoothed while using high value of β. Therefore, in order to preserve most of the corners and to effectively smooth areas without corners, we suggest setting β to 90 percent of the maximal value of $c(x)$. This compromise is empirically achieved by testing different values for β.

The performance of the improved structure-adaptive anisotropic filter is compared to the structure-adaptive anisotropic filter (Yang, 1996) and to the conventional median filter. The conventional median filter is carried out by numerical sorting of all pixel values in a surrounding neighborhood of 3x3 pixels, and then replacing the pixel being considered with the middle pixel value.

The maximal support radius used for the structure-adaptive anisotropic filter is 3 pixels as suggested by Yang (1996), while the maximal support radius for the improved structure-adaptive filter was empirically set to 2 pixels. The comparison is carried out on different kinds of images, which are commonly used for testing noise filtering algorithms (Fig.). All the test images are of resolution 79x79 dots per inch. The size is 182x144 for Bird image, 95x95 for Girl and Area images, and 256x256 pixels for Fingerprint image.

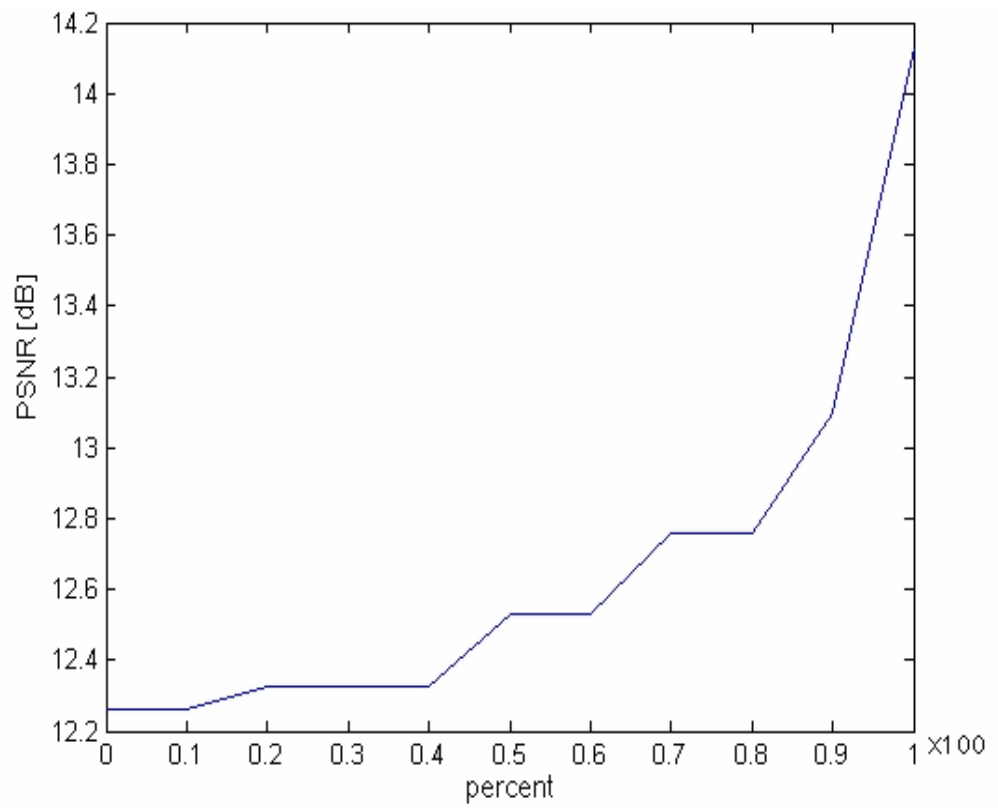
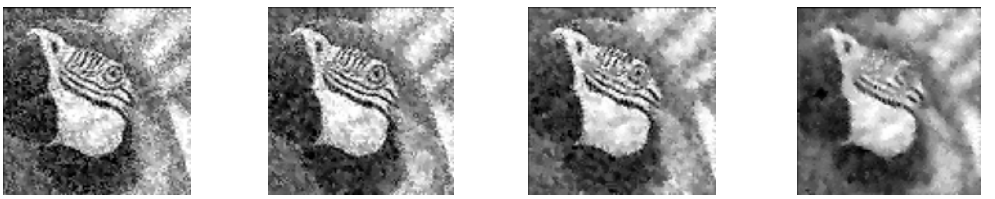
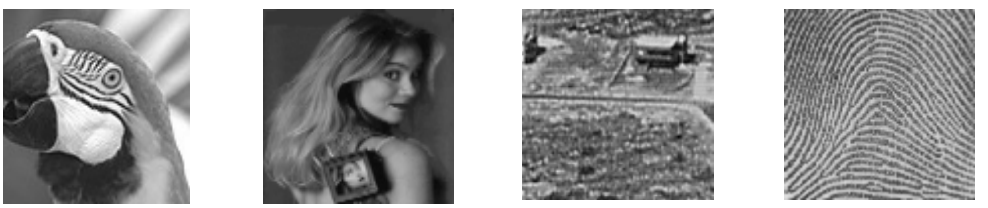


Fig. 4. PSNR of reconstructed Bird image using different β



(a) (b) (c) (d)

Fig. 5. Result of applying the improved filter with different β : (a) 50%, (b) 75%, (c) 90% and (d) 100% of the maximal value of $c(x)$.



(a) (b) (c) (d)

Fig. 6. Test images: (a) Bird ,(b) Girl (c) Area and (d) Fingerprint image

The filters were tested on images contaminated by Gaussian noise and 'Salt and Pepper' noise for different SNR (Signal to Noise) levels. Figure 7 demonstrates the Bird image contaminated with 'Salt and Pepper' noise and Gaussian noise.

(a) (b) (c)

Fig. 7. (a) original Bird image, contaminated with (b) 'Salt and Pepper' noise with SNR=21dB, and (c) Gaussian noise with SNR=20dB.

The performance comparison is based on the SAD (Sum of Absolute Differences) criteria, which computes the sum of absolute differences between the original image and the reconstructed one. Figure 8 shows the filters performance results for Gaussian noise for different levels of SNR ranging from 15dB to 80dB. Figure 9 shows the performance results for 'Salt and Pepper' noise for different levels of SNR ranging from 16dB to 52dB. Yang's structure-adaptive anisotropic filter performs better than the median filter. The improved structure-adaptive anisotropic filter outperforms both median and Yang's structure-adaptive anisotropic filter over the whole range of SNR levels. Figure 10 shows some examples of applying the different filters on images contaminated by Gaussian noise. It can be seen that the Yang's structure-adaptive anisotropic filter reconstructs better than the

median filter, and the improved structure-adaptive anisotropic filter outperforms both filters in reconstructing from noisy image.

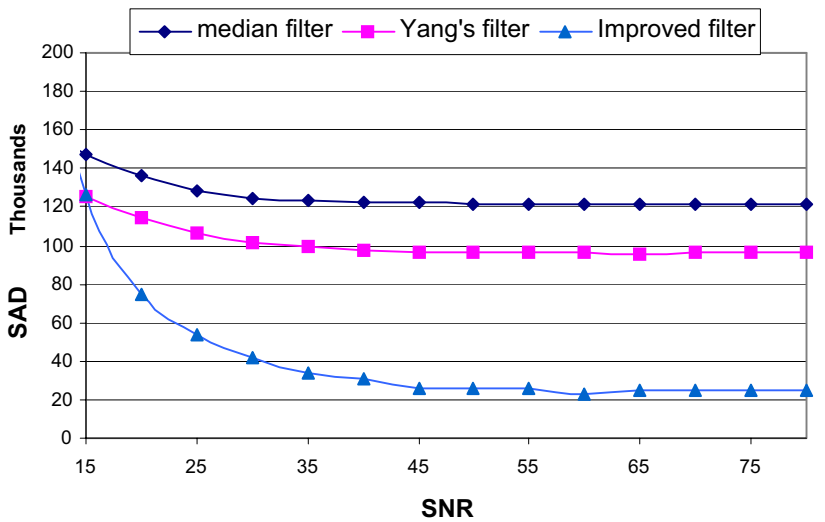


Fig. 8. Performance results of applying a conventional median filter, Yang's structure-adaptive anisotropic and the improved structure-adaptive anisotropic filters on the Bird image contaminated by a Gaussian noise at different SNR levels.

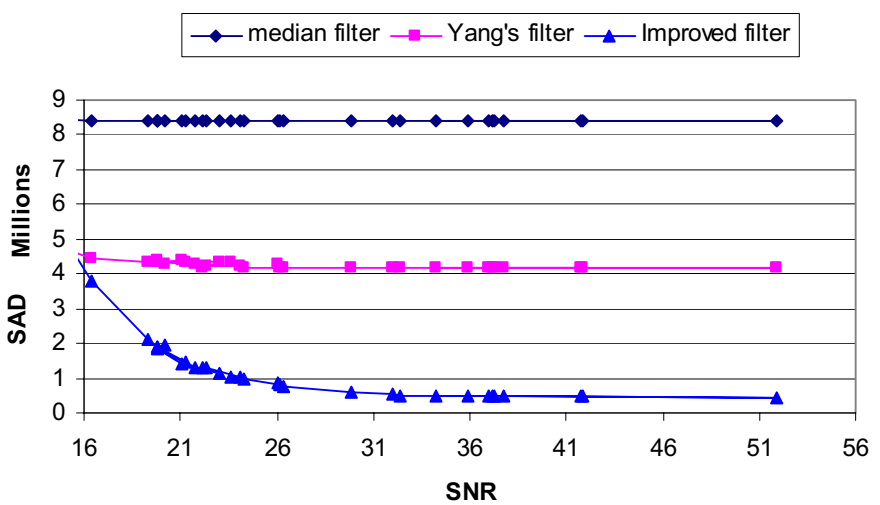


Fig. 9. Performance results of applying a conventional median filter, Yang's structure-adaptive anisotropic and the improved structure-adaptive anisotropic filters on the Bird image contaminated by a 'Salt and Pepper' noise at different SNR levels.

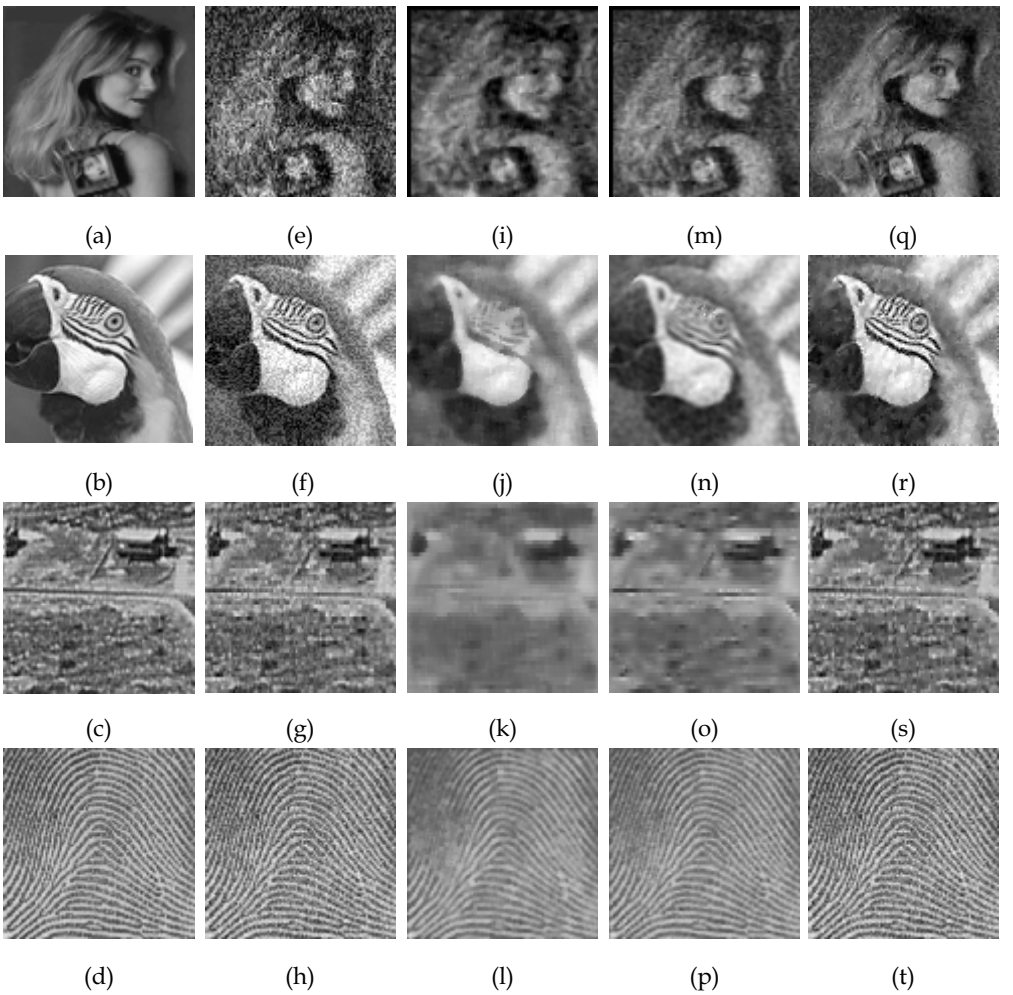


Fig. 10. Applying the different filters on images contaminated by Gaussian noise: Original images (a)-(d), with Gaussian noise (SNR:10dB) (e) and (SNR:20dB) (f)-(h), images reconstructed by the median filter(i)-(l), by the structure-adaptive anisotropic filter(m)-(p) and by the proposed improved structure-adaptive anisotropic filter (q)-(t).

4. The modified structure-adaptive anisotropic filter

In this section we propose some modifications to the structure-adaptive anisotropic filter (Yang et al., 1996) in the frequency domain and present the modified structure-adaptive anisotropic filter.

The structure-adaptive filter has the low-pass filter characteristics. We propose to convert it to the band-pass filter by multiplying its kernel by a scaling factor S and adding an offset V .

The obtained is the modified structure-adaptive anisotropic filter, which has the following general form:

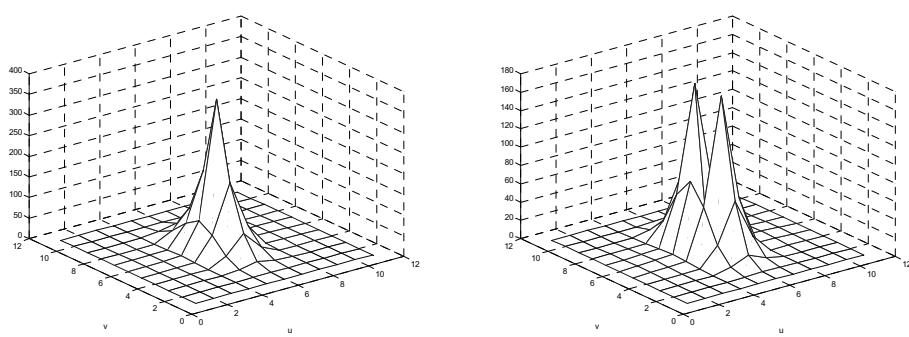
$$h(x_0,x) = V + S\cdot\rho\,(x-x_0)\exp\left\{-\left[\frac{((x-x_0)\cdot n)^2}{\sigma_1{}^2(x_0)}+\frac{((x-x_0)\cdot n_\perp)^2}{\sigma_2{}^2(x_0)}\right]\right\} \tag{12}$$

where V and S are the parameters, which must be adjusted to the specific application. Applying a 2D Fourier transform on (12) we obtain the filter's frequency response:

$$\begin{gathered} H(u,v) = V\cdot 4\pi^2\delta(u,v)+\frac{1}{\pi\beta}S\left(\frac{\sin(ur)}{u}*\exp\left\{-\frac{u^2}{4\beta}\right\}\right)\cdot\left(\frac{\sin(vr)}{v}*\exp\left\{-\frac{v^2}{4\beta}\right\}\right) \\ \beta = \left(\frac{\cos\theta}{\sigma_1(x_0)}\right)^2+\left(\frac{\sin\theta}{\sigma_2(x_0)}\right)^2 \end{gathered} \tag{13}$$

where θ is the local pattern orientation, r is the kernel's maximal support radius and $*$ is a convolution operator.

The general form of the modified structure-adaptive anisotropic filter can obtain different frequency response behavior types (low-pass and band-pass) by matching the values of V and S (Figure 11). The structure-adaptive filter (Yang et al., 1996) can be seen as a special case of the modified structure-adaptive anisotropic filter and it is obtained by setting the values to S=1 and V=0.



(a) (b)

Fig. 11. Example of different frequency behavior types of the modified structure-adaptive anisotropic filter :(a) low-pass filter type (V=1 S=10) and (b) band-pass filter type (V=-2 S=10)

Band-pass form of the modified structure-adaptive anisotropic filter is effective in filtering images in which oriented patterns in a local neighborhood form a sinusoidal-shaped plane wave with a well-defined frequency and orientation (i.e. fingerprint images including ridges and valleys).

5. The unique structure-adaptive anisotropic

In this Section the application of the modified structure-adaptive anisotropic filter to fingerprint image enhancement is made and the unique structure-adaptive anisotropic filter is proposed. Fingerprints are today the biometric features most widely used for personal identification. Fingerprint recognition is one of the basic tasks of the Integrated Automated Fingerprint Identification Service (IAFIS) of the most famous police agencies (Lee & Gaensslen, 1991). A fingerprint pattern is characterized by a set of ridgelines that often flow parallel, but intersect and terminate at some points. The uniqueness of a fingerprint is determined by the local ridge characteristics and their relationships (Hong et al., 1998), (Lee & Gaensslen, 1991). Most automatic systems for fingerprint comparison are based on minutiae matching (Hollingum, 1992). Minutiae characteristics are local discontinuities in the fingerprint pattern and represent the two most prominent local ridge characteristics: terminations and bifurcations A ridge termination is defined as the point where a ridge ends abruptly, while ridge bifurcation is defined as the point where a ridge forks or diverges into branch ridges (Figure 12). A typical fingerprint image contains about 40-100 minutiae (Hong et al., 1998).

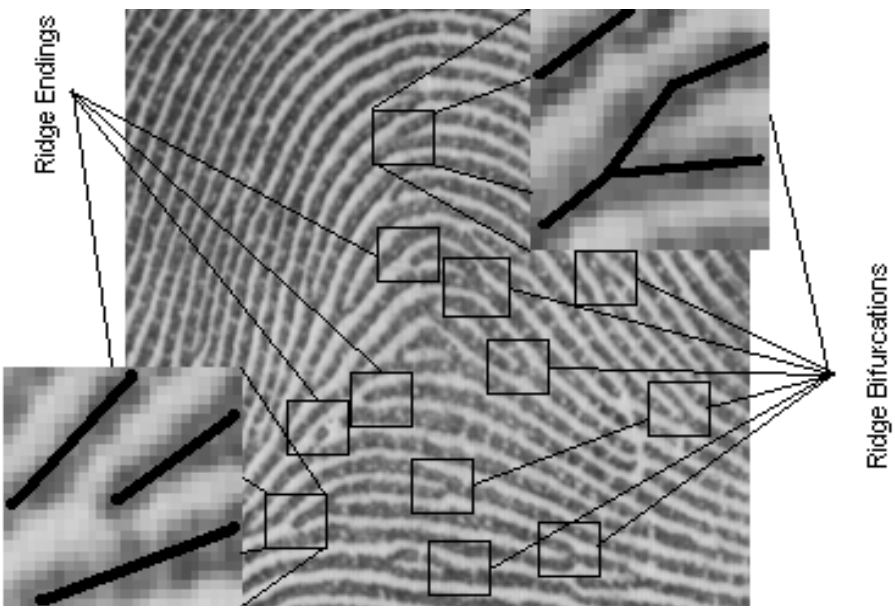


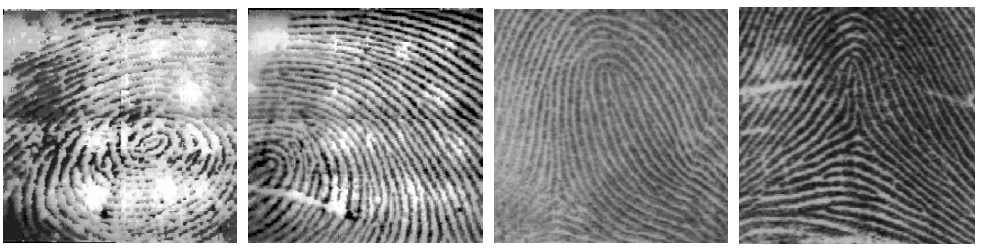
Fig. 12. Examples of minutiae (ridge ending and bifurcation) in a fingerprint image

An automatic fingerprint image matching process, which enables a personal identification, strongly depends on comparison of the Minutiae Points of Interest (MPOI) and their relationships. Reliable automatic extraction of these MPOI is a critical step in fingerprint classification.

The performance of minutiae extraction algorithm relies heavily on the quality of the fingerprint images (Hong et al., 1998). The ridge structures in poor-quality fingerprint images are not always well defined and, hence, cannot be correctly detected. This might result in the creation of spurious minutiae and the ignoring of genuine minutiae. Therefore

large errors in minutiae localization may be introduced (Hong et al., 1998). Examples of poor-quality fingerprint images are shown in Figure 13. In order to ensure robust performance of minutiae extraction algorithm an enhancement algorithm that improves the clarity of the ridge structures is necessary (Hong et al., 1998), (Hong et al., 1996).

Most of the fingerprint image enhancement techniques, proposed in the literature, are applied to binary images, while some others operate directly on gray-scale images (Lee & Gaensslen, 1991), (O'Gorman, L. & Nickerson, 1989), (Sherlock et al., 1994). The binarization process may cause loss of information about true ridge structure and it has inherent limitations (Hong et al., 1998).



(a) (b) (c) (d)

Fig. 13. Examples of poor quality fingerprint images due to: noisy acquisition device (a), (b) and variation in impression conditions (c), (d), resulting in corrupted ridgelines

Ko (2002) and Sherlock et al. (1994) suggested gray-scale image enhancement techniques, which are applied in the frequency domain, while Hong et al. (1998) and Huvanandana et al. (2003) employed their techniques directly in the space domain. Cheng et al. (2002) applies scale space theory to fingerprints enhancement by filtering the image in an iterative manner using both local and global image charackteristics. Almansa & Lindeberg (2000) propose a diffusion technique, which estimates iteratively local features and perfoms directional filtering in regions with well-defined orientation, while in areas without a dominant orientation it applies isotropic filtering. Wang & Wang (2004) propose similar technique, which applies isotropic filtering in the frequency domain at regions without clear dominant orientation.

Different gray-scale fingerprint images enhancement techniques assume that the local ridge frequency and orientation can be reliably estimated. However, this assumption is not valid for poor-quality fingerprint images. Although, other decomposition methods (Hong et al., 1998), (Hong et al., 1996) which apply a bank of Gabor filters to the input fingerprint images, can obtain reliable orientation estimation even for corrupted images, they are computationally expensive.

Hong et al. (1998) proposed a fast enhancement algorithm, which can adaptively improve the clarity of ridge and valley structures of input fingerprint images based on the estimated local ridge orientation and frequency.

We present some improvement to the Hong (Hong et al., 1998) method by using the modified structure-adaptive anisotropic filter, adapted to fingerprint images. Instead of using both local ridge orientation and local frequency information, only the orientation information is used in our approach. Our proposed unique structure-adaptive anisotropic filter, which eliminates the need to estimate local frequency information, can replace the Gabor filter. The proposed enhancement algorithm with the unique filter is faster and efficient as well.

We have adjusted the modified structure-adaptive anisotropic filter specifically to fit fingerprint images, and empirically set the filter parameters to $V = -2$ and $S = 10$. The filter frequency response has bandpass filter characteristics. The proposed filter was found to be effective in fingerprint image enhancement, while preserving the local ridge frequency of the fingerprint image (see Figure 15). The frequency bands transferred by the filter include almost all typical local ridge frequencies that lie within a certain range for a given image resolution (Hong et al., 1998).

The space constants $\sigma_1^2(x_0)$ and $\sigma_2^2(x_0)$ of the structure-adaptive anisotropic filter kernel are controlled through both the corner detector $c(x)$ and by the measurement of anisotropism $g(x)$ as defined by (5) and (6). These equations include gradient estimation via calculations of first order derivatives of the input image. Problems may arise if the input data is noisy. This is because taking derivatives is a highpass filtering process, which amplifies the effect of noise. The space constants have a strong influence on filter performance on fingerprint images, and since they are highly affected by noise, we suggest setting them to constants $\sigma_1^2(x_0) = 4$ and $\sigma_2^2(x_0) = 2$. This setting produced a filter in the form of a Gaussian-shaped kernel with a double ratio between the axis running in the main direction and the axis perpendicular to it. By setting the space constants to constant values we obtain a filter that is more robust to noise. However, the filter is optimal only for a fingerprint set, which was used in our experiments. Therefore in our future work we will develop more robust to noise local anisotropic measurements that will control the space constants. Figure 14 shows a comparison of impulse and frequency responses between the structure-adaptive anisotropic filter (Yang et al., 1996) and the unique filter ($V = -2$, $S = 10$). It can be seen that both filters have directional Gaussian-like shaped kernels in a space domain. However, they are different in the frequency domain: the Yang's anisotropic filter shows Low-Pass (one peak in the center) filter characteristics, while the unique anisotropic filter expresses Band-Pass filter characteristics (two peaks symmetrically located around the center).

The unique filter has one undesired property: its value at infinity is unequal to zero. Therefore its response depends on the chosen support size of the filter. However, the unique filter is adjusted to transfer all typical local ridge frequencies, which lie within a certain range for a given image resolution (Hong et al., 1998), and it works well on all set of fingerprint images with given resolution, as demonstrated in the next Section.

The performance of the unique structure-adaptive anisotropic filter is studied in the context of fingerprint image enhancement algorithm simulated by Hong (1998). We compare the applying of the structure-adaptive anisotropic filter J (Yang, 1996), the unique structure-adaptive anisotropic filter I, Gabor-based filter G (Hong, 1998) and the modified Gabor-based filter (Greenberg et al., 2000 & Greenberg et al., 2002) H, on the same set of fingerprint images. We use performance results of the structure-adaptive anisotropic filter I and Gabor filters G, H taken from a different comparative study (Greenberg et al., 2000 & Greenberg et al., 2002) conducted on direct gray-scale fingerprint enhancement methods. In this work we extend this comparative study with performance results of the unique structure-adaptive anisotropic filter J applied on the same fingerprint images set as in (Greenberg et al., 2000 & Greenberg et al., 2002). The sample set is composed of 10 fingerprints taken from NIST, FBI sample and using an optoelectronic device. The gradient used by compared filtering algorithms was obtained using first order approximation.

Figure 15 shows a comparison of the enhancement results obtained using different filters, for poor-quality fingerprint images, which contain regions that do not form a well-defined local ridge frequency. These regions are mostly encountered in the neighbourhood of fingerprint image singular points: core and delta (Lee & Gaensslen, 1991). Both the structure-adaptive and the unique structure-adaptive anisotropic filters outperform the Gabor-based filters for those regions, which contain singular points.

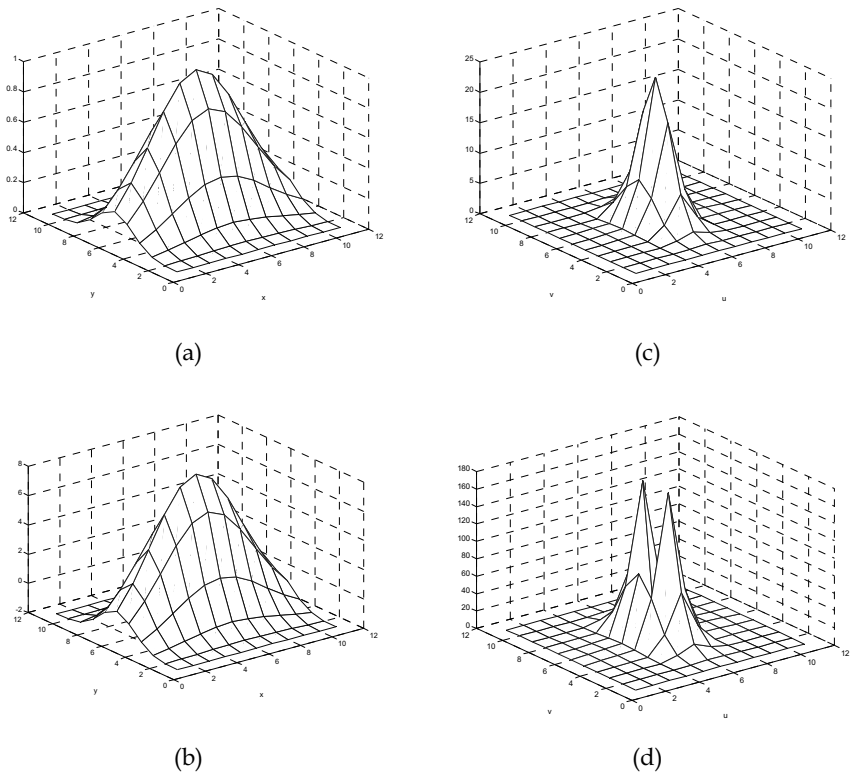


Fig. 14. Comparison of impulse and frequency response between the structure-adaptive anisotropic filter and the proposed unique structure-adaptive anisotropic filter ($V = -2$, $S = 10$). Both filters are 11x11 pixels kernel size, and have a directional Gaussian-like shaped kernel in a space domain (a) and (b). However they are different in the frequency domain: (c) the structure-adaptive anisotropic filter shows lowpass filter characteristics, while (d) the unique filter expresses bandpass filter characteristics (two peaks around the center).

(a) (b) (c) (d) (e)

Fig. 15. Example of enhancement results of fingerprint image region with a singular point (core). Original image (a); enhanced image after applying the Gabor-based filter (b), the modified Gabor-based filter(c), the structure-adaptive anisotropic filter (d) and the proposed unique structure-adaptive anisotropic filter (e)

Figure 16 compares the success rate obtained by applying gray-scale filtering technique (Greenberg et al., 2000 & Greenberg et al., 2002) using the four filters (G, H, I, J) on the same fingerprint image set. The average error percentage is expressed in terms of false (minutiae that was found in the region not containing true minutiae), dropped (minutiae that was not found in the neighborhood of true minutiae) and exchanged minutiae (minutiae differing from the true minutiae type in the same image region). The average error percentage presented by our approach I (unique structure-adaptive anisotropic filter), is comparable to the errors produced by approach H (modified Gabor-based filter). Both I filter and H filter create less errors than the G (Gabor-based) and I (structure-adaptive anisotropic) filters.

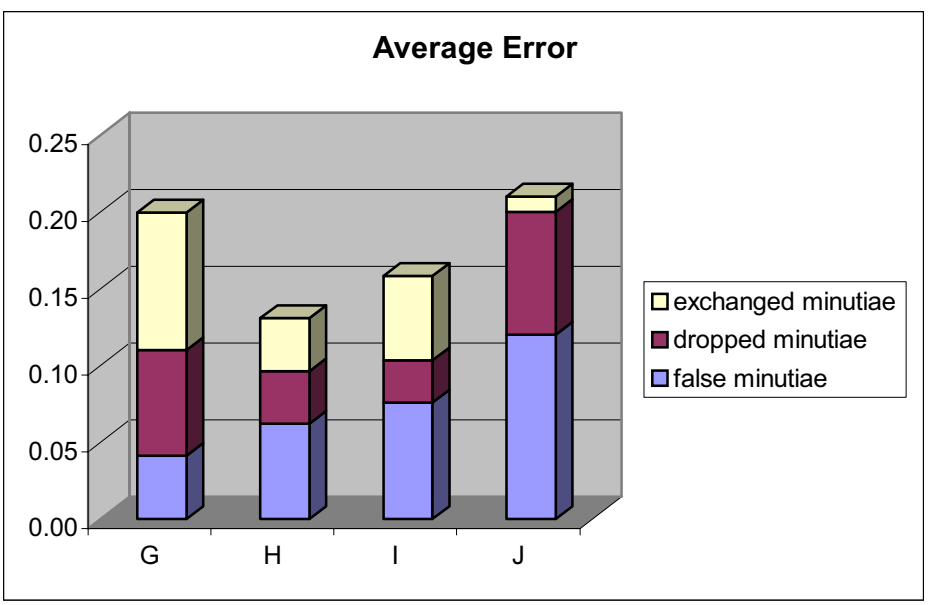


Fig. 16. Comparison of the filters performance: G-Gabor-based filter, H-modified Gabor–based filter, I-unique proposed structure-adaptive anisotropic filter and J-structure-adaptive anisotropic filter.

Figure 17 demonstrates the enhancement results of applying the structure-adaptive anisotropic (Yang, 1996), the unique structure-adaptive anisotropic and the modified Gabor-based (Greenberg et al., 2000 & Greenberg et al., 2002) filters to some fingerprint images from the sample set.

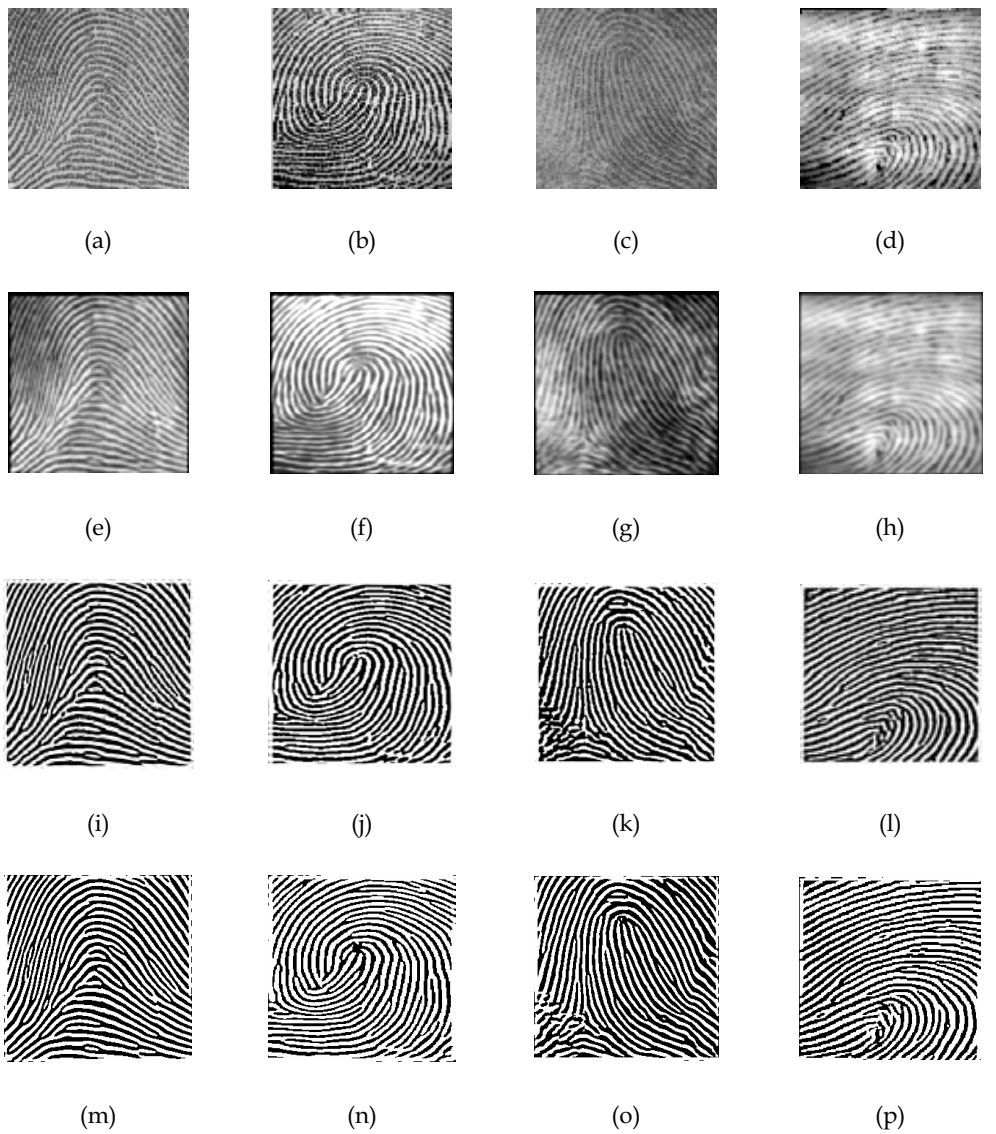


Fig. 17. Enhancement results of applying different filters to fingerprint images from the same sample set: (a)-(d) original fingerprint images and after enhancement using (e)-(h) the structure-adaptive anisotropic filter, (i)-(l) the proposed unique structure-adaptive anisotropic filter and (m)-(p) the improved Gabor-based filter, accordingly.

Table 1 shows the wall time for different stages of the Gabor-based enhancement algorithm simulated by Hong (1998) and the total time on a Pentium 200MHz PC. The enhancement algorithm based on the anisotropic filter does not require the estimation of the local ridge frequency information. Therefore it saves about 4% of the processing effort compared to the Gabor-based enhancement algorithm.

Normalization (Seconds)	Orientation (Seconds)	Frequency (Seconds)	Region Mask (Seconds)	Filtering (Seconds)	Total (Seconds)
0.11	0.14	0.09	0.07	2.08	2.49

Table 1. The wall time of the Gabor-based enhancement algorithm on a Pentium 200MHZ PC (taken from (Hong, 1998), Table 2)

6. References

Mastin, G.A. (1985). Adaptive filters for digital image noise smoothing: an evaluation, *Computer Vision, Graphics and Image Processing*, No. 31, 103-121.

Donahue, M.J. & Rokhlin, I. (1993). On the Use of Level Curves in Image Analysis, *Image Understanding*, No. 57, 185-203.

Yang, G.Z., Burger, P., Firmin, D.N. & Underwood, S.R. (1996). Structure adaptive anisotropic filtering, *Image and Vision Computing*, No. 14, 135-145.

Zucker, A., Lev, S.W. & Rosenfeld, A. (1977). Iterative enhancement of noisy images, *IEEE Trans. Systems, Man and Cybernetics*, No. 7, 435-441.

Wang, D.C., Vagucci, A.H. & Li, C.C. (1981). Gradient inverse weighted smoothing scheme and the evaluation of its performance, *Computer Graphics and Image Processing*, No. 15, 167-181.

Davis, L.S. & Rosenfeld, A. (1978). Noise cleaning by iterated local averaging, *IEEE Trans. Systems, Man and Cybernetics*, No. 8, 705-710.

Blake, A. & Zisserman, A. (1987). In: *Visual Reconstruction*, MIT Press, Cambridge, MA.

Perona, P. & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, 629-639.

Smolka, B., Plataniotis, K.N., Lukac, R. & Venetsanopoulos, A.N. (2003). On the Forward and Backward Anisotropic Diffusion Framework, *Proceedings of the 2003 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, (NSIP-03) in Grado, Italy.

Almansa, A. & Lindeberg, T. (2000). Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection, *IEEE Transactions on Image Processing*, Vol. 9, 2027-42.

Weickert, J. (2001). Applications of nonlinear diffusion in image processing and computer vision, *Acta Mathematica Universitatis Comenianae*, Vol. 70, 33-50.

Freeman, W.T. & Adelson, E.H. (1991). The design and use of steerable filters, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 13, 891-906.

Perona, P. (1992). Steerable-scalable kernels for edge detection and junction analysis, *Image and Vision Computing*, Vol. 10, 663-672.

Yu, W., Daniilidis, K. & Sommer, G. (2001). Approximate Orientation Steerability Based on Angular Gaussians, *IEEE Trans. on Image* Processing, Vol. 10, 193-205.

Yang, G.Z., Burger, P., Firmin D.N. & Underwood, S.R. (1996). Structure adaptive anisotropic filtering, *Image and Vision Computing*, Vol. 14, 135-145.

Hong, L., Jain, A.K., Pankanti, S. & Bolle, R. (1996). Fingerprint Enhancement, *Proc. First IEEE WACV*, Sarasota, Fla., 292-207.

Hong, L., Wan, Y. & Jain, A. (1998). Fingerprint Image Enhancement: Algorithm and Performance Evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, 777-789.

Donahue, M.J. & Rokhlin, S.I. (1993). On the Use of Level Curves in Image Analysis, *Image Understanding*, Vol. 57, 185-203.

Lee, H.C. & Gaensslen, R.E. (1991). *Advanced in Fingerprint Technology*, Ny: Elsevier.

Hollingum, J. (1992). Automated Fingerprint Analysis Offers Fast Verification, *Sensor Review*, Vol. 12, 12-15.

Hong, L., Jain, A.K., Pankanti, S. & Bolle, R. (1996). Fingerprint Enhancement, *Proc. First IEEE WACV*, Sarasota, Fla., 292-207.

O'Gorman, L. & Nickerson, J.V. (1989). An Approach to Fingerprint Filter Design, *Pattern Recognition*, Vol. 22 ,29-38.

Sherlock, D., Monro, D.M. & Millard, K. (1994). Fingerprint Enhancement by Directional Fourier Filtering, *IEE Proc. Visual Imaging Signal Processing*, Vol. 141, 87-94.

Ko, T. (2002). Fingerprint enhancement by spectral analysis techniques, *Proc. 31st Applied Imagery Pattern Recognition Workshop*, pp.133-139.

Huvanandana, S., Malisuwan, S., Santiyanon, J. & Hwang, J.N. (2003). A hybrid system for automatic fingerprint identification, *Proc. 2003 International Symposium on Circuits and Systems (ISCAS'03)*, Vol. 2, 952-955.

Cheng, H., Tian, J. & Zhang, T. (2002). Fingerprint enhancement with dyadic scale-space, *16th Int. Conf. on Pattern Recognition*, Vol. 1, 200–203.

Almansa, A. & Lindeberg, T. (2000). Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection, *IEEE Transactions on Image Processing*, Vol. 9, 2027-42.

Wang, S. & Wang, (2004). Fingerprint enhancement in the singular point area, *IEEE Signal Processing Letters*, Vol. 11, 16-19.

Greenberg, S., Aladjem, M., Kogan, D. & Dimitrov, I. (2000). Fingerprint image enhancement using filtering techniques, *15th Int. Conf. on Pattern Recognition*, Barcelona 3, 326-329.

Greenberg, S., Aladjem, M. & Kogan, D. (2002). Fingerprint image enhancement using filtering techniques, *Real-Time Imaging*, Vol. 8, 227-236.

Lee, H.C. & Gaensslen, R.E. (1991). *Advanced in Fingerprint Technology*, Ny: Elsevier.

27

Real-Time Pattern Recognition with Adaptive Correlation Filters

Vitaly Kober, Victor H. Diaz-Ramirez[1], J. Angel Gonzalez-Fraga[2] and Josue Alvarez-Borrego[3]
[1]*Computer Science Department, CICESE,* [2]*Faculty of Sciences, UABC of Ensenada,*
[3]*Optics Department, CICESE,* [3]*Faculty of Ingineering, UABC of Ensenada*
Mexico

1. Introduction

Since the introduction of the matched spatial filter (MSF) (VanderLugt, 1964), many different types of filters for pattern recognition based on correlation have been proposed. One of the reasons of such growing interest to design effective methods of pattern recognition stems from the need to deal with more complex images in various applications of automated image processing and from the need to process large images in real time. For some of the more critical applications, optical or hybrid optodigital techniques allow faster processing of images. This is why our approach in this area is based on correlation filters, which possess good mathematical fundamentals and can be effectively implemented digitally or optodigitally (Moreno et al., 1998).

In pattern recognition two essentially different types of tasks are distinguished: detection of a target and estimation of its exact position. When correlation filters are used, these problems can be solved in two steps. First, the detection is carried out by searching correlation peaks in the filter output, and then coordinates of these peaks are taken as position estimations. The quality of both procedures is limited by the presence of noise in an observed scene. The detection capabilities of correlation filters can be quantitatively expressed in terms of probability of detection errors (false alarms), signal-to-noise ratio, discrimination capability, peak-to-output energy ratio, etc. (Vijaya Kumar & Hassebrook, 1990). Some of the measures can be essentially improved using an adaptive approach to the filter design. According to this concept, we are interested in a filter with good performance characteristics for a given observed scene, i.e., with a fixed set of patterns or a fixed background to be rejected, rather than in a filter with average performance parameters over an ensemble of images. After the detection task has been solved we still faced with small errors of target position estimations due to distortion of the object by noise. The coordinate estimations lie in the vicinity of their actual values. So the target location can be characterized only by means of the variance of measurement errors along coordinates (Kober & Campos, 1996).

One of the most important performance criteria in pattern recognition is the discrimination capability (DC), or how well a filter detects and discriminates different classes of objects. A correlation filter with a minimum probability of anomalous detection errors (false alarms)

referred to as the optimal filter (OF) was suggested (Yaroslavsky, 1993). An important feature of the OF is its scene-adaptivity in applications to pattern recognition or target detection because its frequency response takes into account the power spectrum of wrong objects in the observed scene or the background to be rejected. The disadvantage of the OF in optical implementation is its extremely low light efficiency. A filter with maximum light efficiency is the phase-only filter (POF) (Horner & Gianino, 1984). The drawback of the POF is its poor discrimination capability for a low-contrast target embedded into a complicated background scene. An approximation of the OF by means of phase-only filters with a quantization was made (Kober et al., 1994). There, the approximate filters with high light efficiency and discrimination capability close to that of the OF were suggested. When the object to be recognized is in the presence of disjoint background noise, the design of the optimal filter was also obtained (Javidi & Wang, 1994).

It is commonly known that the MSF is very sensitive to small distortions of the object caused by variations in scale, rotation, or point of view. One of the first attempts to overcome the problem of distortion in pattern recognition was the introduction of synthetic discriminant functions (SDFs), (Hester & Casasent, 1980; Casasent, 1984). The SDF filters use a set of training images to synthesize a template that yields a prespecified central correlation output in the response to training images. The main shortcoming of the SDF filters is appearance of sidelobes owing to the lack of control over the whole correlation plane. As a result, the SDF filters often possess a low discrimination capability. A partial solution of this problem was suggested (Mahalanobis et al., 1987). They proposed to control over the whole correlation plane by producing sharp correlation peaks for easy detection of the target as well as by minimizing the average correlation energy to suppress the presence of extraneous correlation peaks. However, these filters are not tolerant to input noise. They perform control over false alarms by an indirect way, and, finally, they are more sensitive to interclass variations than other composite filters (Billet & Singher, 2002).

This chapter treats the problems of real-time pattern recognition exploiting adaptive distortion-invariant correlation filters (González-Fraga et al., 2006; Diaz-Ramirez et al., 2006; Kober et al., 2006). The distinctive feature of the proposed methods is the use of an adaptive approach to the filters design. Specifically, we shall look at two problems: detection of known objects possessing small geometric distortions and corrupted with additive sensor's noise, and implementation of the designed filters in an optodigital setup.

The first problem is to decide on presence or absence of a distorted object. New adaptive composite filters for reliable recognition of the object in a cluttered background are presented. The information about an object to be recognized, false objects, and a known background to be rejected is utilized in iterative training procedure to design a correlation filter with a given value of discrimination capability. The synthesis of the adaptive filters also takes into account additive sensor's noise by training with a noise realization. Therefore, the filters may possess a good robustness to the noise.

The second problem concerns real-time implementation of the adaptive correlation filters. For some of the more critical applications, optical or hybrid optodigital techniques allow faster processing of images. The advantage of optical systems over computers lies in inherent ability of optical systems to process data in a parallel way. For instance the classical optical correlator allows to perform fully parallel matched filtering over an input scene containing multiple patterns. Recent progress in optical spatial light modulators gives new opportunities for creation of optodigital systems. Such modulators can be addressed

electronically that allows rapidly and flexibly change the object or the filter for real-time applications. We implemented the adaptive filters in a hybrid system using the joint transform correlator scheme. The hybrid system additionally takes into account real characteristics of used optoelectronics devices. Computer simulation and experimental results are provided and disscussed.

2. Adaptive Digital Systems

2.1. Conventional correlation filters

Consider the problem of detecting the presence and location of a known distorted target in an observed scene using the correlation operation. The correlation can be effectively implemented in a computer with the help of the fast Fourier transform. When the correlation output is obtained then coordinates of the correlation peaks can be taken as position estimations of a target.

A basic correlation filter is the MSF whose impulse response is the flipped version of a reference object. This filter is optimal with respect to the signal-to-noise ratio at the filter output when an input signal is on presence of additive white noise. A drawback of the MSF in optical implementation is its low light efficiency. A filter with maximum light efficiency is the POF. The transfer function of a basic POF (Horner & Gianino, 1984) is given by

$$H_{POF}(u,v) = \frac{T^*(u,v)}{|T(u,v)|} = \begin{cases} \exp(-i\Phi_t(u,v)), & if\ |T(u,v)| \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $T(u,v)$, $\Phi_t(u,v)$ are the Fourier transform and the phase distribution of the target, respectively. The asterisk denotes complex conjugate.

The transfer function the OF can be approximated in the Fourier domain as

$$H_{OF}(u,v) = \frac{T^*(u,v)}{|T(u,v)|^2 + |S(u,v)|^2}, \tag{2}$$

where $S(u,v)$ is the Fourier transform of the input scene (Yaroslavsky, 1993).

The performance of conventional correlation filters degrades rapidly with image distortions. An attractive approach to distortion-invariant pattern recognition is based on SDF filters. These filters (called composite filters) use a set of training images (patterns), which are sufficiently descriptive and representative for expected distortions. A basic SDF filter is a linear combination of MSFs for different patterns (Casasent, 1984). The coefficients of the linear combination are chosen to satisfy a set of constraints on the filter output. Two different recognition problems can be solved with the composite filters.

Intraclass Recognition Problem

Let $\{t_i(x,y);\ i=1,2,\ldots,N\}$ be a set of (linearly independent) training images each with d pixels. The SDF filter function $h(x,y)$ in the spatial domain can be expressed as a linear combination of a set of reference images, i.e.,

$$h(x,y) = \sum_{i=1}^{N} a_i t_i(x,y), \tag{3}$$

where $\{a_i;\ i=1,2,\ldots,N\}$ are weighting coefficients, and they are chosen to satisfy the following conditions:

$$t_i \otimes h = q_i \ . \tag{4}$$

Here the symbol $\otimes$ denotes the correlation, and $\{q_i;\ i=1,2,\ldots,N\}$ are prespecified values in the correlation output at the origin for each training image.
Let **R** denote a matrix with N columns and d rows (number of pixels in each training image), where its ith column is given by the vector version of $t_i(x,y)$. Let **a** and **u** represent column vectors of $\{a_i\}$ and $\{q_i\}$, respectively. We can rewrite Eqs. (3) and (4) in matrix-vector notation as follows:

$$\mathbf{h} = \mathbf{R}\mathbf{a} \ , \tag{5}$$

$$\mathbf{q} = \mathbf{R}^{+}\mathbf{h} \ , \tag{6}$$

where superscript $^+$ means conjugate transpose.
By substituting Eq. (5) into Eq. (6) we obtain

$$\mathbf{q} = (\mathbf{R}^{+}\mathbf{R})\mathbf{a} \ . \tag{7}$$

The (i,j)th element of the matrix $\mathbf{Q}=(\mathbf{R}^{+}\mathbf{R})$ is the value at the origin of the cross-correlation between the training images $t_i(x,y)$ and $t_j(x,y)$. If the matrix **Q** is nonsingular, the solution of the equation system is given by

$$\mathbf{a} = (\mathbf{R}^{+}\mathbf{R})^{-1}\mathbf{q} \ , \tag{8}$$

and the filter vector is

$$\mathbf{h}_{SDF} = \mathbf{R}(\mathbf{R}^{+}\mathbf{R})^{-1}\mathbf{q} \ . \tag{9}$$

The SDF filters with equal output correlation peaks can be used for intraclass distortion-invariant pattern recognition, i.e., detection of distorted patterns belonging to the true-class of objects. This can be done by setting all elements of **q** to unity, i.e.,

$$\mathbf{q} = [1\ 1\ \ldots\ 1]^T \ . \tag{10}$$

Multiclass Recognition Problem
Assume that there are distorted versions of a reference object and various classes of objects to be rejected. For simplicity, we consider two-class recognition problem. Thus, we design a correlation filter to recognize training images from one class (called true class) and to reject training images from another class (called false class). Suppose that there are M training images from the false class $\{p_i(x,y);\ i=1,2,\ldots,M\}$. According to the SDF approach, the composite image $h(x,y)$ is a linear combination of all training images $\{t_1(x,y),\ldots, t_N(x,y), p_1(x,y),\ldots, p_M(x,y)\}$. The both intraclass recognition and interclass discrimination problems can be solved by means of SDF filters. We can set the filter output $\{q_i=1;\ i=1,2,\ldots,N\}$ for the true class objects and $\{q_i=0,\ i=N+1,N+2,\ldots,N+M\}$ for the false class objects, i.e.,

$$\mathbf{q} = [1\ 1\ \ldots\ 1\ 0\ 0\ \ldots\ 0]^T \ . \tag{11}$$

Using the filter given in Eq. (9), we expect that the central correlation peak will be close to unity for the true class objects and it will be close to zero for the false class objects. Obviously, the preceding approach can be easily extended to any number of classes to be discriminated. Note that this simple procedure is the lack of control over the full correlation output because we are able to control only the correlation output at the location of cross-correlation peaks. Therefore, other sidelobes (false peaks) may appear everywhere on the correlation plane. To reduce the sidelobes, a composite correlation filter (called MACE filter) with a sharp correlation peak at the output was proposed (Mahalanobis et al., 1987). The MACE filter is synthesized in the frequency domain as follows:

$$\mathbf{H}_{MACE} = \mathbf{D}^{-1}\mathbf{P}(\mathbf{P}^{+}\mathbf{D}^{-1}\mathbf{P})^{-1}\mathbf{q}\,, \tag{12}$$

where **D** is a diagonal matrix, **P** is a matrix with N columns and d rows, where its ith column is given by the vector version of $T_i(u,v)$ (Fourier transform of $t_i(x,y)$). The entries along the diagonal are obtained by averaging the power spectrum of each image $(|T_i(u,v)|^2;$ $i=1,2,\ldots,N)$ and then scanning the average from left to right, and from top to bottom.

2.2. Design of Adaptive Correlation Filters

To achieve good recognition of the target it is necessary to reduce correlation function levels at all false peaks except at the origin of the correlation plane, where the constraint on the peak value must be met. For a given object to be recognized, false objects, and a background to be rejected, it can be done with the help of an iterative algorithm. At each iteration, the algorithm suppresses the highest sidelobe peak and therefore monotonically increases the value of discrimination capability until a prespecified value will be reached. The discrimination capability is formally defined as ability of a filter to distinguish a target among other different objects. If a target is embedded into a background that contains false objects, then the DC can be expressed as follows:

$$DC = 1 - \frac{\left|C^{B}(0,0)\right|^{2}}{\left|C^{T}(0,0)\right|^{2}}\,, \tag{13}$$

where C^B is the maximum in the correlation plane over the background area to be rejected, and C^T is the maximum in the correlation plane over the area of target position. The area of target position is determined in the close vicinity of the actual target location. The background area is complementary to the area of target position. Negative values of the DC indicate that a tested filter fails to recognize the target.

We are interested in a correlation filter that identifies a target with a high discrimination capability in cluttered and noisy input scenes. Actually in this case, conventional correlation filters yield a poor performance (Javidi & Wang, 1992). With the help of adaptive composite filters, a given value of the DC can be achieved. The algorithm of the filter design requires knowledge of the background image. Thus, we are looking for the target with unknown location in the known input scene background. The background can be described either stochastically, for instance, it can be considered as a realization of a stochastic process, or deterministically, which can be a picture. The background can also contain false objects with unknown locations. The first step is to carry out correlation between the background and a basic SDF filter, which is initially trained only with the target. Next, the maximum of the

filter output is set as the origin, and around the origin we form a new object to be rejected from the background. This object has the region of support equals to that of the target. The created object is added to the false class of objects. Now, two-class recognition problem described in Section 2.1 is utilized to design a new SDF filter; that is, the true class contains only the target and the false class consists of the false class objects. The described iterative procedure is repeated till a given value of the DC is obtained. Finally, note that if other objects to be rejected are known, they can be directly included into the false class and used for the design of adaptive SDF (ASDF) filter. A block-diagram of the procedure is shown in Fig. 1.

The proposed algorithm consists of the following steps:

1. Design ASDF filter as a conventional SDF filter trained only with the target.
2. Carry out correlation between the background and the ASDF filter.
3. Calculate the DC using Eg. (13).
4. If the value of the DC is greater or equal to the desired value, then the filter design procedure is finished, else go to the next step.
5. Create a new object to be rejected from the background. The origin of the object is at the highest sidelobe position in the correlation plane. The object is included into the false class of objects.
6. Design a new ASDF filter utilizing two-class recognition problem. The true class contains only the target and the false class consists of the false class objects. Go to step 2.

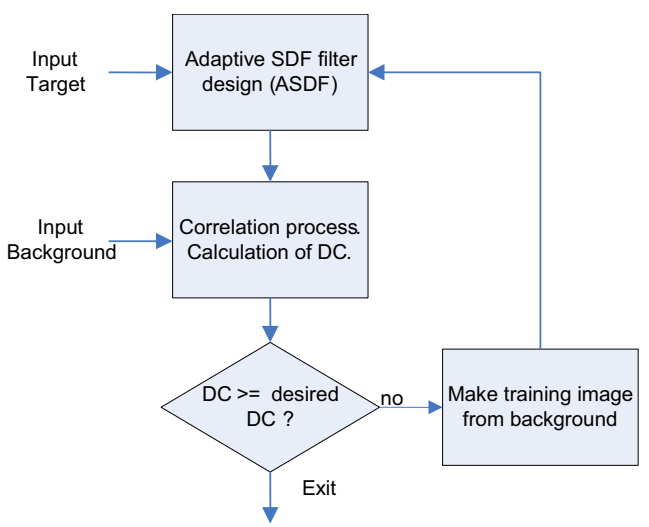


Fig. 1. Block-diagram of the iterative algorithm to design the adaptive SDF filter.

At each iteration, the algorithm chooses among all sidelobes such a peak to be suppressed in next step to ensure a monotonically increasing behavior of the DC versus the iteration index during the filter design. As a result of the procedure, the adaptive composite filter is synthesized. The performance of the filter in recognition process is expected to be close to that of in the synthesis process. Extensive computer simulations showed that for

complicated input scenes with real and stochastic cluttered backgrounds the number of iterations needed to achieve the value of the DC higher than 0.9 is about 10.

2.3. Computer Simulations

In this section, computer simulation results obtained with adaptive SDF filters are presented. The results are compared with those of the POF, the OF, and the MACE filters. The target is the airplane shown in Fig. 2(a).

(a) (b)

Fig. 2. Test images: (a) target, (b) real background.

The size of all images used in the experiments is 256×256 pixels. The signal range is 0 to 255. The mean value and the standard deviation over the target area are 130 and 42, respectively. The size of the target is about 69×26 pixels. In the first experiment, we use a real spatially inhomogeneous background shown in Fig. 2(b). The mean value and the standard deviation of the background are 104 and 40, respectively.

Figure 3 shows the performance of the adaptive filter in the filter design process in terms of the DC versus the iteration index.

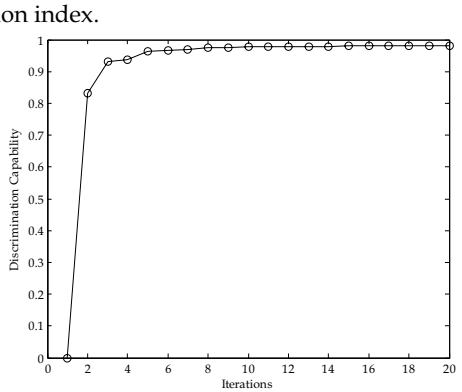


Fig. 3. Performance of the adaptive SDF filter in the design process.

After the first iteration the value of the DC is negative. After 20 iterations, the obtained ASDF filter yields DC=0.982. This means that a high level of control over the correlation plane for an input scene constructed from the background and the target can be achieved. Next, we test the recognition performance with various correlation filters when the target is imbedded into the background at arbitrary coordinates. We carried out 30 statistical trails of

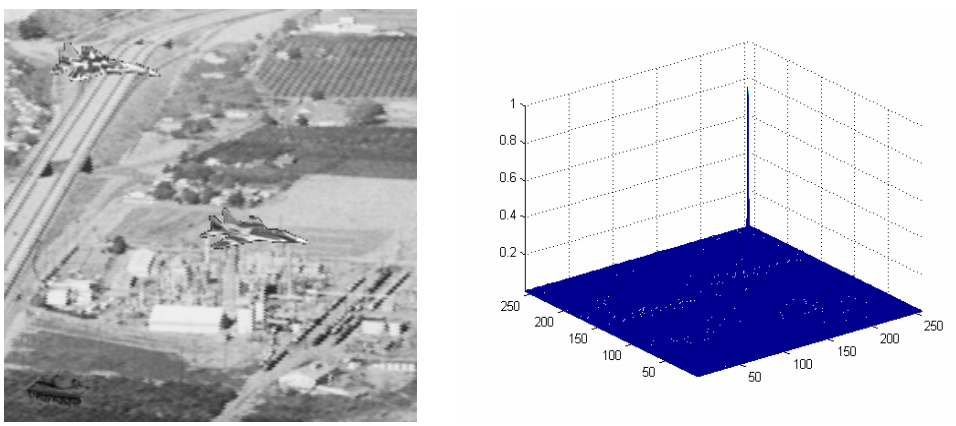
the experiment for different positions of the target. With 95% confidence the performance of the ASDF, the POF, and the OF with respect to the DC are given in line 1 of Table 1.

	POF	OF	ASDF
Scene without false target	0.35±0.22	0.66±0.10	0.95±0.01
Scene with false target	0.27±0.22	0.60±0.12	0.95±0.01

Table 1. Performance of correlation filters in terms of DC.

It can be seen that the proposed adaptive filter yields the best performance in terms of discrimination capability.

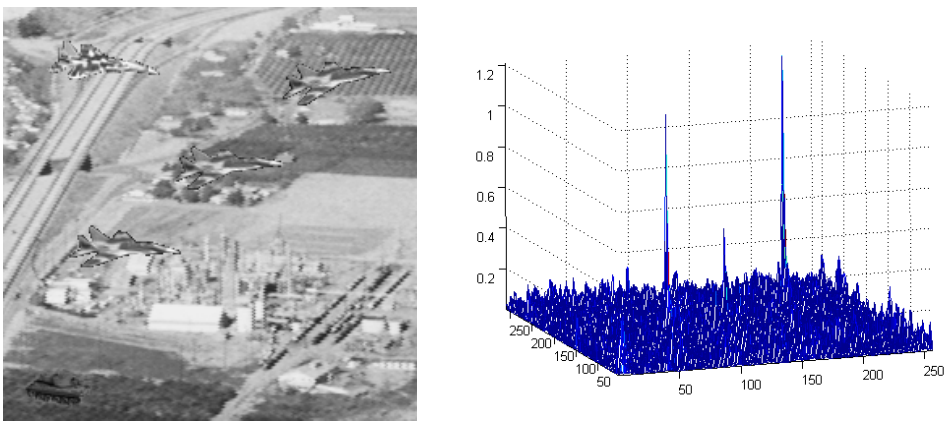
Next, we place a false object into the input scene, as it is shown in Fig. 4 (a). The performance of the correlation filters are given in line 2 of Table 1. One can observe that the ASDF filter yields the best performance with respect to the DC. Figure 4(b) shows the intensity distribution of the correlation plane obtained with the ASDF filter.



(a) (b)

Fig. 4. (a) Test scene, (b) correlation intensity plane obtained with the ASDF filter.

Now we investigate tolerance of the correlation filters to small geometric image distortions. Several methods have been proposed to improve pattern recognition in the presence of such distortions. These methods can be broadly classified into two groups. The first class concerns formally with 2-D scaling and rotation distortions. Such methods include space-variant transforms and circular harmonic functions (Arsenault & Hsu, 1983). The second class of filters uses training images that are sufficiently descriptive and representative of the expected distortions. The proposed method is based on the second approach. In our experiments, geometric distortion by means of rotation is investigated. Distorted versions of the target shown in Fig. 2(a) are used. The step and the range of object rotation are 1 deg and [0, 30], respectively. The ASDF filter is designed with seven versions of the object rotated by 0, 5, 10, 15, 20, and 25 degrees and the background scene shown in Fig. 2(b). After 30 iterations, the obtained ASDF filter yields DC=0.92. The test scene with three targets rotated by 4, 14, and 20 degrees is shown in Fig. 5(a). Figure 5(b) shows the intensity distribution of the correlation plane obtained with the ASDF filter.



(a) (b)

Fig. 5. (a) Test scene, (b) correlation intensity plane obtained with the ASDF filter.

The performance of the ASDF and MACE filters is given in Figs. 6 and 7, respectively. The MACE filter was synthesized with the same objects as the ASDF filter. Note that the conventional SDF filter fails to detect the rotated target in the cluttered background.

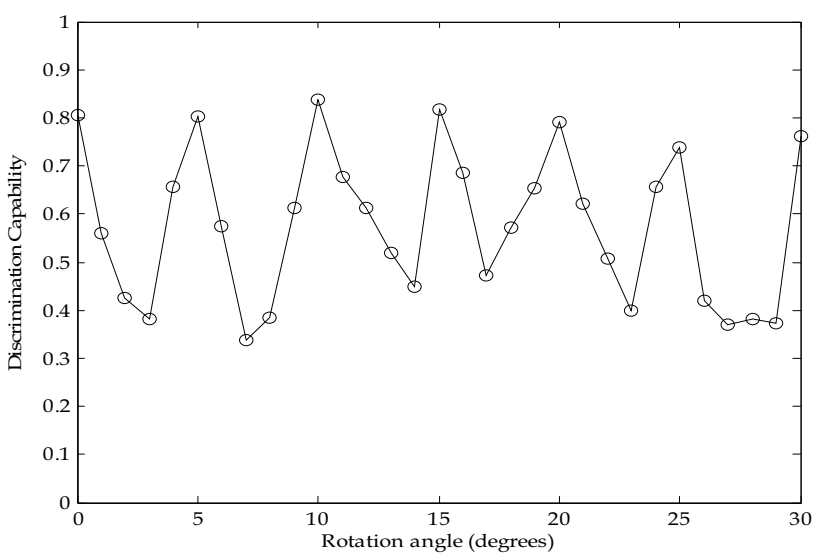


Fig. 6. Tolerance of the ASDF filter to rotation.

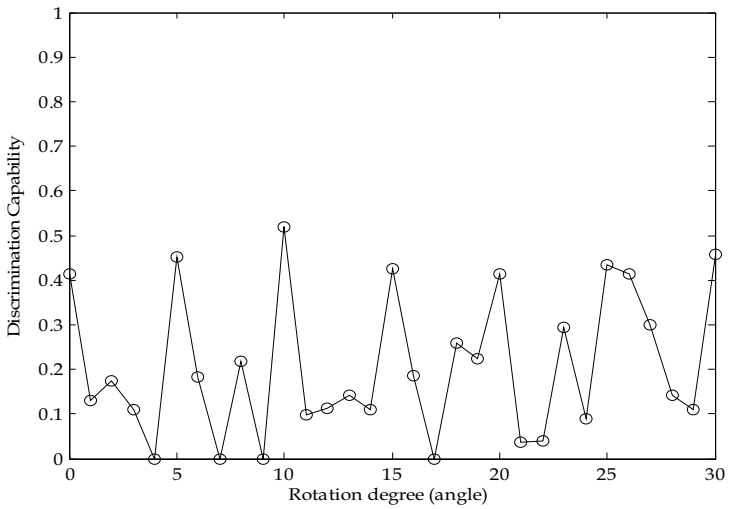
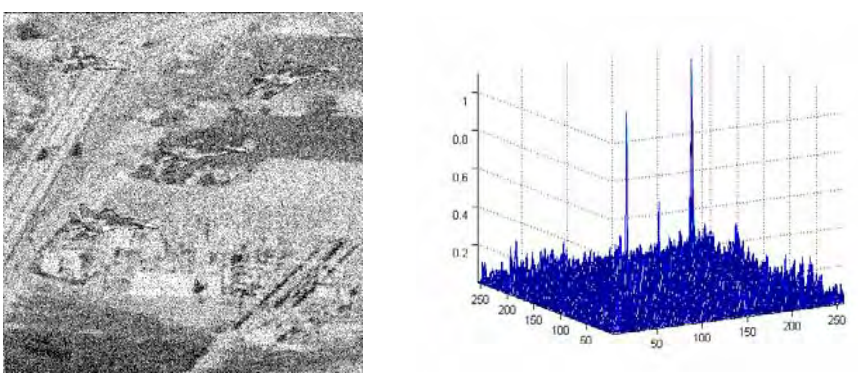


Fig. 7. Tolerance of the MACE filter to rotation.

We can see that the proposed filter possesses much better tolerance to rotation than the MACE filter. The ASDF filter adapts well by training to rotations of the target. Obviously, the preceding approach can be easily extended to any small geometric distortion of a target.

Finally we test robustness of correlation filters to additive sensor's noise that is always present in input scenes. The test scene shown in Fig. 5 (a) is used. The scene is corrupted by additive zero-mean white Gaussian noise while the standard deviation of additive noise is varied. Figure 8(a) shows the input scene corrupted by additive zero-mean white Gaussian noise with the standard deviation of 40. Figure 8(b) shows the intensity distribution of the correlation plane obtained with the ASDF filter.



(a) (b)

Fig. 8. (a) Input scene corrupted by zero-mean additive white noise with a standard deviation of 40, (b) correlation intensity plane obtained with the ASDF filter.

The tolerance of correlation filters to additive noise in terms of the DC is presented in Fig. 9.

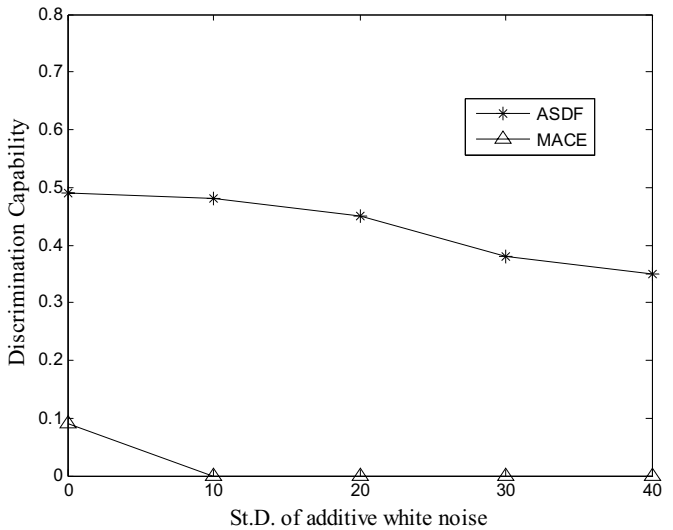


Fig. 9. Tolerance of correlation filters to additive white noise.

Since the synthesis of the ASDF filter takes into account additive noise by training with a noise realization, the filter provides a good robustness to the noise. In contrast, the performance of the MACE filter deteriorates quickly when signal noise fluctuation increases.

3. Adaptive Hybrid Optodigital Systems

Real-time pattern recognition systems based on correlation were vastly investigated in the last decades. This is because correlation filters can be implemented optically or by using hybrid (optodigital) systems exploiting the parallelism inherent in optical systems. These systems are able to carry out the recognition process at a high rate. Hybrid systems with the use of liquid crystal displays (LCDs) as spatial light modulators (SLMs) are flexible. Optodigital systems for real-time pattern recognition can be implemented on the basis of two principal architectures: 4f correlator (4FC) (VanderLugt, 1964) and joint transform correlator (JTC) (Weaver & Goodman, 1966). The advantage of the JTC compared to the 4FC is that the former is less sensitive to misalignments of an optical setup such as scale, horizontal, vertical, and azimuthal differences between the input and frequency planes. The SDF filters for distortion invariant pattern recognition were originally introduced on the basis of the 4FC. Many efforts were made to find an effective implementation of SDF filters with the JTC. In this chapter we describe an iterative algorithm to design adaptive correlation filters for the JTC architecture. The proposed algorithm takes into account calibration lookup tables of all optoelectronic devices used in real experiments.

3.1. Joint Transform Correlators

The JTC introduced in 1966 by Weaver and Goodman is shown in Fig. 10.

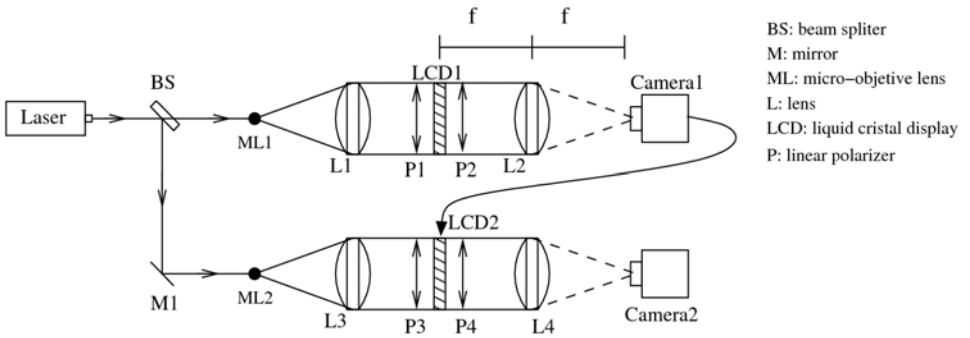


Fig. 10. Block diagram of the classical JTC.

The input plane (joint image) $f(x,y)$ is composed by the scene image $s(x,y)$ alongside the reference image $t(x,y)$ separated by a distance Δ each from origin. The joint image (displayed in LCD1, see Fig. 10) can be written as

$$f(x,y) = s(x,y+\Delta) + t(x,y-\Delta)\ , \tag{14}$$

and its Fourier transform (generated by L1)

$$F(u,v) = S(u,v)\exp(i\Delta v) + T(u,v)\exp(-i\Delta v)\ . \tag{15}$$

The joint power spectrum (captured with CCD camera 1) is given by

$$\begin{aligned} E(u,v) &= \left|F(u,v)\right|^2 = \left|S(u,v)\right|^2 + \left|T(u,v)\right|^2 \\ &+ S(u,v)T^*(u,v)\exp(i2\Delta v) + T(u,v)S^*(u,v)\exp(-i2\Delta v). \end{aligned} \tag{16}$$

Applying the inverse Fourier transform to Eq. (16) (by action of L4) we obtain

$$\begin{aligned} e(x,y) &= s(x,y)\otimes s(x,y) + t(x,y)\otimes t(x,y) \\ &+ s(x,y+2\Delta)\otimes t(x,y+2\Delta) + s(x,y-2\Delta)\otimes t(x,y-2\Delta) \end{aligned}\ . \tag{17}$$

We can see that the autocorrelations of the scene and target images mainly contribute at the origin, whereas the cross-correlation terms, which are the terms of interest, are placed at the distances $\pm 2\Delta$. A drawback of the classical JTC is its low tolerance to geometrical distortions of objects and to noise when objects are embedded in a nonstationary background noise. Assumes that the input image $f(x,y)$ contains the input objects $s(x,y)$ (desired and nondesired) and the non-overlapping background $b(x,y)$:

$$f(x,y) = s(x,y+\Delta) + \tilde{b}(x,y+\Delta) + t(x,y-\Delta), \tag{18}$$

where

$$\tilde{b}(x,y) = w(x-x_0, y-y_0)b(x,y), \tag{19}$$

and (x_0, y_0) are unknown coordinates of the target in the input scene; $w(x-x_0, y-y_0)$ is a binary function defined as

$$w(x-x_0, y-y_0) = \begin{cases} 0, & \text{within the object area} \\ 1, & \text{otherwise} \end{cases}. \tag{20}$$

The joint power spectrum is given by

$$\begin{aligned} |F(u,v)|^2 &= |S(u,v)|^2 + |T(u,v)|^2 + |\tilde{B}(u,v)|^2 \\ &+ \left[T(u,v)S^*(u,v) + T(u,v)\tilde{B}^*(u,v) + S(u,v)\tilde{B}^*(u,v) \right] \exp(i2\Delta v) \\ &+ \left[T^*(u,v)S(u,v) + T^*(u,v)\tilde{B}(u,v) + S^*(u,v)\tilde{B}(u,v) \right] \exp(-i2\Delta v). \end{aligned} \tag{21}$$

Note that the joint power spectrum contains the Fourier transforms with phase the factors of $\exp(\pm i2\Delta v)$ corresponding to the cross-correlation terms between the target and the input objects, the target and the background, and the input objects and the background. The later correlation term severely affects the DC.

To improve the correlation performance of the JTC, several partial solutions were proposed: the nonlinear JTC (Javidi, 1989) and the fringe-adjusted JTC (Alam & Karim, 1993). In the former a nonlinear element-wise transformation of the joint power spectrum is carried out before applying the inverse Fourier transform. In the latter the joint power spectrum is multiplied by the frequency response of a real-valued filter before applying the inverse Fourier transform. These two approaches yield a better performance compared to that of the classical JTC in terms of correlation peak intensity, correlation width, and discrimination capability.

3.2. Adaptive Joint Transform Correlator

We wish to design a JTC that ensures a high correlation peak corresponding to the target while suppressing possible false peaks. To achieve a good recognition of the target, it is necessary to reduce correlation function levels at all sidelobes except at the origin of the correlation plane, where the constraint on the peak value must be meet. For a given object to be recognized and for false objects and background to be rejected, an iterative algorithm is used. At each iteration, the algorithm suppresses the highest sidelobe peak and therefore monotonically increases the value of discrimination capability until a prespecified value is reached. With the help of adaptive SDF filters, a given value of the DC can be achieved.

The first step is to carry out the joint transform correlation between the background and a basic SDF filter, which is initially trained only with the target. Next the intensity maximum of the filter output is set as the origin, and around the origin we form a new object to be rejected from the background. The created object is added to the false class of objects. Now a two-class recognition problem is utilized to design a new SDF filter; that is, the true class contains only the target and the false class consists of the false-class objects. The described iterative procedure is repeated until a given value of DC is obtained. Note that if other false-objects are known, they can be directly included in the false class and used for the design of the adaptive filter. A block diagram of the procedure is shown in Fig. 11.

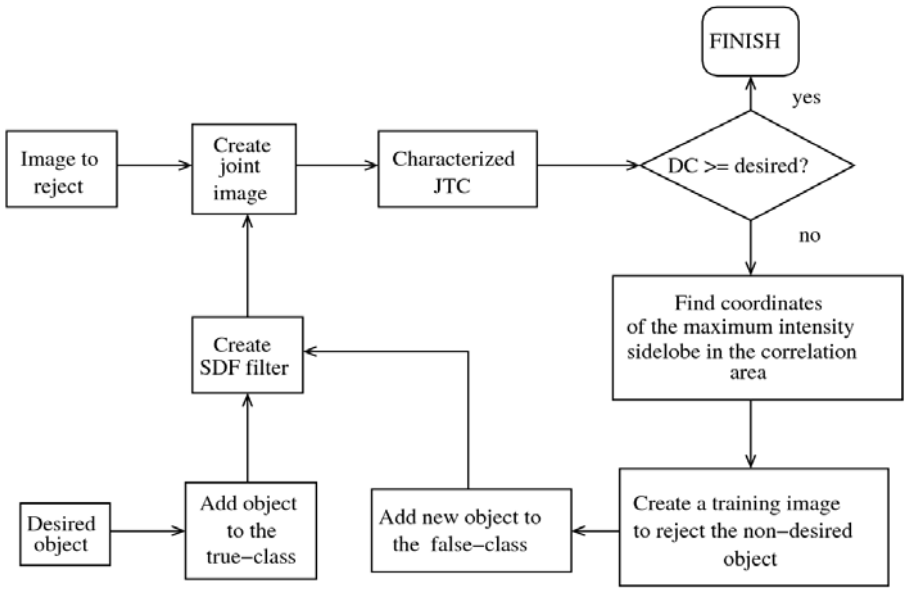


Fig. 11. Block diagram of the iterative algorithm for the design of the adaptive JTC.

The proposed algorithm consists of the following steps:

1. Create a basic SDF filter trained only with the target.
2. Create the input image (see Eq. (14)) by composing the designed SDF filter and the image to be rejected (nondesired objects or a background).
3. Carry out the joint transform correlation including calibration lookup tables of all optoelectronics devices such as a real SLM and a CCD camera.
4. Calculate the DC using Eq. (13).
5. If the value of the DC is greater or equal to the desired value, then the filter design procedure is finished; otherwise, go to the next step.
6. Create a new object to be rejected from the background. The origin of the object is at the highest sidelobe position in the intensity correlation plane. The region of support of the new object is the union of the shapes of all objects involved in the process (desired and non-desired objects). The object is included in the false class of objects.
7. Design a new SDF filter utilizing the two-class recognition problem. The true class contains only the target and the false class consists of the false class objects. Go to step 2.

3.3. Optodigital Implementation

Twisted nematic LCDs are widely used for real-time pattern recognition. Their important characteristics are as follows:

1. They are electrically controlled with standard video signals.
2. They can operate as amplitude-only or phase-only modulators by changing the direction of the polarization vector of the incident light (Lu & Saleh, 1990).
3. They operate at the speed of conventional television standards.

4. They can handle a dynamic range of $[0,255]$ for amplitude modulation and a phase range of $[-\pi,\pi]$ for phase modulation.

In general, the impulse response of SDF filters is a bipolar image. To introduce these kinds of images into spatial light modulators we use two methods.
First method is called bipolar decomposition method. Assume that $h(x,y)$ is a bipolar impulse response:

$$h(x,y)=h^{+}(x,y)-h^{-}(x,y), \tag{22}$$

where

$$h^{+}(x,y)=\begin{cases} h(x,y), & h(x,y)>0 \\ 0, & otherwise \end{cases}, \tag{23}$$

and

$$h^{-}(x,y)=\begin{cases} h(x,y), & h(x,y)\le 0 \\ 0, & otherwise \end{cases}. \tag{24}$$

The intensity cross-correlation between $s(x,y)$ and $h(x,y)$ may be written as follows:

$$\begin{aligned} c(x,y)&=\left|e(x,y)\right|^2=\left|s(x,y)\otimes\left[h^{+}(x,y)\otimes h^{-}(x,y)\right]\right|^2 \\ &=\left|s(x,y)\otimes h^{+}(x,y)\right|^2+\left|s(x,y)\otimes h^{-}(x,y)\right|^2-2\sqrt{\left|s(x,y)\otimes h^{+}(x,y)\right|^2}\sqrt{\left|s(x,y)\otimes h^{+}(x,y)\right|^2}. \end{aligned} \tag{25}$$

It can be seen from Eq. (25), that with the help of decomposition and simple postprocessing, how to obtain the output of the JTC when the reference image has positive and negative values. Note that with the bipolar decomposition method two independent optical correlations are needed.
The second method is referred to as constant addition method. The idea of the method is to transform the input composed bipolar image into an input composed nonnegative image. It can be easily done by adding a bias value to the input bipolar image. Next the joint transform correlation with the input composed nonnegative image is performed. Simple postprocessing is required to obtain the output of the JTC. Note that we need only one optical correlation. The transformed nonnegative joint image can be written as

$$f(x,y)=\tilde{s}(x,y+\Delta)+\tilde{h}(x,y-\Delta), \tag{26}$$

where $\tilde{s}(x,y)=s(x,y)+c$ and, $\tilde{h}(x,y)=h(x,y)+c$, $s(x,y)$ is the scene image, $h(x,y)$ is the bipolar image, and $c=MIN[h(x,y)]$ is a constant value. The intensity output of the JTC with the new joint image is given by

$$c(x,y)=\left|\tilde{s}(x,y)\otimes\tilde{s}(x,y)\right|^2+\left|\tilde{h}(x,y)\otimes\tilde{h}(x,y)\right|^2 \\ +\left|\tilde{s}(x,y+2\Delta)\otimes\tilde{h}(x,y+2\Delta)\right|^2+\left|\tilde{h}(x,y-2\Delta)\otimes\tilde{s}(x,y-2\Delta)\right|^2. \tag{27}$$

The two latter terms of Eq. (27) are the terms of interest. The intensity of the cross-correlation between $s(x,y)$ and $h(x,y)$ can be computed from the intensity of the cross correlation between nonnegative images as follows:

$$\begin{aligned}\left|s(x,y)\otimes h(x,y)\right|^2 &= \left|\left[s(x,y)+c-c\right]\otimes\left[h(x,y)+c-c\right]\right|^2=\left|\left[\tilde{s}(x,y)-c\right]\otimes\left[\tilde{h}(x,y)-c\right]\right|^2 \\ &=\left|\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right|^2+\left|\tilde{h}(x,y)\otimes c\right|^2+\left|\tilde{s}(x,y)\otimes c\right|^2+\left|c\otimes c\right|^2 \\ &-2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]\left[\tilde{h}(x,y)\otimes c\right]\right\}-2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]\left[\tilde{s}(x,y)\otimes c\right]\right\} \\ &+2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]\left[c\otimes c\right]\right\}+2\left\{\left[\tilde{h}(x,y)\otimes c\right]\left[\tilde{s}(x,y)\otimes c\right]\right\} \\ &-2\left\{\left[\tilde{h}(x,y)\otimes c\right]\left[c\otimes c\right]\right\}-2\left\{\left[\tilde{s}(x,y)\otimes c\right]\left[c\otimes c\right]\right\}.\end{aligned} \tag{28}$$

Further simplifying, we can write

$$\begin{aligned}\left|s(x,y)\otimes h(x,y)\right|^2 &= \left|\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right|^2+C_1^2+C_2^2+C_3^2-2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]C_1\right\} \\ &-2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]C_2\right\}+2\left\{\left[\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right]C_3\right\}+2\left(C_1C_2\right)-2\left(C_1C_3\right)-2\left(C_2C_3\right).\end{aligned} \tag{29}$$

Here, $\tilde{s}(x,y)\otimes\tilde{h}(x,y)$ can be obtained by applying the pointwise square root to the intensity $\left|\tilde{s}(x,y)\otimes\tilde{h}(x,y)\right|^2$, constants $C_1=\tilde{h}(x,y)\otimes c$, $C_2=\tilde{s}(x,y)\otimes c$, and $C_3=c\otimes c$ are computed in the following way:

$$\begin{aligned}C_1 &= \alpha\left[\tilde{h}(x,y)\otimes c\right]=\alpha\left[c\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\tilde{h}(x+\tau_x,y+\tau_y)d\tau_x d\tau_y\right]\approx\alpha\left\{c\sum\left[\tilde{h}(x,y)\right]\right\}, \\ C_2 &\approx\alpha\left\{c\sum\left[\tilde{s}(x,y)\right]\right\}\approx\alpha c^2,\end{aligned} \tag{30}$$

where, α is a normalization factor and the symbol "$\sum[\ ]$" denotes the summation of all elements of the image.

3.4. Experimental Results

First we characterized optoelectronics devices such as a twisted neimatic LCD of 800x600 pixels and a monochrome CCD camera of 640x480 pixels. The LCD worked in the amplitude-only modulation regime. Figure 12 shows the experimental calibration lookup table of the intensity response of the LCD captured with the CCD camera.

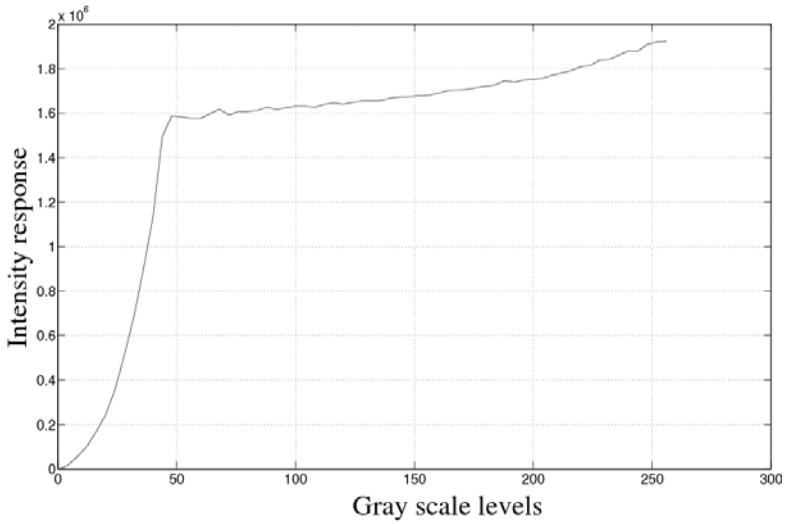
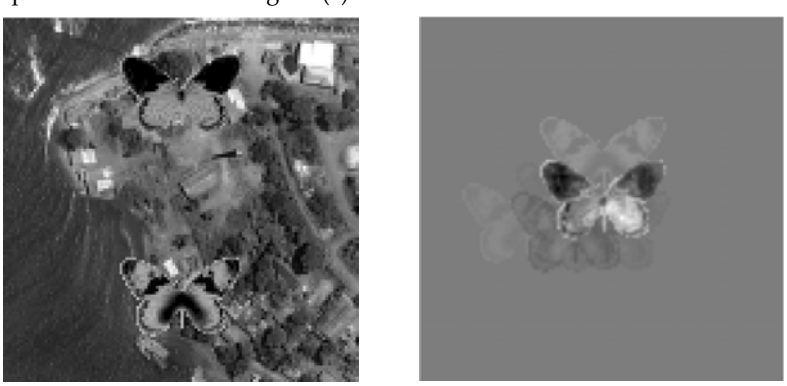


Fig. 12. Intensity response of a twisted neimatic LCD captured with a CCD camera.

It can be seen from Fig. 12 that a gray-scale dynamic range is [0-48]. It is interesting to note that in this range the plot is nonlinear due quantization effects, and it is well approximated with a *k*th-law nonlinearity $\text{Output} = \left(|\text{Input}|^2\right)^{-k}$ when $k = 0.7$. We used this information in the iterative process of the adaptive JTC design.

The size of the input images used in our experiments is 128 x 128 pixels. The signal rage is [0, 255]. The input scene is shown in Fig. 13 (a).

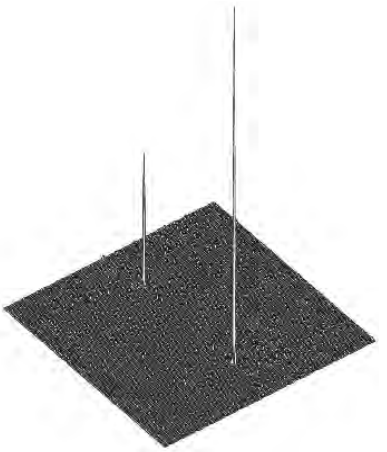


(a) (b)

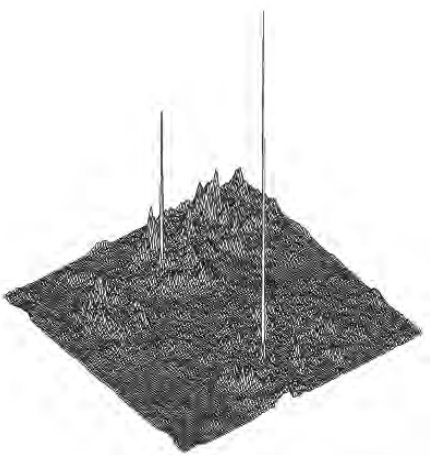
Fig. 13. (a) Input scene containing two objects with similar shapes but with different information content; (b) bipolar reference image obtained with the proposed method.

The scene contains two objects with a similar shape and size (approximately 44x28 pixels) but with different gray-level contents. The target is the upper butterfly with black- wings. The objects are embedded into an aerial picture at unknown coordinates. The performance

of the adaptive JTC in the design process after eight iterations reaches DC = 0.95. The obtained bipolar reference image is shown in Fig. 13 (b).



(a)



(b)

Fig. 14. Computer simulation results obtained for the input scene in Fig. 13 (a) with: (a) binary JTC, (b) fringe-adjusted JTC.

We compare the performance of proposed adaptive JTC with those of the binary JTC and the fringe-adjusted JTC. The intensity correlation planes obtained with latter two systems are shown in Fig. 14. We see that the binary JTC and the fringe-adjusted JTC fail to discriminate the target against the false object with a similar shape. Next we test digitally the recognition performance with the adaptive JTC. The correlation intensity plane obtained with the adaptive JTC for the input scene in Fig. 13 (a) is shown in Fig. 15.

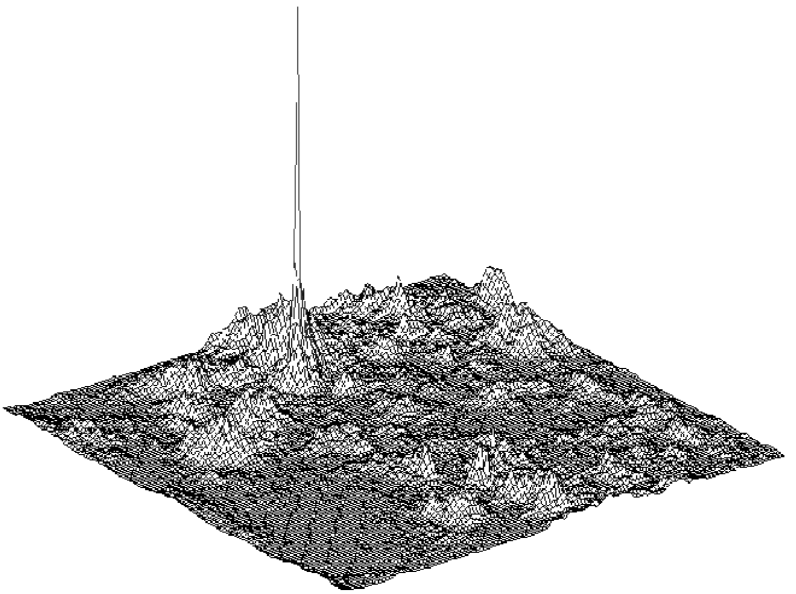


Fig. 15. Computer simulation result obtained for the input scene with the adaptive JTC.

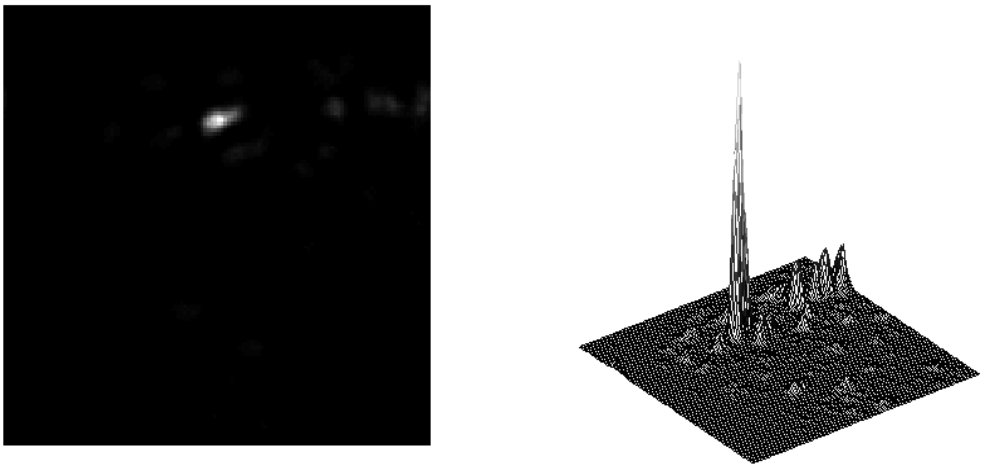
Note that the target is clearly detected. The adaptive JTC architecture can reliably detect a target embedded in a noisy background even if the target presents small geometric image distortions. We used 50 statistical trials of our experiment for different positions of the target. With 95% confidence, the DC obtained in computer simulation is equal to 0.82±0.003.

Bipolar Decomposition Method Results

The first optodigital experiment is based on the bipolar decomposition method. The reference image in Fig. 13(b) has real positive and negative values. We decompose this image into two nonnegative images (see Eqs. (22)-(24)). Two experiments are performed. In the first experiment the input scene is composed with the positive part of the reference image and the joint transform correlation is carried out. The experiment is repeated with the negative part of the reference image. The intensity correlation plane obtained after the postprocessing given in Eq. (25) is shown in Fig. 16. The DC obtained in the experiment is equal to 0.78.

Constant Addition Method Results

The second optodigital experiment is based on the constant addition method described. We use the input image and the reference image shown in Fig. 13. The SLM has a finite size (less than the size of the optical lens), and, after adding a high constant bias to the joint image, the signal at the plane of the SLM may be considered as a signal masked by a rectangular window.



(a) (b)

Fig. 16. Cross-correlation intensity plane obtained with bipolar decomposition method: (a) intensity plane, (b) intensity distribution.

The joint image formed for the constant addition method is shown in Fig. 17.

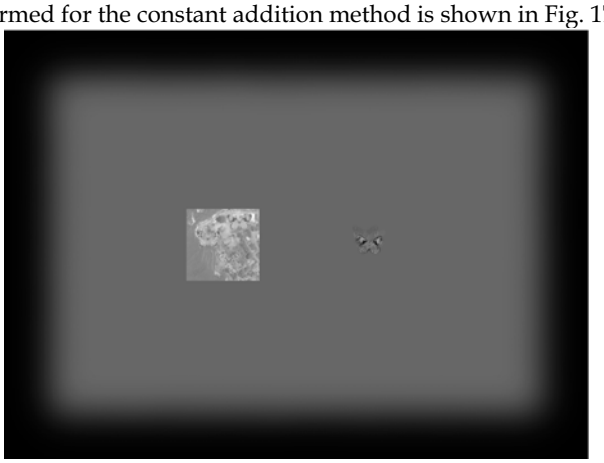
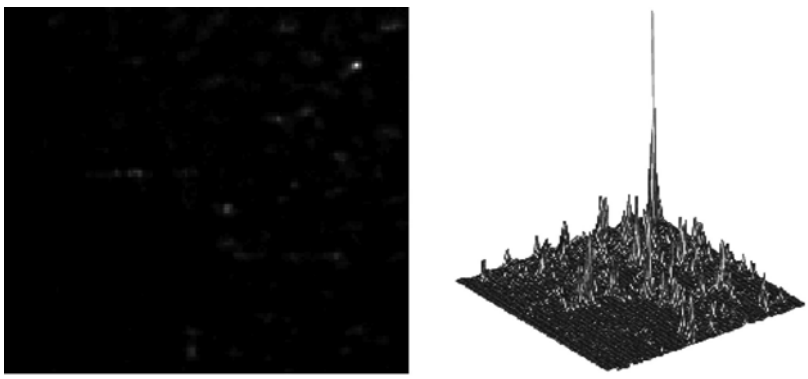


Fig. 17. Joint image formed for the constant addition method.

The Fourier transform of such a signal is the convolution between the spectrum of the joint image and a sinc function (Fourier transform of the rectangular window). Actually, the sinc function possesses high sidelobes that may severely affect the joint power spectrum. To avoid these effects, the input joint image is masked by a window with smoothed edges. Next we calculate all needed constants C_1, C_2, and C_3 given in Eq. (30). Figure 12 gives the relationship between a dynamic range of the used optodigital LCD and CCD camera and a digital range of a signal. Whereas digital images possess a range of [0-255] gray-scale levels, the signals in the optodigital domain have a range of [0-48] levels. We need to scale all images and the constant bias involved in the optodigital setup. The needed constants are equal to $C_1 = 31.75$, $C_2 = 23.03$, and $C_3 = 40$. The α value can be estimated as $\alpha = 1/cs$,

where s is the number of image pixels. The cross-correlation intensity plane obtained in the optodigital JTC after postprocessing is shown in Fig. 18.



(a) (b)

Fig. 18. Cross-correlation plane obtained with constant addition method; (a) intensity plane, (b) intensity distribution.

One can observe that the target is successfully recognized with DC=0.648. Finally, note that this method requires only one optical correlation, whereas the bipolar decomposition method uses two correlations to reconstruct the desired output.

4. Conclusion

Adaptive pattern recognition is still in state of rapid evolution. In this chapter we proposed digital and hybrid optodigital systems designed on the base of adaptive correlation filters to improve recognition of objects in cluttered backgrounds. It was shown that the proposed iterative filter design algorithms with a few training iterations helps us to take the control over the whole correlation plane. The digital systems are based on iterative training of the SDF filters. The hybrid systems additionally take into account real characteristics of used optoelectronics devices. The digital systems can be easily implemented in a computer, whereas the hybrid systems are able to provide real-time pattern recognition. The computer simulation and experimental results demonstrated a good performance of the proposed filters for pattern recognition comparing with known correlation filters. The suggested filters possess high scene-adaptivity, good robustness to small geometric image distortions and input noise.

5. References

Alam, M. S. & Kairm, M. A. (1993). Fringe adjusted joint transform correlator. *Applied Optics*, Vol. 32, No. 23, (August 1993) (4344-4350), ISSN 0003-6935.

Arsenault, H. & Hsu, Y. (1983). Rotation invariant discrimination between almost similar objects. *Applied Optics*, Vol. 22, No. 1, (January 1983) (130-132), ISSN 0003-6935.

Billet, O. & Singher, L. (2002). Adaptive multiple filtering. *Optical Engineering*, Vol. 41, No. 1, (January 2002) (55-68), ISSN 0091-3286.

Casasent, D. (1984). Unified synthetic discriminant function computational formulation. *Applied Optics*, Vol. 23, No. 10, (May 1984) (1620-1627), ISSN 0003-6935.

Diaz-Ramirez, V. H.; Kober, V. & Alvarez-Borrego, J. (2006). Pattern recognition with an adaptive joint transform correlator. *Applied Optics*, Vol. 45, No. 23, (August 2006) (5929-5941), ISSN 0003-6935.

González-Fraga, J. A.; Kober, V. & Alvarez-Borrego, J. (2006). Adaptive synthetic discriminant function filters for pattern recognition. *Optical Engineering*, Vol. 45, No. 5, (May 2006) (0570051-05700510), ISSN 0091-3286.

Horner, J. L. & Gianino, P. D. (1984). Phase-only matched filtering. *Applied Optics*, Vol. 23, No. 6, (March 1984) (812-816), ISSN 0003-6935.

Hester, C. F. & Casasent, D. (1980). Multivariant technique for multiclass pattern recognition. *Applied Optics*, Vol. 19, No. 11, (June 1980) (1759-1761), ISSN 0003-6935.

Javidi, B. (1989). Nonlinear joint power spectrum based optical correlation. *Applied Optics*, Vol. 28, No. 12, (June 1989) (2358-2366), ISSN 0003-6935.

Javidi, B. & Wang, J. (1994). Design of filters to detect a noisy target in nonoverlapping background noise. *Journal OSA (A)*, Vol. 11, No. 10, (October 1994) (2604-2612), ISSN 1084-7529.

Kober, V. & Campos, J. (1996). Accuracy of location measurement of a noisy target in a nonoverlapping background. *Journal OSA (A)*, Vol. 13, No. 8, (August 1996) (1653-1666), ISSN 1084-7529.

Kober, V.; Yaroslavsky, L.P.; Campos, J. & Yzuel, M.J. (1994). Optimal filter approximation by means of a phase only filter with quantization. *Optics Letters*, Vol. 19, No. 13, (July 1994) (978-980), ISSN 0146-9592.

Kober, V.; Mozerov. M & Ovseevich I.A. (2006). Adaptive correlation filters for pattern recognition. *Pattern Recognition and Image Analysis*, Vol. 16, No. 3, (2006) (432-431), ISSN 1054-6618.

Lu, K. & Saleh, B. E. A. (1990). Theory and design of the liquid crystal TV as an optical spatial phase modulator. *Optical Engineering*, Vol. 29, No. 3, (March 1990) (240-247), ISSN 0091-3286.

Moreno, I; Campos, J; Yzuel, M.J. & Kober, V. (1998). Implementation of bipolar real-valued input scenes in a real-time optical correlator: application to color pattern recognition. *Optical Engineering*, Vol. 37, No. 1, (January 1998) (144-150), ISSN 0091-3286.

Mahalanobis, A.; Vijaya Kumar, B.V.K. & Casasent, D. (1987). Minimum average correlation filters. *Applied Optics*, Vol. 26, No. 17, (September 1987) (3633-3640), ISSN 0003-6935.

VanderLugt, A. B. (1964). Signal detection by complex spatial filtering, *IEEE Trans. Inf. Theory*. Vol. 10, No. 2, (April 1964) (139-145), ISSN 0018-9448.

Vijaya Kumar, B.V.K. & Hassebrook, L. (1990). Performance measures for correlation filters. *Applied Optics*, Vol. 29, No. 20, (July 1990) (2997-3006), ISSN 0003-6935.

Waver, C. S. & Goodman, J. L. (1966). Technique for optically convolving two functions. *Applied Optics*, Vol. 5, No. 7, (1966) (1248-1249), ISSN 0003-6935.

Yaroslavsky, L.P. (1993). The theory of optimal methods for localization of objects in pictures. In: *progress in Optics XXXII*, E. Wolf, (Ed.), (145-201), Elsevier, ISBN: 0-444-86923-9, North-Holland.