

YU-JIN ZHANG

ADVANCES IN Image and Video Segmentation



Advances in Image and Video Segmentation

Yu-Jin Zhang, Tsinghua University, Beijing, China



IRM Press
**Publisher of innovative scholarly and professional
information technology titles in the cyberage**

Hershey • London • Melbourne • Singapore

Acquisitions Editor: Michelle Potter
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Copy Editor: Michael Goldberg
Typesetter: Jessie Weik
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
IRM Press (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033-1240
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: <http://www.irm-press.com>

and in the United Kingdom by
IRM Press (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: <http://www.eurospanonline.com>

Copyright © 2006 by Idea Group Inc. All rights reserved. No part of this book may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher.

Product or company names used in this book are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Zhang, Yu-Jin, 1954-

Advances in image and video segmentation / by Yu-Jin Zhang.

p. cm.

Summary: "This book attempts to bring together a selection of the latest results of state-of-the art research in image and video segmentation, one of the most critical tasks of image and video analysis that has the objective of extracting information (represented by data) from an image or a sequence of images (video)"--Provided by publisher.

ISBN 1-59140-753-2 (hardcover) -- ISBN 1-59140-754-0 (soft-cover) -- ISBN 1-59140-755-9 (ebook)

1. Image processing--Digital techniques. 2. Digital video. I. Title.

TA1637.Z52 2006

621.367--dc22

2006009296

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Advances in Image and Video Segmentation

Table of Contents

Preface	vii
---------------	-----

Section I: Introduction

Chapter I

An Overview of Image and Video Segmentation in the Last 40 Years	1
--	---

Yu-Jin Zhang, Tsinghua University, Beijing, China

Section II: Image Segmentation

Chapter II

Optimal Image Segmentation Methods Based on Energy Minimization	17
---	----

Francisco Escolano, Universidad de Alicante, Spain

Miguel A. Lozano, Universidad de Alicante, Spain

Chapter III

Variational Problems in Image Segmentation and Γ -Convergence Methods	46
--	----

Giovanni Bellettini, University of Roma, Italy

Riccardo March, Italian National Research Council, Italy

Chapter IV

A Graph-Based Image Segmentation Algorithm Using Heirarchical Social Metaheuristic	72
--	----

Abraham Duarte, Rey Juan Carlos University, Spain

Angel Sanchez, Rey Juan Carlos University, Spain

Felipe Fernandez, Polytechnique University of Madrid, Spain

Antonio S. Montemayor, Rey Juan Carlos University, Spain

Chapter V	
Modeling Complex Dynamic Systems for Image Segmentation	93
<i>Liang Zhao, University of São Paulo, Brazil</i>	

Section III: Video Segmentation

Chapter VI	
Joint Space-Time-Range Mean Shift-Based Image and Video Segmentation	113
<i>Irene Yu-Hua Gu, Chalmers University of Technology, Sweden</i>	
<i>Vasile Gui, Technical University Timisoara, Romania</i>	
Chapter VII	
Fast Automatic Video Object Segmentation for Content-Based Applications	140
<i>Ee Ping Ong, Institute for Infocomm Research, Singapore</i>	
<i>Weisi Lin, Institute for Infocomm Research, Singapore</i>	
<i>Bee June Tye, Dell Global BV, Singapore</i>	
<i>Minoru Etoh, NTT DoCoMo, Japan</i>	

Chapter VIII	
A Fully Automated Active Shape Model for Segmentation and Tracking of Unknown Objects in a Cluttered Environment	161
<i>Besma Rouai-Abidi, University of Tennessee, USA</i>	
<i>Sangkyu Kang, LG Electronics Inc., Korea</i>	
<i>Mongi Abidi, University of Tennessee, USA</i>	

Chapter IX	
Video Shot Boundary Detection and Scene Segmentation	188
<i>Hong Lu, Fudan University, China</i>	
<i>Zhenyan Li, Nanyang Technological University, Singapore</i>	
<i>Yap-Peng Tan, Nanyang Technological University, Singapore</i>	
<i>Xiangyang Xue, Fudan University, China</i>	

Section IV: Segmenting Particular Images

Chapter X	
Color Image Segmentation in Both Feature and Image Spaces	209
<i>Shengyang Dai, Northwestern University, USA</i>	
<i>Yu-Jin Zhang, Tsinghua University, Beijing, China</i>	
Chapter XI	
Optimizing Texture Primitive Description, Analysis, Segmentation, and Classification Using Variography	228
<i>Assia Kourgli, U.S.T.H.B. University, Algeria</i>	
<i>Aichouche Belhadj-Aissa, U.S.T.H.B. University, Algeria</i>	

Chapter XII	
Methods and Applications for Segmenting 3D Medical Image Data	250
<i>Hong Shen, Siemens Corporate Research, USA</i>	
Chapter XIII	
Parallel Segmentation of Multi-Channel Images Using Multi-Dimensional Mathematical Morphology	270
<i>Antonio Plaza, University of Extremadura, Spain</i>	
<i>Javier Plaza, University of Extremadura, Spain</i>	
<i>David Valencia, University of Extremadura, Spain</i>	
<i>Pablo Martinez, University of Extremadura, Spain</i>	
 Section V: Special Segmentation Applications	
Chapter XIV	
Fuzzy Clustering-Based Approaches in Automatic Lip Segmentation from Color Images	292
<i>Shilin Wang, Shanghai Jiaotong University, China</i>	
<i>Wing Hong Lau, City University of Hong Kong, China</i>	
<i>Alan Wee-Chung Liew, Chinese University of Hong Kong, China</i>	
<i>Shu Hung Leung, City University of Hong Kong, China</i>	
Chapter XV	
Mathematical Morphology-Based Automatic Restoration and Segmentation for Degraded Machine-Printed Character Images	318
<i>Shigueo Nomura, Kyoto University, Japan</i>	
<i>Keiji Yamanaka, Federal University of Uberlândia, Brazil</i>	
<i>Osamu Katai, Kyoto University, Japan</i>	
<i>Hiroshi Kawakami, Kyoto University, Japan</i>	
<i>Takayuki Shiose, Kyoto University, Japan</i>	
Chapter XVI	
Segmentation in Food Images	340
<i>Domingo Mery, Pontificia Universidad Católica de Chile, Chile</i>	
<i>Franco Pedreschi, Universidad de Santiago de Chile, Chile</i>	
Chapter XVII	
Segmentation via Thresholding Methodologies by Using Measure of Fuzziness towards Blind Navigation	355
<i>Farrah Wong, Universiti Malaysia Sabah, Malaysia</i>	
<i>R. Nagarajan, Northern Malaysia University College of Engineering, Malaysia</i>	
<i>Sazali Yaacob, Northern Malaysia University College of Engineering, Malaysia</i>	

Section VI: Segmentation Evaluation

Chapter XVIII

Unsupervised and Supervised Image Segmentation Evaluation 365

Christophe Rosenberger, Université d'Orléans, France

Sébastien Chabrier, Université d'Orléans, France

Hélène Laurent, Université d'Orléans, France

Bruno Emile, Université d'Orléans, France

Chapter XIX

Objective Evaluation of Video Segmentation Quality 394

Paulo Lobato Correia, Technical University of Lisbon, Portugal

Fernando Pereira, Technical University of Lisbon, Portugal

Chapter XX

A Summary of Recent Progresses for Segmentation Evaluation 423

Yu-Jin Zhang, Tsinghua University, Beijing, China

About the Authors 441

Index 452

Preface

Image and video segmentation is one of the most critical tasks of analysis which has the objective of extracting information (represented by data) from an image or a sequence of images (video). In the last 40 years, this field has experienced significant growth and progress, resulting in a virtual explosion of published information.

The field of image and video segmentation is still a very hot topic, with much advancement in recent years. As a consequence, there is considerable need for books like this one, which attempts to bring together a selection of the latest results from researchers involved in state-of-the-art work on image and video segmentation.

This book is intended for scientists and engineers who are engaged in research and development in image and video segmentation and who wish to keep pace with advances in this field. Comprehensive coverage of various branches of image and video segmentation is provided by more than 50 experts around the world. The book includes 20 chapters and they are organized into six sections.

Section I is for the purpose of providing background information, which consists of one introductory survey chapter (Chapter I). Section II is focused on image segmentation, which consists of four chapters (Chapters II through V) showing the advances of image segmentation by using optimization, variational model, meta-heuristic and dynamical systems. Section III is focused on video segmentation, which consists of four chapters (Chapters VI through IX) showing the advances of video segmentation in mean shift-based filtering, video object segmentation, active shape model, and shot boundary detection. Section IV consists of four chapters (Chapters X through XIII) presenting several new algorithms for segmenting particular types of images, such as color, texture, 3-D medical and multi-channel images. Section V contains four chapters (Chapters XIV through XVII) depicting various applications of segmentation techniques in fresh areas, such as human lip segmentation, machine-printed character segmentation, food image segmentation and blind navigation. Section VI presents higher level topics of segmentation evaluation, which consists of three chapters (Chapters XVIII through XX). Unsupervised and supervised evaluations, objective video evaluation as well as a summary of recent evaluation progress are presented.

Chapter I is entitled, “An Overview of Image and Video Segmentation in the Last 40 Years.” A general rendering of research and development of image segmentation in the last 40 years is provided. The history of segmentation of digital images using computers can be traced back 40 years, and since then, this field has evolved very quickly and undergone great change. In this chapter, the position of image segmentation in the general scope of image techniques is first introduced; the formal definition and extension of image segmentation, as well as three layers of research on image

segmentation, are then explained. Based on the introduction and explanations, statistics for the number of developed algorithms is provided, a scheme for classifying different segmentation algorithms is discussed and a summary of existing survey papers for image segmentation is presented.

Chapter II is entitled, “Optimal Image Segmentation Methods Based on Energy Minimization.” Three “case studies” are taken as representatives of recent work on solving segmentation problems (region segmentation, deformable templates matching and grouping) from the energy minimization perspective. Each of the three problems is solved via an optimization approach: respectively jump-diffusion, belief propagation and Bayesian inference. The purpose of the chapter is to show the connection between the formulation of the corresponding cost function and the optimization algorithm. This selection of only three problems and solutions allows the presentation of the fundamental elements of optimization in each particular case and brings the reader to the arena of optimization-based segmentation.

Chapter III is entitled, “Variational Problems in Image Segmentation and Γ -Convergence Methods.” Variational models for image segmentation aim to recover a piecewise, smooth approximation of a given input image together with a discontinuity set which represents the boundaries of the segmentation. In the variational method introduced by Mumford and Shah, the length of the discontinuity boundaries in the energy is included; such a geometric term makes the minimization of the corresponding functional a difficult numerical problem. A mathematical framework for the Mumford-Shah functional is considered. The use of the Γ -convergence theory to approximate the functional by elliptic is suggested. The design of an iterative numerical scheme for image segmentation based on the Γ -convergent approximation is discussed. The relation between the Mumford-Shah model and the Perona-Malik equation has also been discussed.

Chapter IV is entitled, “A Graph-Based Image Segmentation Algorithm Using Hierarchical Social Metaheuristic.” This chapter proposes a new evolutionary graph-based image segmentation method to improve quality results. Such an approach is quite general. It starts from an image described by a simplified, undirected weighted graph where nodes represent either pixels or regions and weighted edges measure the dissimilarity between pairs of pixels or regions. The resulting graph is successively partitioned into two sub-graphs in a hierarchical fashion, corresponding to the two most significant components of the actual image, until a termination condition is met. This graph-partitioning task is solved as a variant of the min-cut problem (normalized cut) using a Hierarchical Social (HS) meta-heuristic. As a consequence of this iterative graph bipartition stage, pixels or regions are initially merged into the two most coherent components, which are successively bi-partitioned according to this graph-splitting scheme.

Chapter V is entitled “Modeling Complex Dynamical Systems for Image Segmentation.” As motivated by biological experimental findings, two network models of coupled chaotic elements for image segmentation are introduced in this chapter. In both models, time evolutions of chaotic elements that correspond to the same object in a given image are synchronized with one another, while this synchronized evolution is desynchronized with respect to time evolution of chaotic elements corresponding to other objects in the image. The first model is a continuous flow, and the segmentation process incorporates geometrical information of input images, while the second model is a network of discrete maps for pixel clustering, accompanying an adaptive moving mechanism to eliminate pixel ambiguity.

Chapter VI is entitled, “Joint Space-Time-Range Mean Shift-Based Image and Video Segmentation.” In this chapter, image and video segmentation is addressed by using mean shift-based filtering. A variety of mean shift filtering approaches are described for image/video segmentation and nonlinear edge-preserving image smoothing. A joint space-time-range domain mean shift-based video segmentation approach is presented. Segmentation of moving/static objects/background is obtained through inter-frame mode-matching in consecutive frames and motion vector mode estimation. Newly appearing objects/regions in the current frame, due to new foreground objects or uncovered background regions, are segmented by intra-frame mode estimation.

Chapter VII is entitled, “Fast Automatic Video Object Segmentation for Content-Based Applications.” An algorithm has been devised for fast, fully automatic and reliable object segmentation from live video for scenarios with static camera. Methods for: (1) adaptive determination of the threshold for change detection; (2) robust stationary background reference frame generation, which, when used in change detection, can reduce segmentation fault rate and solve the problems of occluded objects appearing as part of segmented moving objects; (3) adaptive reference frame selection to improve segmentation results; and (4) spatial refinement of modified change detection mask by incorporating information from edges, gradients and motion to improve the accuracy of segmentation contours are proposed.

Chapter VIII is entitled, “A Fully Automated Active Shape Model for Segmentation and Tracking of Unknown Objects in a Cluttered Environment.” A fully automated active shape model (ASM) for the tracking of non-rigid unknown objects in a cluttered and changing environment is described. The segmentation of shapes is automated, using a new objective function to deform and move a contour toward the actual shape. New profile modeling and optimization criteria to automatically find corresponding points are also applied for segmentation and tracking of people in cluttered backgrounds. This algorithm presents a major extension to the state-of-the-art and the original ASM, which was designed for known objects in smooth nonchanging backgrounds, and where the landmark points need to be manually picked offline. This is a fully automated, real time ASM that deals with changing backgrounds and does not require prior knowledge of the object to be segmented and tracked.

Chapter IX is entitled, “Video Shot Boundary Detection and Scene Segmentation.” This chapter presents a new and efficient method for shot boundary detection (SBD) and scene segmentation. The new SBD method is based on sequential change detection to achieve improved detection accuracy. The method is then extended to segment videos into scenes. Compared with existing scene segmentation methods, the proposed method can also obtain more accurate results over a large set of test videos.

Chapter X is entitled, “Color Image Segmentation in Both Feature and Image Spaces.” Watershed algorithm is traditionally applied on image domain. It fails to capture the global color distribution information. In this chapter, the watershed algorithm is first applied in feature space to extract clusters with irregular shapes. After getting the initial segmentation result by feature space analysis, attention is turned to image space, and the final result is obtained by minimizing a global energy function based on Markov Random Field theory. Two efficient energy minimization algorithms, Graph Cuts and Highest Confidence First (HCF), are explored under this framework.

Chapter XI is entitled, “Optimising Texture Primitives Description, Analysis, Segmentation, and Classification Using Variography.” Most approaches dealing with various aspects of texture analysis and segmentation require the application of a template

to a given image, pixel by pixel, to yield a new image. The selection of an appropriate window size is critical and affects directly the results obtained. In this chapter, a new approach based on the concept of variography is proposed to automatically select the optimal window. Some direct applications, including textural primitive's description, mathematical morphology, textural segmentation and textural classification are reported.

Chapter XII is entitled, “Methods and Applications for Segmenting 3D Medical Image Data.” An overview of the popular and relevant methods that may be applicable for the general problem of 3D medical image segmentation is provided, with a discussion about their advantages and limits. Specifically, the issue of incorporating prior knowledge into the segmentation of anatomic structures is discussed and the concept and issues of Knowledge Based Segmentation are described in detail. Typical sample applications will accompany the discussions throughout this chapter. This will help an application developer to gain insights in the understanding and application of various computer vision approaches to solve real-world problems of medical image segmentation.

Chapter XIII is entitled, “Parallel Segmentation of Multichannel Images Using Multidimensional Mathematical Morphology.” It is recognized that mathematical morphology-based segmentation of multi-channel imagery has not been fully achieved yet, mainly due to the lack of vector-based strategies to extend classic morphological operations to multidimensional imagery. In this chapter, a vector-preserving framework to extend morphological operations to multi-channel images is described, and a fully automatic, multi-channel watershed segmentation algorithm that naturally combines spatial and spectral/temporal information is proposed. Due to the large data volumes often associated with multi-channel imaging, a parallel implementation strategy to speed up performance is also developed.

Chapter XIV is entitled, “Fuzzy Clustering-Based Approaches for Automatic Lip Segmentation from Color Images.” Lip image segmentation plays an important role in lip image analysis, which has recently received much attention because the visual information extracted has been shown to provide significant improvement for speech recognition and speaker authentication, especially in noisy environments. This chapter describes different lip image segmentation techniques, with emphasis on segmenting color lip images. The state-of-the-art classification-based techniques for color lip segmentation—the “spatial fuzzy c-mean clustering (SFCM)” and the “fuzzy c-means with shape function (FCMS)” are described in detail. These methods integrate color information along with different kinds of spatial information into a fuzzy clustering structure.

Chapter XV is entitled, “Mathematical Morphology Based Automatic Restoration and Segmentation for Degraded Machine-Printed Character Images.” This chapter presents a morphological approach for automatic segmentation of seriously degraded machine-printed character images. This approach consists of four modules: (1) detecting and segmenting natural pitch characters based on the vertical projection of their binary images; (2) detecting fragments in broken characters and merging these fragments before the eventual segmentation; (3) employing a morphological thickening algorithm on the binary image for locating the separating boundaries of overlapping characters; and (4) executing a morphological thinning algorithm and calculating segmentation cost for determining the most appropriate coordinate at the image for dividing touching characters.

Chapter XVI is entitled, “Segmentation in Food Images.” A robust algorithm to segment food image from a background is presented using colour images in this chapter. The proposed method has three steps: (1) computation of a high contrast grey value image from an optimal linear combination of the RGB colour components; (2) estimation of a global threshold using a statistical approach; and (3) morphological operation in order to fill the possible holes presented in the segmented binary image. The segmentation performance was assessed by computing the area A_z under the Receiver Operation Characteristic (ROC) curve.

Chapter XVII is entitled, “Segmentation via Thresholding Methodologies by Using Measure of Fuzziness Towards Blind Navigation.” Blind navigation is specialized research directed toward the development of navigation aids for blind people to minimize assistance from sighted individuals during navigation. In this paper, two methodologies of segmentation are detailed and certain aspects of the methodologies are compared. Measure of fuzziness is applied in both the segmentation methodologies to find the threshold values. The first methodology was developed for a single camera, whereas the second was developed for stereo camera systems.

Chapter XVIII is entitled, “Unsupervised and Supervised Segmentation Evaluation.” Though many segmentation methods have been proposed in the literature, it is difficult to compare their efficiency. In order to solve this problem, some evaluation criteria have been proposed for the last decade to quantify the quality of a segmentation result. Supervised evaluation criteria use some *a priori* knowledge, such as a ground truth, while unsupervised evaluation computes some statistics in the segmentation result according to the original image. The main objective of this chapter is to review both types of evaluation criteria from the literature first, then to make a comparative study in order to identify their efficiency for different types of images.

Chapter XIX is entitled, “Objective Evaluation of Video Segmentation Quality.” The current practice for the evaluation of video segmentation quality is based on subjective testing, which is an expensive and time-consuming process. Objective segmentation quality evaluation techniques can alternatively be used, once appropriate algorithms become available. The evaluation methodologies and objective segmentation quality metrics, both for individual objects and complete segmentation partitions, are introduced. Standalone and relative evaluation metrics are proposed for use when reference segmentation is missing, or available for comparison, respectively.

Chapter XX is entitled, “A Summary of Recent Progress for Segmentation Evaluation.” This chapter provides a summary of the recent (especially in the 21st century) progress in evaluating image and video segmentation. It is seen that much more attention has been given to this subject recently than several years ago. A number of works are based on previous proposed principles, several works made modifications and improvements on previous proposed techniques and some works presented new ideas. The generality and complexity of the evaluation methods and performance criteria used in these works have been thoroughly compared. As the research in this field is still on the rise, some existing problems and several future research directions are also pointed out.

*Yu-Jin Zhang
Editor
Tsinghua University, Beijing, China*

Acknowledgments

First, credit goes to the senior academic editor of Idea Group Publishing, Mehdi Khosrow-Pour, for the invitation made to me to organize and edit this book.

Sincere thanks go to the other 50 authors, coming from 16 countries and regions (20 from Asia, 20 from Europe, four from North America, four from South America and two from Africa), who made their great contributions to this project by submitting chapters, and reviewing chapters, among other things.

Special gratitude goes to all the staff at Idea Group Publishing for their valuable communication, guidance and suggestions along the development process.

Last, but not least, I am indebted to my wife, my daughter and my parents for their encouragement, patience, support, tolerance, and understanding during the last two years.

*Yu-Jin Zhang
Editor
Tsinghua University, Beijing, China*

Section I: Introduction

Chapter I

An Overview of Image and Video Segmentation in the Last 40 Years

Yu-Jin Zhang, Tsinghua University, Beijing, China

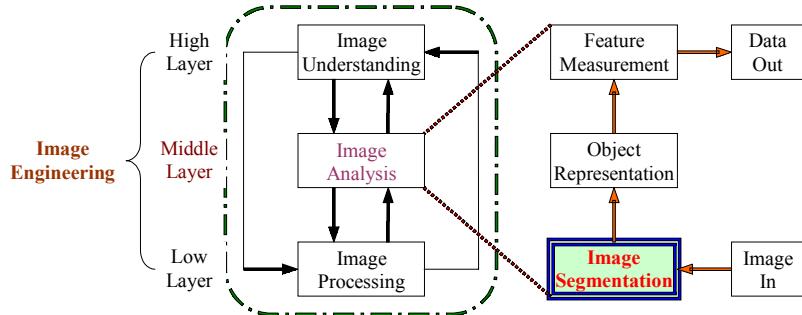
ABSTRACT

The history of segmentation of digital images using computers could be traced back 40 years. Since then, this field has evolved very quickly and has undergone great change. In this chapter, the position of image segmentation in the general scope of image techniques is first introduced; the formal definition and extension of image segmentation as well as three layers of research on image segmentation are then explained. Based on the introduction and explanations, statistics for a number of developed algorithms is provided, the scheme for classifying different segmentation algorithms is discussed and a summary of existing survey papers for image segmentation is presented. These discussions provide a general rendering of research and development of image segmentation in the last 40 years.

INTRODUCTION

Image, from its general sense, could embrace all media that can be visualized by human beings, such as still image, video, animation, graphics, charts, drawings and even text. From images, human beings obtain the majority of information from the real world. To better perceive images and to gain more information from these perceptions, various techniques have been developed and many applications have been discovered.

Figure 1. Image engineering and image segmentation



All image techniques can be grouped under a general framework—image engineering (IE), which consists of three layers: image processing (low layer), image analysis (middle layer) and image understanding (high layer), as shown in Figure 1 (Zhang, 2002a). In recent years, image engineering has formed a new discipline and made great progress (Zhang, in press).

Image segmentation is the first step and also one of the most critical tasks of image analysis. It has the objective of extracting information (represented by data) from an image via image segmentation, object representation and feature measurement (Figure 1). It is evident that the results of segmentation will have considerable influence over the accuracy of feature measurement (Zhang, 1995).

Image segmentation is often described as the process that subdivides an image into its constituent parts and extracts those parts of interest (objects). It is one of the most critical tasks in automatic image analysis because the segmentation results will affect all the subsequent processes of image analysis, such as object representation and description, feature measurement and even the following higher level tasks such as object classification and scene interpretation.

The first development of techniques for image segmentation can be traced back 40 years. In 1965, an operator for detecting edges between different parts of an image, the Roberts operator (also called the Roberts edge detector), was introduced (Roberts, 1965). This detector was the first step toward decomposing an image into its constitutional components. Since then, a large number of techniques and algorithms for image segmentation have been proposed, the result of much effort devoted to the research and application of image segmentation processes and development. In the meantime, concept and scope of images have been extended greatly. The extension of 2-D images to 3-D, still images to moving images or sequences of images (video), gray level images to color or multi-band images, etc. have also helped the concepts and techniques of image segmentation expand widely.

In spite of several decades of investigation, image segmentation remains a challenging research topic. Two bibliographical factors supporting this are:

1. Many conferences on image techniques have sessions for image segmentation. The number of papers on image segmentation increases steadily every year (Zhang, 2006).
2. Almost all books on image processing, analysis and understanding (computer vision) have chapters for image segmentation. However, to our knowledge, very few books (monographs) specialize in image segmentation (Mediode, 2000; Zhang, 2001a).

The first factor shows that the research on image segmentation is still evolving, and the second that the research is far from maturation. It is then evident that an overview of the progress of image segmentation would be useful for further development.

BACKGROUND

Formal Definition

Considering image segmentation as the partition of an image into a set of non-overlapping regions whose union is the entire image, some rules to be followed for regions resulting from the image segmentation can be stated as (Haralick, 1985):

1. They should be uniform and homogeneous with respect to some characteristics;
2. Their interiors should be simple and without many small holes;
3. Adjacent regions should have significantly different values with respect to the characteristic on which they are uniform; and
4. Boundaries of each segment should be simple, not ragged, and must be spatially accurate.

A formal definition of image segmentation, supposing the whole image is represented by R and R_i , where $i = 1, 2, \dots, n$ are disjoint non-empty regions of R , consists of the following conditions (Fu, 1981):

1. $\bigcup_{i=1}^n R_i = R$;
2. for all i and j , $i \neq j$, there exists $R_i \cap R_j = \emptyset$;
3. for $i = 1, 2, \dots, n$, it must have $P(R_i) = \text{TRUE}$;
4. for all $i \neq j$, there exists $P(R_i \cup R_j) = \text{FALSE}$;

where $P(R_i)$ is a uniformity predicate for all elements in set R_i and \emptyset represents an empty set.

Some have thought the following condition is also important:

5. For all $i = 1, 2, \dots, n$, R_i is a connected component.

In the above, condition (1) points out that the summation of segmented regions could include all pixels in an image; condition (2) points out that different segmented regions could not overlap each other; condition (3) points out that the pixels in the same segmented regions should have some similar properties; condition (4) points out that the pixels belonging to different segmented regions should have some different properties; and finally, condition (5) points out that the pixels in the same segmented region are connected.

Definition Extension

As mentioned in the introduction, the concept and scope of image have been extended widely. If the basic 2-D still gray level image is represented by $f(x, y)$, then the extension of 2-D images to 3-D can be represented by $f(x, y) \Rightarrow f(x, y, z)$; the extension of still images to moving images or sequences of images can be represented by $f(x, y) \Rightarrow f(x, y, t)$; a combination of the above extensions can be represented by $f(x, y) \Rightarrow f(x, y, z, t)$; and the extension of gray level images to, for example, color images or multi-band images (in combining all the above extensions) can be represented by $f(x, y) \Rightarrow f(x, y, z, t)$.

Considering the extension of images, the definitions of image segmentation may also need to be extended. With the extension of $f(x, y) \Rightarrow f(x, y, z)$, $f(x, y) \Rightarrow f(x, y, t)$ and $f(x, y) \Rightarrow f(x, y, z, t)$, the regions in all the above conditions should be extended to some high-dimensional blobs. With the extension of $f(x, y) \Rightarrow f(x, y, z, t)$, the properties of image elements become vectors, so the logic predicate defined for conditions (3) and (4) should be modified to incorporate vector information. Once done, the above five conditions can still be used to define the image segmentation.

Two notes that relate to the extension of the concepts of images and image segmentation are as follows: First, when 2-D images are extended to 3-D images, i.e., $f(x, y) \Rightarrow f(x, y, z)$, the original pixel should be replaced by a 3-D voxel (volume element). For even higher dimensional images, no universal image element has been defined. Second, when in cases of $f(x, y) \Rightarrow f(x, y, t)$ and $f(x, y) \Rightarrow f(x, y, z, t)$, the extended images can be segmented either in space (i.e., x, y, z) or in time domain (i.e., temporal segmentation). In both cases, the principle indicated by conditions (3) and (4) is still the similar properties inside each component and the different properties for adjacent components.

Three Levels of Research

Research on image segmentation began with developing techniques for segmenting images. However, there is yet no general theory for image segmentation. So, this development has traditionally been an *ad hoc* process. As a result, many research directions have been exploited, some very different principles have been adopted and a wide variety of segmentation algorithms have appeared in the literature. It has been noted by many that none of the developed segmentation algorithms are generally applicable to all images and different algorithms are not equally suitable for particular applications.

With the increase in the number of algorithms for image segmentation, evaluating the performance of these algorithms becomes indispensable in the study of segmentation. Considering the various modalities for acquiring different images and the large number of applications requiring image segmentation, selecting appropriate algorithms

becomes an important task. A number of evaluation techniques have been proposed; for those published in the last century, see Zhang (1996, 2001b) for survey papers.

The technique of evaluation of image segmentation can be categorized into two types: characterization and comparison. Characterization may be seen as an intra-technique process while technique comparison as an inter-technique one. Both emphasize the evaluation of an algorithm's performance but not its development. In other words, not the design but the behavior of an algorithm is taken into account.

While evaluation techniques have gained more and more attention, with numerous evaluation methods newly designed, how to characterize the different existing methods for evaluation has also attracted interest. In fact, different evaluation criteria and procedures, their applicability, advantages and limitations need to be carefully and systematically studied.

According to the above discussion, the research for image segmentation is carried on at three levels. The first, and also the most basic, is the level of algorithm development. The second, at the middle, is the level of algorithm evaluation, and the third, at the top, is the systematic study of evaluation methods.

The present chapter will concentrate on the first level of segmentation, while discussion of the state-of-art in second and third levels will be given in Chapter XX.

MAIN THRUST

The current study focuses on statistics about the number of segmentation algorithms developed, how different segmentation techniques are classified and on a general overview of survey papers published in the last 40 years.

Number of Developed Algorithms

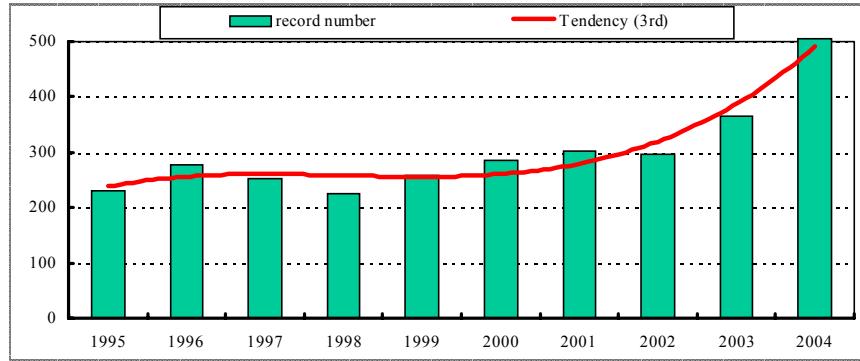
Over the last 40 years, the research and development of segmentation techniques has gone on steadily. A great number of segmentation algorithms have been developed and this number is continually increasing. More than 10 years ago, an estimation of the number of internationally proposed algorithms for image segmentation had been made (Zhang, 1994). It was first pointed out that the cumulative number should approach one thousand (instead of "hundreds" as some were still predicting) at that time. Now, with the advent of network search engines, a search using the term "image segmentation" in title field from EI Compendex provides the list of English records (papers) as shown in Table 1. These records were collected in April 2006.

From Table 1, it is obvious that the estimation made more than 10 years ago has been verified. The record of numbers in the last 10 years is plotted in Figure 2, together with a tendency curve (third order polynomial).

Table 1. List of image segmentation records found in EI Compendex

1965-1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	Total
965	232	278	253	226	258	287	303	298	365	506	326	4297

Figure 2. Number of records and the tendency of development in 1995~2004



It can be seen from Figure 2 that the curve is flat for the first 5 years and increases quickly for the last 5 years. A related but distinct study can be found in Zhang (2006), from which some statistics for an even wider scope of image techniques (including image segmentation) over the last 10 years may be found, and a comparison of the developmental tendency of image segmentation with that of another popular image technique—image coding—is also provided.

Classification of Algorithms

With so many algorithms having been developed, classification of various techniques for image segmentation becomes an essential task. Different schemes have been proposed. For example, segmentation algorithms have been divided into three groups (Fu, 1981):

1. Thresholding or clustering (the latter is the multi-dimensional extension of the former)
2. Edge detection
3. Region extraction

The problem with this classification is that as thresholding is also a region extraction approach in reality, group (1) is just a special sub-group of group (3).

Another study considers various segmentation algorithms in six groups (Pal, 1993):

1. Thresholding
2. Pixel classification (including relaxation, Markov random field based approaches and neural network based approaches)
3. Range image segmentation
4. Color image segmentation
5. Edge detection

6. Methods based on fuzzy set theory (including fuzzy thresholding, fuzzy clustering and fuzzy edge detection)

It is clear that the above six groups are somehow overlapping from the technique point of view. For example, the groups of range image segmentation and color image segmentation emphasize how images are to be segmented. However, the algorithms for segmenting these images are still based on thresholding, pixel classification or edge detection, as indicated by the authors (Pal, 1993). On the other hand, the group of algorithms based on fuzzy set theory is a combination of fuzzy set theory with groups (1), (2) and (5). Thus, in fact, only three groups of segmentation algorithms are distinguishable here. Lastly, the algorithms in groups (1) and (2) have many similarities (Fu, 1981), while commonly employed region-growing techniques, for example, cannot be included among these groups.

A classification of algorithms into groups, in principle, is a problem of set partition into subsets. With reference to the definition of segmentation (Fu, 1981), it was believed that the resultant groups, after an appropriate classification of segmentation algorithms according to the process and objective, should satisfy the following four conditions (Zhang, 1997):

1. Every algorithm must be in a group
2. All groups together can include all algorithms
3. Algorithms in the same group should have some common properties
4. The algorithms in different groups should have certain distinguishable properties

Classifications of algorithms are performed according to specific classification criteria. The first two conditions imply that the classification criteria should be suitable for all algorithms. The last two conditions imply that the criteria should determine the representative properties of each algorithm group. Keeping these conditions in mind, the following two criteria appear to be suitable for the classification of segmentation algorithms.

Gray level image segmentation is generally based on one of two basic properties of gray level values in images: discontinuity and similarity (Conzalez, 2002). Thus, two categories of algorithms can be distinguished: the boundary-based ones that detect object contours explicitly by using the discontinuity property and the region-based ones that locate object areas explicitly according to the similarity property. These two categories may be considered as complementary. On the other hand, according to the processing strategy, segmentation algorithms can be divided into sequential and parallel classes (Rosenfeld, 1981). In the former, some cues from the early processing steps are taken for the subsequent steps. While in the latter, all decisions are made independently and simultaneously. Both strategies are also complementary from the processing point of view.

Combining the two types of categorizations, four groups of techniques: G1, G2, G3 and G4 can be defined as shown in Table 2.

It can be verified that such a classification scheme satisfies the above four conditions for algorithms. These four groups can cover/include all existing segmentation algorithms, such as those surveyed by Fu (1981) as well as Pal (1993). Most edge detection based segmentation procedures can be categorized as belonging to group G1,

Table 2. General classification of segmentation algorithms

Classification	Edge-based (discontinuity)	Region-based (similarity)
Parallel process	G1: Edge-based parallel process	G3: Region-based parallel process
Sequential process	G2: Edge-based sequential process	G4: Region-based sequential process

while other edge-based algorithms using processes such as edge linking and boundary following, which are inherently sequential, could be better classified in the G2 group. All thresholding and clustering techniques and many procedures considering segmentation as a pixel/voxel classification problem belong to the G3 group. Methods based on multi-resolution structure, region growing as well as region split and merge are often labeled under the group G4.

The algorithms in each group have some common characteristics. In the study of segmentation algorithms, typical examples are often selected as representative of the group. For example, in an evaluation of different groups of segmentation algorithms (Zhang, 1997), two algorithms, the Canny operator edge detecting and boundary closing, have been taken from group G1; the algorithm using dynamic programming techniques for contour searching is taken from group G2; the algorithm based on improved histogram concavity analysis is taken from group G3; while the algorithm employs split, merge and group approach is taken from group G4. Recently, a number of researchers have combined the primitive algorithms in diverse groups to form new composite ones. Though different strategies can be used (Munoz, 2002), the fundamental principles of basic algorithms are unaffected.

New algorithms based on many different mathematical theories and models, such as Bayesian theory, Brownian string, expert system, fractal, Gabor filtering, Gaussian mixture models, generic algorithms, Gibbs Random Field, hidden Markov models, Markov random field (MRF), multi-scale edge detection, simulated annealing, wavelet modulus maxima, and so forth, have attracted the consideration of many researchers. The above general classification scheme is still applicable for these new algorithms. For example, algorithms based on the SUSAN operator belong to group G1; ACM and ASM belong to group G2; different thresholding techniques, no matter what they are based on wavelet transformation, maximum/minimum entropy or fuzzy divergence, or even fuzzy C-means, belong to group G3; watershed algorithms correspond to the boundary of object, but segmentation techniques using watershed are usually based on region attributes (Roerdink, 2000); like region-growing techniques, watershed uses region-based properties to determine the region boundary and thus could be categorized into group G4.

Compared to the spatial-nature of (static) images, video has both spatial nature and temporal nature. Segmenting a frame of video in spatial domain is just like segmenting a static image. Segmenting a sequence of video frames in temporal domain is called temporal segmentation or shot detection (Zhang, 2002b). The purpose is to divide a video sequence into its constitute units—shots. The principle used in this segmentation is still

like that in spatial domain. In fact, the difference between adjacent frames and the similarity among consecutive frames could be used to determine the frontier of shots. The former corresponds to edge-based techniques while the latter corresponds to region-based techniques. In edge-based image segmentation, the inter-region disparity between one region and its comparison to their neighbors is considered. In edge-based video segmentation, neighbors should be adjacent frame and most temporal segmentation, shot-detection methods are dependent on discrepancy between frames. In region-based image segmentation, the intra-region homogeneity is taken into account. In region-based video segmentation, motion uniformity across frames or the temporal stability of certain region features can be used.

In Table 2, the classification is shown for the top group level. For each group, sub-group level classification is still possible. For example, thresholding is a popular tool used in image segmentation and a wide range of thresholding techniques has been developed—a survey of them can be found in Sahoo (1988). Determination of appropriate threshold values is the most important task involved in thresholding techniques. Threshold values have been determined in different techniques by using rather different criteria. A classification of thresholding techniques can be based on how the threshold values are determined (Zhang, 1990). The threshold T is a function of the form $T = T[x, y, f(x, y), g(x, y)]$, where $f(x, y)$ is the gray level of a point located at (x, y) , and $g(x, y)$ denotes some local properties of this point. When T depends solely on $f(x, y)$, the thresholding technique is point-dependent. If T depends on both $f(x, y)$ and $g(x, y)$, then the thresholding technique is region-dependent. If, in addition, T depends also on the spatial coordinates x and y , the thresholding technique will be coordinate-dependent.

Another classification of thresholding techniques takes into account the application range of thresholds. The thresholds obtained by both point-dependent and region-dependent techniques will be applied to the whole image, so these techniques could be called global techniques. The thresholds obtained by coordinate-dependent techniques, on the other hand, will be applied to each pixel of each sub-image, so these techniques could be called local techniques.

Further classification could still be made. For example, according to the information exploited (Marcello, 2004), the above mentioned global techniques have been classified into the following groups (this list could be and in fact is being augmented):

1. Histogram shape-based methods (where the peaks, valleys, curvatures, etc., of the smoothed histogram are analyzed)
2. Clustering-based methods (where the grey level samples are clustered in two parts as background and foreground or, alternately, are modeled as two Gaussian distributions)
3. Entropy-based methods (where the entropy of the foreground-background regions, the cross-entropy between the original and segmented image, etc., are calculated)
4. Object attribute-based methods (where a measure of similarity between the grey-level and segmented images, such as fuzzy similarity, shape, edges, number of objects, etc., are investigated)
5. Spatial relation-based methods (where probability mass function models take into account correlation between pixels on a global scale are used)

Summary of Survey Papers

Along with the development of image segmentation algorithms, a number of survey papers for general image segmentation algorithms have been presented in the literature over the last 40 years (Davis, 1975; Zucker, 1976; Riseman, 1977; Zucker, 1977; Weszka, 1978; Fu, 1981; Rosenfeld, 1981; Peli, 1982; Haralick, 1985; Nevatia, 1986; Pavlidis, 1986; Borisenko, 1987; Sahoo, 1988; Buf, 1990; Sarkar, 1993; Pal, 1993), though they only partially cover the large number of techniques developed. In partitioning the last 40 years into four decades, it is interesting to note that all these survey papers are dated in the second and third decades. The reason for a lack of surveys in the first decade is because the research results were just cumulating during that period. The reason for no survey results in the last decade may be attributed to the factor that so many techniques have already been presented, that a comprehensive survey becomes less feasible.

Though no general survey for the whole scope of image segmentation has been made in the last 10 years, some specialized surveys have nevertheless been published in recent years. These survey papers can be classified into two sub-categories:

1. Survey focused on particular group of segmentation algorithms:

Many segmentation algorithms have been developed by using certain mathematical/theoretical tools, such as fuzzy logic, genetic algorithms, neural networks (NN), pattern recognition, wavelet, and so forth, or based on unique frameworks, such as active contour models (ACM), thresholding, watershed, and so forth. Some surveys for algorithms using the same tools or based on the same frameworks have been made, for example:

Because using fully automatic methods sometimes would fail and produce incorrect results, the intervention of a human operator in practice is often necessary. To identify the patterns used in the interaction for the segmentation of medical images and to develop qualitative criteria for evaluating interactive segmentation methods, a survey of computational techniques involving human-computer interaction in image segmentation has been made (Olabarriaga, 2001). This survey has taken into account the type of information provided by the user, how this information affects the computational part of the method and the purpose of interaction in the segmentation process for the classification and comparison of a number of human-machine dialog methods.

Algorithms combining edge-based and region-based techniques will take advantage of the complementary nature of edge and region information. A review of different segmentation methods which integrate edge and region information has been made (Freixenet, 2002). Seven different strategies to fuse such information have been highlighted.

Active shape model (ASM) is a particular structure for finding the object boundary in images. Under this framework, various image features and different search strategies can be used, which makes for a range of ASM algorithms. A number of these variations for the segmentation of anatomical bone structures in radiographs have been reviewed in Behiels (2002).

Thresholding technique is a relative simple and fast technique. A survey of thresholding methods with a view to assess their performance when applied to remote

sensing images has been made recently (Marcello, 2004). Some image examples are taken from oceanographic applications in this work.

2. Surveys focused on a particular application of image segmentation:

Image segmentation has many applications. For each application, a number of segmentation algorithms could be developed. Some surveys for particular applications have been made. In medical imaging applications, image segmentation is used for automating or facilitating the delineation of anatomical structures and other regions of interest. A survey considering both semi-automated and automated methods for the segmentation of anatomical medical images has been done (Pham, 2000), wherein their advantages and disadvantages for medical imaging applications are discussed and compared.

While video could be considered as a particular type of general image, its segmentation is an extension of image segmentation. For video data, a temporal segmentation is used for determining the boundary of shots. A survey is made for techniques that operate on both uncompressed and compressed video streams (Koprinska, 2001). Both types of shot transitions, abrupt and gradual, are considered. The performance, relative merits and limitations of each approach are comprehensively discussed.

For temporal video segmentation, excepting the ability and correctness of shot detection, the computation complexity is also a criterion that should be considered, especially for real-time application. A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval has been made (Lefèvre, 2003). Depending on the information used to detect shot changes, algorithms based on pixel, histogram, block, feature and motion have been selected.

Vessel extraction is essentially a segmentation process. A survey for related algorithms to this process has been made (Kirbas, 2003). Six groups of techniques proposed for this particular application are involved: (1) pattern recognition techniques; (2) model-based approaches; (3) tracking-based approaches; (4) artificial intelligence-based approaches; (5) neural network-based approaches; and (6) miscellaneous tube-like object detection approaches.

In many vision applications, moving shadows must be detected. Moving shadows can be considered as objects in video streams and the detection of moving shadows is basically a video segmentation problem. A survey has been made for four classes of techniques (two statistical and two deterministic) that are specially designed for detecting moving shadows (Prati, 2003).

FUTURE TRENDS

The subject of image and video segmentation covers a very large area, and further developments could move in many directions; a few of them are indicated as follows:

1. Mathematical models and theories

It is said there is yet no general theory for image and video segmentation. However, this does not prevent the introduction of various mathematical theories into the research

of image and video segmentation. Many novel models have also been created over the years which have had certain success. To further push the research on image and video segmentation, and to drive the research beyond being *ad hoc* process, more mathematical models and theories would be required and used in the future.

2. High level study

As discussed in the beginning of this chapter, the research on image and video segmentation is currently conducted in three levels: the development of segmentation algorithms, the evaluation of segmentation quality and performance as well as the systematic study of evaluation methods. With a large number of segmentation algorithms being developed, the performance evaluation of these algorithms has attracted more research efforts (see Chapter XX). The results obtained from high-level study could greatly help the development of new segmentation algorithms and/or the effective utilization of the existing segmentation algorithms (Zhang, 2000).

3. Incorporating human factors

Image (and video) segmentation is a critical step of image analysis occupying the middle layer of image engineering, which means it is influenced not only from data but also from human factors. It seems that the assistance of humans, knowledgeable in the application domain, will remain essential in any practical image segmentation method. Incorporating high-level human knowledge algorithmically into the computer remains a challenge.

4. Application-oriented segmentation

Image and video segmentation have been proved necessary in many applications. Though the general process of segmentation is well defined in all applications, the particular requirements for segmentation can be different, and this difference leads to a variety of application-oriented segmentation. For example, in target detection, capturing a recognizable target, instead of segmenting it precisely, would be more significant. Another example is that the extraction of meaningful regions (Luo, 2001), instead of precisely segmenting objects, has proved to be effective in content-based visual information retrieval tasks (Zhang, 2005).

CONCLUSION

Image segmentation, forty years' old, is a critical task for image analysis which is at the middle layer of image engineering. The concepts of image and image segmentation have been extended widely since their initial appearance. The research on image and video segmentation is currently conducted at three different levels: developing segmentation algorithms, evaluating algorithm performance and studying the behavior of evaluation methods.

An overview of the development of image and video segmentation in the last 40 years is provided in this chapter. Several thousands of segmentation algorithms have

been designed and applied for various applications, and this number has increased steadily at a rate of several hundreds per year since 2000. This increase makes it quite hard to work out a comprehensive survey on the techniques of image and video segmentation, and a suitable classification scheme for hierarchical cataloging the whole technique.

After 40 years' growth, the domain of image and video segmentation is still immature. Many research topics, such as introducing more mathematical theories into this field, using high-level research results to guide low-level development, incorporating human factors and working toward application-oriented segmentation, need to be exploited. Even more, many unsettled problems need to be defined and solved in this area. However, as a Chinese saying states: "A person will not be puzzled after 40 years of age." Due to the accumulation of solid research results and the progress of science and technology comprising the 40 years' experience, further directions have become clearer. It is firmly believed that the domain of image and video segmentation will be greatly advanced in the future.

ACKNOWLEDGMENTS

This work has been supported by the Grants NNSF-60573148 and RFDP-20020003011.

REFERENCES

- Behiels, G., Maes, F., Vandermeulen, D., et al. (2002). Evaluation of image features and search strategies for segmentation of bone structures in radiographs using active shape models. *Medical Image Analysis*, 6(1), 47-62.
- Borisenko, V. I., Zlatotol, A. A., & Muchnik, I. B. (1987). Image segmentation (state of the art survey). *Automatic Remote Control*, 48, 837-879.
- Buf, J. M. H., & Campbell, T. G. (1990). A quantitative comparison of edge-preserving smoothing techniques. *Signal Processing*, 21, 289-301.
- Davis, L. S. (1975). A survey of edge detection techniques. *CGIP*, 4, 248-270.
- Freixenet, J., Munoz, X., Raba, D., et al. (2002). Yet another survey on image segmentation: Region and boundary information integration. In *ECCV*(pp. 408-422).
- Fu, K. S., & Mui, J. K. (1981). A survey on image segmentation. *Pattern Recognition*, 13, 3-16.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). NJ: Prentice Hall.
- Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *CVGIP*, 29, 100-132.
- Kirbas, C., & Quek, F. K. H. (2003). Vessel extraction techniques and algorithms: A survey. In *Proceedings of the 3rd Bioinformatics and Bioengineering Symposium* (pp. 238-245).
- Koprinska, I., & Carrato, S. (2001). Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5), 477-500.
- Lefèvre, S., Holler, J., & Vincent, N. (2003). A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. *Real-Time Imaging*, 9(1), 73-98.

- Luo, Y., Zhang, Y. J., Gao, Y. Y., et al. (2001). Extracting meaningful region for content-based retrieval of image and video. *SPIE*, 4310, 455-464.
- Marcello, J., Marques, F., & Eugenio, F. (2004). Evaluation of thresholding techniques applied to oceanographic remote sensing imagery. *SPIE*, 5573, 96-103.
- Medioni, G., Lee, M. S., & Tang, C. K. (2000). *A computational framework for segmentation and grouping*. New York: Elsevier.
- Munoz, X., Freixenet, J., Cufi, X., et al. (2002). Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters*, 23, 375-392.
- Nevatia, R. (1986). Image segmentation. In T.Y. Young & K.S. Fu (Eds.), *Handbook of pattern recognition and image processing* (Vol. 86, pp. 215-231). Orlando, FL: Academic Press.
- Olabarriaga, S. D., & Smeulders, A.W. M. (2001). Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2), 127-142.
- Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26, 1277-1294.
- Pavlidis, T. (1986). Critical survey of image analysis methods. In *Proceedings of the 8ICPR* (pp. 502-511).
- Peli, T., & Malah, D. (1982). A study of edge determination algorithms. *CGIP*, 20, 1-20.
- Pham, D., Xu, C., & Prince, J. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2, 315-337.
- Prati, A., Mikic, I., Trivedi, M. M., et al. (2003). Detecting moving shadows: Algorithms and evaluation. *IEEE PAMI*, 25(7), 918-923.
- Riseman, E. M., & Arbib, M. A. (1977). Computational techniques in the visual segmentation of static scenes. *CGIP*, 6, 221-276.
- Roberts, L. G. (1965). Machine perception of three-dimensional solids. In J. Tippett et al. (Eds.), *Optical and electro-optical information processing* (pp. 159-197).
- Roerdink, J., & Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamental Informatica*, 41, 187-228.
- Rosenfeld, A. (1981). Image pattern recognition. *Proceedings of IEEE*, 69(5), 596-605.
- Sahoo, P. K., Soltani, S., Wong, A.K.C., et al. (1988). A survey of thresholding techniques. *CVGIP*, 41, 233-260.
- Sarkar, S., & Boyer, K. L. (1993). Perceptual organization in computer vision: A review and a proposal for a classificatory structure. *IEEE SMC*, 23, 382-399.
- Weszka, J. S. (1978). A survey of threshold selection techniques. *CGIP*, 7, 259-265.
- Zhang, Y. J. (1995). Influence of segmentation over feature measurement. *Pattern Recognition Letters*, 16(2), 201-206.
- Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335-1346.
- Zhang, Y. J. (1997). Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18(10), 963-974.
- Zhang, Y. J. (2001a). *Image segmentation*. Beijing, China: Science Publisher.
- Zhang, Y. J. (2001b, August 13-16). A review of recent evaluation methods for image segmentation. In M. Deriche, N. Shaikh-Husin, & I. Kamisian (Eds.), *Proceedings of the 6th International Symposium on Signal Processing and Its Applications*, Lumpur, Malaysia (pp. 148-151). Piscataway, NJ: IEEE Computer Society.

- Zhang, Y. J. (2002a). Image engineering and related publications. *International Journal of Image and Graphics*, 2(3), 441-452.
- Zhang, Y. J. (2005). New advancements in image segmentation for CBIR. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (Vol. 4, pp. 2105-2109). Hershey, PA: Idea Group Reference.
- Zhang, Y. J. (2006). A study of image engineering. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (2nd ed.) [online].
- Zhang, Y. J., & Gerbrands, J. J. (1994). Objective and quantitative segmentation evaluation and comparison. *Signal Processing*, 39(3), 43-54.
- Zhang, Y. J., Gerbrands, J. J., & Back, E. (1990). Thresholding three-dimensional image. *SPIE*, 1360, 1258-1269.
- Zhang, Y. J., & Lu, H. B. (2002b). A hierarchical organization scheme for video data. *Pattern Recognition*, 35(11), 2381-2387.
- Zhang, Y. J., & Luo, H. T. (2000). Optimal selection of segmentation algorithms based on performance evaluation. *Optical Engineering*, 39(6), 1450-1456.
- Zucker, S. W. (1976). Region growing: Childhood and adolescence. *CGIP*, 5, 382-399.
- Zucker, S. W. (1977). Algorithms for image segmentation. In *Digital image processing and analysis* (pp. 169-183).

Section II:

Image Segmentation

Chapter II

Optimal Image Segmentation Methods Based on Energy Minimization

Francisco Escolano, Universidad de Alicante, Spain

Miguel A. Lozano, Universidad de Alicante, Spain

ABSTRACT

In this chapter we present three “case studies” as representative of recent work on solving several segmentation problems (region segmentation, deformable templates matching and grouping) from the energy minimization perspective. Each of the latter problems is solved via an optimization approach, respectively: jump-diffusion, belief propagation and Bayesian inference. Our purpose is to show the connection between the formulation of the corresponding cost function and the optimization algorithm and also to present some useful ideas coming from Bayesian and information theory. This selection of only three problems (and solutions) allows us to present the fundamental elements of optimization in each particular case and to bring the readers to the arena of optimization-based segmentation.

INTRODUCTION

From the energy minimization point of view (see www.emmcvpr.org for a recent series of workshops devoted to propose optimization methods for solving computer

vision and pattern recognition problems) segmentation involves both the design of cost functions and the proposal of optimization algorithms for minimizing them. In recent years, approaches in this regard (statistical mechanics methods, Bayesian inference, Markov random fields, belief propagation, PDEs and variational methods, information theory, and so on) have proved to be effective for solving many computer vision problems including region segmentation, clustering, perceptual grouping, contour segmentation and deformable templates fitting. Due to the methodological importance of energy minimization for designing a wide range of segmentation methods, we devote this chapter to describing a selection of optimization methods. We try to present clearly the connection between the models (cost functions) and the algorithms (simulated annealing, jump-diffusion, stochastic gradient descent, optimization via Lagrange multipliers, and so on). Due to space constraints it is impossible to provide here a detailed presentation of all the optimization methods and principles used for solving segmentation problems. However, there are some representative cases in the literature. We have selected three problems (region segmentation, deformable templates fitting and perceptual grouping) and three optimization methods (jump-diffusion, belief propagation and Bayesian A*) which are proved to solve respectively the latter problems with an acceptable computational cost. We present them here as “case studies.”

We begin by introducing a Bayesian inference formulation for the problem *region segmentation* (where the number of regions is unknown). Here we present existing recent work in solving the latter problem through a jump-diffusion technique. Such a particular approach is particularly interesting because it exploits data coming from bottom-up processes and thus reconciles generative approaches to segmentation with bottom-up ones. The second approach consists of solving a model-based segmentation problem, namely *deformable templates fitting*. In this latter case it is interesting to introduce recent advances in belief propagation methods for grasping global consistency by propagating local knowledge, in particular the connection between belief propagation rules and minimizing the Bethe free energy. Finally, in the last section of the chapter we will introduce a solution inspired in Bayesian inference to the problem of *feature extraction and grouping*. The type of cost functions described here for solving path-searching problems (finding contours in images) generalize the cost functions used in contour-based segmentation (snakes models) and allow the researchers to think about existing fundamental limits which constrain the possibility of even solving the problem independently of the algorithm chosen. The presented algorithm (Bayesian A*) is a good example of how to reduce the complexity of a segmentation task by exploiting information theory results.

REGION SEGMENTATION: METROPOLIS AND JUMP-DIFFUSION

Formulation of Bayesian Image Segmentation

Given an image $I(\mathbf{x})$ defined over a lattice S , the segmentation problem can be posed in terms of “finding a partition” of the lattice into an unknown number of regions K so that:

$$S = \bigcup_{i=1}^K R_i, \quad R_i \cap R_j = \emptyset, \quad \forall i \neq j \quad (1)$$

From a generative point of view, each image region $I_i(\mathbf{x})$, with $\mathbf{x} \in R_i$ is assumed to be the realization of a model encoded by a set of parameters Θ_i and indexed by ℓ_i , and this is quantified by the probability $P(I_i | \Theta_i, \ell_i)$. Then, from this point of view the purpose of segmentation is to estimate the vector \mathbf{W} which describes the world state for generating the image I , that is:

$$\mathbf{W} = (K, \{(R_i, \Theta_i, \ell_i) | i = 1, 2, \dots, K\}) \quad (2)$$

Defining the solution space as Ω , the maximum a posteriori (MAP) formulation of the segmentation problem consists of inferring:

$$\mathbf{W} \sim P(\mathbf{W} | I) \propto P(I | \mathbf{W})P(\mathbf{W}), \quad \mathbf{W} \in \Omega \quad (3)$$

where the first factor is the likelihood and the second one is the prior. Assuming independent stochastic processes for (Θ_i, ℓ_i) , the likelihood $P(I | \mathbf{W})$ may be given by the product of individual likelihoods with respect to each model, whereas the prior $P(\mathbf{W})$ is usually driven by the complexity of the model:

$$P(I | \mathbf{W}) = \prod_{i=1}^K p(I_i | \Theta_i, \ell_i), \quad P(\mathbf{W}) \propto P(K) \prod_{i=1}^K P(\ell_i)P(\Theta_i | \ell_i) \quad (4)$$

where the complexity relies on the number of models and the product of their compatibilities with the parameters chosen for each region.

Segmentation of a 1D Range Image

In order to get a simple example of the MAP formulation, let us concentrate on the problem of segmenting a 1D range signal $I(x) = I_o(x) + N(0, \sigma^2)$, with $x \in [0, 1]$, generated from the original $I_o(x)$ (see Figure 1). In this case, the generative approach (Han, Tu, & Zhu, 2004) states that the $S = X$ space (horizontal axis) must be partitioned into K intervals or regions $R_i = [x_{i-1}, x_i]$ and also that each surface $I_i(x)$ may fit either a line or a circular arc and this is indicated by the label ℓ_i . Consequently we have the following parameters and labels:

$$\Theta_1 = (s, \rho), \ell_1 = \text{line} \quad \text{and} \quad \Theta_2 = (x, y, r), \ell_2 = \text{circle} \quad (5)$$

where the line is defined by the two usual parameters for the Hough transform (orientation and distance to the origin) and the circle is given by the center coordinates and the radius.

As in Equation 2, the unknown world state (hidden variables) is given by:

$$\mathbf{W} = (K, \{([x_{i-1}, x_i], \ell_i, \Theta_i) | i=1,2,\dots,K\}) \quad (6)$$

In this 1D case, the individual likelihoods for each region $I_i(x)$ are assumed to decay exponentially with the squared error between the fragment of the observed signal corresponding to each interval and the hypothesized original signal for such interval, given the parameters of the model used to build the hypothesis:

$$P(I_i | \Theta_i, \ell_i) = \exp \left\{ -\frac{1}{2\sigma^2} \int_{x_{i-1}}^{x_i} (I(x) - I_0(x; \Theta_i, \ell_i))^2 dx \right\} \quad (7)$$

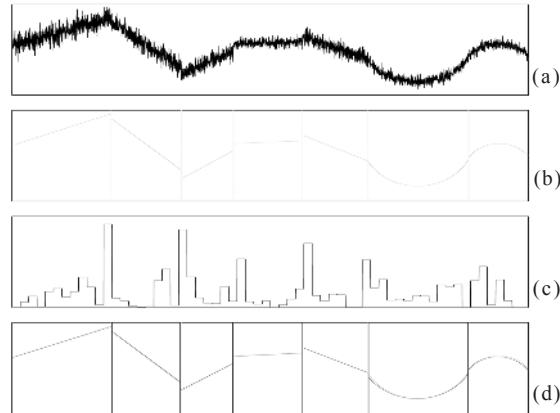
On the other hand, the prior $P(\mathbf{W})$ is given by penalizing the number of regions and the number of parameters:

$$P(K) \propto e^{-\lambda_0 K} \text{ and } P(\Theta_i, \ell_i) \propto e^{-\lambda_1 |\Theta_i|} \quad (8)$$

and also assuming that both models are equally likely *a priori*, that is, $P(\ell_i)$ is uniform in Equation 4. Then, we have the posterior:

$$P(\mathbf{W} | I) \propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^K \int_{x_{i-1}}^{x_i} (I(x) - I_0(x; \Theta_i, \ell_i))^2 dx \right\} \times \exp \{-\lambda_0 K\} \times \exp \left\{ -\lambda_1 \sum_{i=1}^K |\Theta_i| \right\} \quad (9)$$

Figure 1. The 1D range segmentation problem and its MAP solution



(a) Observed signal; (b) original signal; (c) edges, (d) comparison between MAP result (dark gray) and original signal before introducing noise (light gray)

Source: copyright IEEE 2004

whose exponent is the energy function $H(\mathbf{W}) = -\log P(\mathbf{W}|I)$:

$$H(\mathbf{W}) = \frac{1}{2\sigma^2} \sum_{i=1}^K \int_{x_{i-1}}^{x_i} (I(x) - I_0(x; \Theta_i, \ell_i))^2 dx + \lambda_0 K + \lambda_1 \sum_{i=1}^K |\Theta_i| \quad (10)$$

The MAP solution comes from maximizing the posterior (Equation 9) or, equivalently, by minimizing the energy. However, as the number of regions K is unknown, the energy function is defined over a space of variable dimension. We must find not only the optimal solution for a fixed K but the optimal value for such a parameter. We may either start for a high (low) value for K and then try to fuse (partition) regions in a greedy fashion, and according to a given metric. Alternatively, we may embody these two tasks in a unique optimization problem. In this latter case, what we need is a mechanism for “jumping” between spaces of different dimensions and another one for optimizing the energy within a given space. This is the motivation of “jump-diffusion.” In the following subsection we will describe the fundamentals of jump-diffusion in the general case and later we will return to the 1D range segmentation problem.

Fundamentals of Jump-Diffusion

A jump-diffusion strategy consists of iteratively solving an optimization problem by simulating two types of moves (Grenander & Miller, 1994; Green, 1995) (see also Robert & Casella, 1999):

- **Reversible jumps:** Moves between subspaces of different dimensions. These moves are driven by a Metropolis-Hastings sampler.
- **Stochastic diffusions:** Stochastic steepest descent (Langevin equations) within each continuous subspace.

Let us define first the rules driving reversible jumps. Let $\mathbf{x} = (k, \boldsymbol{\theta}^{(k)})$ and $\mathbf{x}' = (l, \boldsymbol{\theta}^{(l)})$, two states corresponding to subspaces Ω_k, Ω_l and let \mathbf{y} be an observation. In order to ensure reversibility, the existence of the bijection $(\boldsymbol{\theta}^{(l)}, \mathbf{u}_l) = T(\boldsymbol{\theta}^{(k)}, \mathbf{u}_k)$ is assumed, which represents that $\boldsymbol{\theta}^{(l)}$ will be completed, if necessary, with \mathbf{u}_l and $\boldsymbol{\theta}^{(k)}$, with \mathbf{u}_k to ensure that $T(\cdot)$ is bijective (“dimension matching”). For instance, we may be interested in moving from $\mathbf{x} = (1, \boldsymbol{\theta})$ and $\mathbf{x}' = (2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. Then, for the move $\mathbf{x}' \rightarrow \mathbf{x}$ we may set $\boldsymbol{\theta} = (\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2)/2$, and for the move $\mathbf{x} \rightarrow \mathbf{x}'$ we may set $\boldsymbol{\theta}_1 = \boldsymbol{\theta} - \mathbf{u}$ and $\boldsymbol{\theta}_2 = \boldsymbol{\theta} + \mathbf{u}$ with \mathbf{u} being a random variable. Then, we have the following bijections:

$$\left(\underbrace{\frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2}{2}, \boldsymbol{\theta}_2}_{T_{11}}, \underbrace{\boldsymbol{\theta}_1}_{T_{12}} \right) = T_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \quad \text{and} \quad \left(\underbrace{\boldsymbol{\theta} - \mathbf{u}}_{T_{21}}, \underbrace{\boldsymbol{\theta} + \mathbf{u}}_{T_{22}} \right) = T_2(\boldsymbol{\theta}, \mathbf{u}) \quad (11)$$

Once bijection is ensured, the next step is to define, for a given jump $\mathbf{x} \rightarrow \mathbf{x}'$, the probability of accepting it $\alpha(\mathbf{x}'|\mathbf{x})$; is given by:

$$\alpha(\mathbf{x}'|\mathbf{x}) = \min \left\{ 1, \frac{P(\mathbf{x}'|\mathbf{y})Q(\mathbf{x} \rightarrow \mathbf{x}')Q_r(\mathbf{u}_r) \left| \frac{\partial T(\boldsymbol{\theta}^{(k)}, \mathbf{u}_k)}{\partial(\boldsymbol{\theta}^{(k)}, \mathbf{u}_k)} \right|}{P(\mathbf{x}|\mathbf{y})Q(\mathbf{x}' \rightarrow \mathbf{x})Q_k(\mathbf{u}_k)} \right\} \quad (12)$$

where $Q(\cdot)$ is the probability of choosing the destination subspace from the current one, $Q_r(\cdot)$ is the density of \mathbf{u}_r , and the last factor is the determinant of the corresponding Jacobian matrix. For instance, for the numerical example shown above we have the Jacobians (see Equation 11):

$$\frac{\partial T_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} = \begin{bmatrix} \frac{\partial T_{11}}{\partial \boldsymbol{\theta}_1} & \frac{\partial T_{11}}{\partial \boldsymbol{\theta}_2} \\ \frac{\partial T_{12}}{\partial \boldsymbol{\theta}_1} & \frac{\partial T_{12}}{\partial \boldsymbol{\theta}_2} \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 0 & 1 \end{bmatrix}, \quad \frac{\partial T_2(\boldsymbol{\theta}, u)}{\partial(\boldsymbol{\theta}, u)} = \begin{bmatrix} \frac{\partial T_{21}}{\partial \boldsymbol{\theta}} & \frac{\partial T_{21}}{\partial u} \\ \frac{\partial T_{22}}{\partial \boldsymbol{\theta}} & \frac{\partial T_{22}}{\partial u} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (13)$$

and their corresponding determinants:

$$\left| \frac{\partial T_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \right| = \frac{1}{2} \text{ and } \left| \frac{\partial T_2(\boldsymbol{\theta}, u)}{\partial(\boldsymbol{\theta}, u)} \right| = 2 \quad (14)$$

From the point of view of function optimization, the Metropolis-Hastings algorithm uses, in general, the acceptance rule (like the one defined in Equation 12) to induce a irreducible Markov chain, that is, a random walk producing a sequence of states that may visit, at least in theory, the complete search space. In the context of jump diffusion this means that all jumps are possible and thus potentially traversing the complete space Ω is possible.

Once we have reviewed the “jumping” side of the coin let us concentrate on the “diffusion” side. The underlying idea is to move within a given space (or subspace in our case) by minimizing a given energy function $E: \mathbf{x} \rightarrow \Omega$. This role may be performed by the following stochastic differential equation, the so called “diffusion equation” build as follows (Grenander & Miller, 1994; Robert & Casella, 1999):

$$d\mathbf{x} = \frac{1}{2} \nabla \log P(\mathbf{x}) dt + d\mathbf{B} \quad (15)$$

where $P(\mathbf{x})$ is a probability function (for instance, when $P(\mathbf{x}) \propto \exp(-H(\mathbf{x})/T)$), $\nabla \log P(\mathbf{x}) = \nabla(-H(\mathbf{x})/T)$ is the gradient of the energy function weighted by the temperature parameter T , and \mathbf{B} is the standard Brownian motion (the random part of the

equation), that is $d\mathbf{B} \sim N(0, \omega^2 |dt|)$ being $\mathbf{B}_0 = 0$, $\mathbf{B}_t \sim N(0, \omega^2 t)$ and $d\mathbf{B}_t = \mathbf{B}_t - \mathbf{B}_{t'}$ independent from $\mathbf{B}_{t'}$ for $t' > t$. Discretizing the Equation 15 we have:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \frac{\sigma^2}{2} \nabla \log P(\mathbf{x}^{(t)}) + \sigma \mathbf{e}^{(t)}, \mathbf{e}^{(t)} \sim N_p(0, \mathbf{I}_p), p = |x| \quad (16)$$

where \mathbf{I}_p is the identity matrix of dimension p . In other words, the diffusion equation is similar to a stochastic gradient descent where the amplitude of the distortion is controlled by σ . Furthermore, let us use a variable amplitude $\sigma(t)$ so that $\sigma(t) = \sqrt{2T(t)}$, where $T(t)$ is the value of the temperature parameter at time t satisfying: $T(t) \rightarrow 0$ when $t \rightarrow \infty$. This means that the stochastic distortion is greater at the beginning of the process (in order to keep the search away from local optima) than at the latest stages when the search is supposed to be caught in the basin of the global minimum. Consequently, Equations 15 and 16 are rewritten as follows:

$$d\mathbf{x} = -\frac{dH(\mathbf{x})}{d\mathbf{x}} dt + \sqrt{2T(t)} d\mathbf{B}_t, \quad d\mathbf{B}_t \sim N_p(0, \mathbf{I}_p(dt)^2) \quad (17)$$

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \frac{dH(\mathbf{x}_t)}{d\mathbf{x}_t} dt + \sqrt{2T(t)} \mathbf{e}^{(t)}, \mathbf{e}^{(t)} \sim N_p(0, \mathbf{I}_p), p = |x| \quad (18)$$

The process described above implements the so called “Langevin diffusion.” From the optimization point of view, the key fact about such a diffusion is that, for a fixed temperature $T(t)$ it implements a stochastic process whose stationary distribution is $P(\mathbf{x})$; in our particular case we have that $P(\mathbf{x}) \propto \exp(-H(\mathbf{x})/T(t))$. Consequently, in the limit $T(t) \rightarrow 0$ we have that the stochastic process yields samples of the Gibbs distribution $\pi(\mathbf{x}) = Z^{-1} \exp(-H(\mathbf{x}))$ being $Z = \sum_{\mathbf{x} \in \Omega} \exp(-H(\mathbf{x}))$, the normalization factor called “partition function.” This means that in the limit, the stochastic process converges, with uniform probability, to one of the global minimizers of H .

Jump-Diffusion for the 1D Range Problem

Adapting the jump-diffusion method to the model-based segmentation of the 1D range profile requires, once the energy function is defined, the specification of both types of “jump” we are going to consider and the Langevin equations.

In order to specify the jumps we have that $\mathbf{W} = (n, \psi)$ is the state of the stochastic process at time t being $\psi \in \Omega_n$ and the state variables (points defining the intervals and $(m \neq n)$ is the destination state of the jump (a jump always implies a change of space). Then, the acceptance probability is given by:

$$\alpha(\mathbf{W}' | \mathbf{W}) = \min \left\{ 1, \frac{P(\mathbf{W}' | I) d\phi}{P(\mathbf{W} | I) d\psi} \times \frac{G(\mathbf{W}' \rightarrow \mathbf{W}) d\psi}{G(\mathbf{W} \rightarrow \mathbf{W}') d\phi} \right\} \quad (19)$$

Where $G(\cdot)$ denotes the “proposal probabilities,” that is, the product of the probability of choosing the destination space and the density of the parameters in the destination space:

$$\begin{aligned} G(\mathbf{W}' \rightarrow \mathbf{W}) &= Q(\mathbf{W}' \rightarrow \mathbf{W}) Q(\psi | n) \\ G(\mathbf{W} \rightarrow \mathbf{W}') &= Q(\mathbf{W} \rightarrow \mathbf{W}') Q(\phi | m) \end{aligned} \quad (20)$$

However, the (standard) jumping scheme has an important drawback: It is too slow for being useful in practice. The densities $Q(\psi | n)$ and $Q(\phi | m)$ are usually set to the uniform distributions. This means, for instance, that we will select randomly between circles and lines when we want to choose a model. Furthermore, as the ratio $P(W' | I) / P(W | I) = e^{-\Delta H}$ may be close to zero, in these cases the jump proposals will be rejected. Under these conditions, the stochastic search progresses too slowly. This situation has motivated the development of a novel data-driven approach (Tu & Zhu, 2002; Han, Tu, & Zhu, 2004) for re-designing the proposal probabilities. Following this approach, let us first consider the three types of jumps that we may have: (1) Changing of model type; (2) Merging two adjacent intervals to build a new one with a given model; and (3) Splitting a given interval between two adjacent ones having new models. Each of these situations implies a redefinition of the destination $Q(\phi | m)$ proposal:

$$Q(\phi | m) = \begin{cases} Q(\theta_i | \ell_i, [x_{i-1}, x_i]) & (21.1) \\ Q(\theta | \ell, [x_{i-2}, x_i]) & (21.2) \\ Q(x | [x_{i-1}, x_i]) Q(\theta_a | \ell_a, [x_{i-1}, x]) Q(\theta_b | \ell_b, [x, x_i]) & (21.3) \end{cases}$$

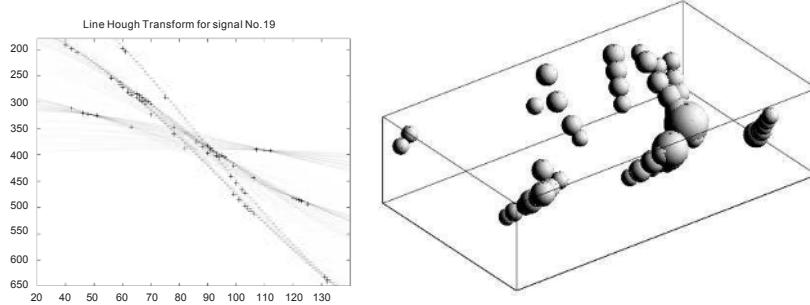
Equation 21.1 defines the probability of switching $[x_{i-2}, x_i]$ to model (ℓ_i, θ_i) ; Equation 21.2 defines the probability of merging intervals $[x_{i-2}, x_{i-1}]$ and $[x_{i-1}, x_i]$ and choosing model (ℓ, θ) for the result; and (21.3) defines the probability of selecting point $x \in [x_{i-1}, x_i]$ to split the interval into and with models and respectively.

In order to compute the proposal probabilities, the data-driven approach relies on bottom-up processes which, in this case, compute the Hough transforms of lines and circles. Here, the original generative approach which has a top-down nature is clearly complemented by bottom-up computations which are key to accelerate the search.

Given the Hough transforms of points and lines (see Figure 2), the Parzen windows technique is used to build the densities of the proposal probabilities. For instance:

$$Q(\theta | \ell, [a, b]) = \sum_{i=1}^{N_\ell} \omega_i G(\theta - \theta_\ell^{(i)}), \ell \in \{\text{line, arc}\} \quad (22)$$

Figure 2. Hough transforms for lines (left) and circles (right). In the case of lines, the crosses indicate line candidates; in the case of circles the size of the balls represents the number of votes received by the center of the ball representing a potential arc. In both cases, the axes represent each of the parameters of the corresponding 2D (for lines) and 3D (for circles) parameter space.



Source: copyright IEEE 2004

where $G(\theta - \theta_l^{(i)})$ denotes the density of a Gaussian kernel centered at a candidate line or circle $\theta_l^{(i)}$ and w_i denotes the weights (number of votes). Then, Equation 21 quantifies the probability of choosing a given model for a given interval.

Besides the probability defined in Equation 22, we need to define the probability of choosing a given point in the interval in order to split it. In this latter case, and following again a bottom-up approach, it seems reasonable to use potential edges as candidates. In order to do so, an “edgeness” measure $f(x | \nabla G * I, \nabla^2 G * I)$ (relying in the two first Gaussian derivatives) is used (later in this chapter we will introduce and exploit alternative, and probabilistic, measures of edgeness). We define the normalized measure as:

$$Q(x|[a,b]) = \frac{f(x | \nabla G * I, \nabla^2 G * I)}{\int_a^b f(x | \nabla G * I, \nabla^2 G * I) dx} \quad (23)$$

Once we have defined the elements to implement the “jumps,” let us introduce the elements for the “diffusion.” Given the energy defined in Equation 10, we have that:

$$E(W) = \frac{1}{2\sigma^2} \sum_{i=1}^K \int_{x_{i-1}}^{x_i} (I(x) - I_0(x; \Theta_i, \ell_i))^2 dx + const \quad (24)$$

and then the stochastic (Langevin) diffusion equations (Equation 16) for moving each point x_i are given by:

$$\frac{dx_i(t)}{dt} = \frac{1}{2\sigma^2} [(I(x_i) - I_0(x_i; \ell_{i+1}, \theta_{i+1}))^2 - (I(x_i) - I_0(x_i; \ell_i, \theta_i))^2] + \sqrt{2T(t)}N(0,1) \quad (25)$$

where the equilibrium of such point is found when the cost of fitting its surface value to the model at its right is equal to the cost of fitting it to the model at its left.

Finally, an important question to consider is how to interleave jumps and diffusions. If we jump very frequently we do not allow the search to be stabilized. If we do it from time to time we will obtain good segmentations but for the wrong K . It has been suggested to interrupt continuous diffusions by jumps when a Poisson event arises (Han, Tu, & Zhu, 2004). Then, if a diffusion is performed at each Δt , the discrete waiting time τ_j between two consecutive jumps is given by:

$$\varpi = \frac{t_{j+1} - t_j}{\Delta t} \sim p(\varpi) = e^{-\tau} \frac{\tau^\varpi}{\varpi!}, \quad (26)$$

that is, $\varpi = \tau$ is the frequency of jumps, and both diffusions and jumps are controlled by a temperature annealing scheme for implementing the temperature lowering with time. In Figure 3, the comparison between three versions of this scheme is shown: MCMCI (original jump-diffusion method, not data driven), MCMCII (uses only data coming from Hough transform, not from measuring edgeness) and MCMCIII (the complete model). The MCMCII or MCMCIII algorithm is summarized in Figure 4. As shown in Figure 3 (left), the algorithm progresses by alternating continuous diffusions and discontinuities due to jumps, and this is why the energy does not fall monotonically.

We have illustrated the data driven jump-diffusion method for the simple case of segmenting 1D range images. However the same methodology was applied in Tu and Zhu (2002) for segmenting grey images. In this latter case, both the models and the jump models are more complex, and thus the cooperation between bottom-up and top-down processes is critical.

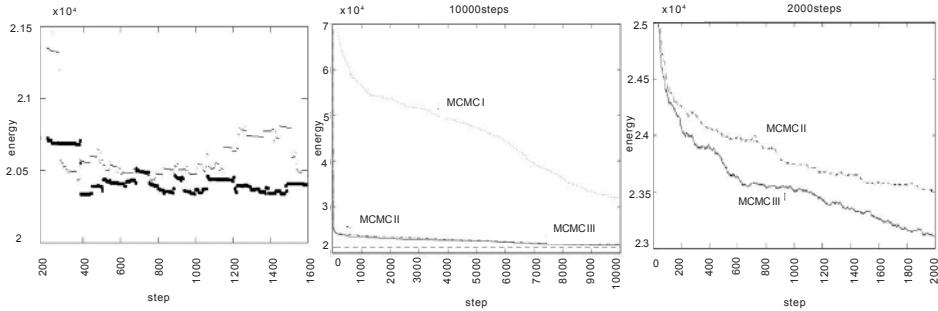
DEFORMABLE TEMPLATES DRIVEN BY BELIEF PROPAGATION

As an example of model-based segmentation, we will show an approach to the recognition of flexible objects proposed by Coughlan (Coughlan & Ferreira, 2002), using deformable templates and a belief propagation (BP) process as inference algorithm. The problem to solve is finding 2D shapes on cluttered greyscale images.

The model for the templates is defined probabilistically. It is a Bayesian deformable template, defined as a Bayesian graphical model of the 2D shape contour that takes into account the variability of the shapes and their intensity. The flexibility of this model yields a hard computational problem, but it is necessary for finding objects, and even edges, in complex images. This model will be invariant to global translations and rotations.

For matching each part of the template to a location in the image efficiently, it can be used as an inference procedure based on BP. This method ensures convergence to

Figure 3. Energy plots for jump-diffusion for MCMCII (thin curve) and MCMCIII (left). Energy curves for the 10.000 steps (center). Zooming showing 2.000 steps (right).



Source: copyright IEEE 2004

Figure 4. Jump-diffusion algorithm for region segmentation (1D problem). The variable “jump” registers the occurrence of a jumping event which follows a Poisson distribution (Equation 26).

```

JUPM-DIFFUSION ALGORITHM: MCMCII-III

INITIALIZE  $\mathbf{W}$  randomly, temperature  $T \leftarrow T_0, t = 0$  ;
WHILE ( $T > T_{FINAL}$ ) OR (convergence)
  WHILE NOT (jump)
    Diffusion for all change points  $x_i$ : Eq. (25)
    
$$\frac{dx_i(t)}{dt} = \frac{1}{2\sigma^2} [(I(x_i) - I_0(x_i; \ell_{i+1}, \theta_{i+1}))^2 - (I(x_i) - I_0(x_i; \ell_i, \theta_i))^2] + \sqrt{2T} N(0,1)$$

    Update  $\mathbf{W}$  with new change points.
    Update temperature:  $t = t + 1, T \leftarrow T(t)$ 
  END
  IF jump
    Generate  $\mathbf{W}'$  as a random perturbation of  $\mathbf{W}$ 
    Data driven: Compute proposals  $Q(\phi | m)$ : Eq. (21)
    Compute acceptance probability: Eq. (19)
    
$$\alpha(\mathbf{W}' | \mathbf{W}) = \min \left\{ 1, \frac{P(\mathbf{W}' | I) d\phi}{P(\mathbf{W} | I) d\psi} \times \frac{G(\mathbf{W}' \rightarrow \mathbf{W}) d\psi}{G(\mathbf{W} \rightarrow \mathbf{W}') d\phi} \right\}$$

    IF acceptance  $\mathbf{W}' \leftarrow \mathbf{W}$ 
  END
END

```

the optimal solution if the graphical model is tree-shaped, but convergence is not ensured if there are loops in the graphical model. Researchers have found empirically that BP converges for a variety of models with loops (Murphy, Weiss, & Jordan, 1999). BP has an advantage over dynamic programming that it may be applied with loops.

Deformable Template

The deformable template is defined as a Bayesian graphical model, composed by a shape prior and an imaging model. The shape prior is a graphical model describing probabilistically the allowed shape configurations, and the imaging model describes how any configuration appears in a greyscale image. Given an image, this Bayesian model assigns a posterior probability to each possible configuration. We will see later how to use an inference algorithm based on BP for finding the most likely configuration.

The shape prior models the variability of the template; it assigns a probability to each possible configuration. The shape is defined as a set of points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ that represent its contour, and normal vectors $\theta_1, \theta_2, \dots, \theta_N$ corresponding to each point of the contour. A configuration of the template is defined as $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N)$, where each node is defined by $\mathbf{q}_i = (\mathbf{x}_i, \theta_i)$ and $\mathbf{x}_i = (x_i, y_i)$. We show an example of a template for “A” character in Figure 5 (left).

For assigning probabilities to each configuration, we have a reference configuration $\tilde{\mathbf{Q}} = (\tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \dots, \tilde{\mathbf{q}}_N)$ of the template. The more similar a configuration is to the reference configuration $\tilde{\mathbf{Q}}$, the higher the probability that it will have this configuration. A Markov Random Field will be used as graphical model, as shown in Figure 5 (right), in order to penalize the deviation between \mathbf{Q} and $\tilde{\mathbf{Q}}$. This model will be invariant to global rotation and translation, but it will have a fixed scale, which is supposed to be known beforehand.

The deviations are measured by the geometric relationship of a pair of nodes $\mathbf{q}_i, \mathbf{q}_j$ expressed in terms of interaction energies $U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$, which will be low for highly probable configurations. Two types of similarities are considered in order to obtain the energy: orientation and location similarities.

In order to make this model invariant to global translation and rotation, the similarities are computed within a local coordinate system defined by \mathbf{Q} and $\tilde{\mathbf{Q}}$.

Relative orientation between θ_i and θ_{i+1} should be similar to the relative orientation between $\tilde{\theta}_i$ and $\tilde{\theta}_{i+1}$. This similarity is modelled by the following term:

$$U_{ij}^C(\mathbf{q}_i, \mathbf{q}_j) = \sin^2\left(\frac{\theta_j - \theta_i - C_{ij}}{2}\right), \quad \text{where } C_{ij} = \tilde{\theta}_j - \tilde{\theta}_i \quad (27)$$

For translation, it's desirable that the positions of \mathbf{x}_j relative to \mathbf{q}_i were similar to the position of $\tilde{\mathbf{x}}_j$ relative to $\tilde{\mathbf{q}}_i$. Being that $\mathbf{n}_i, \tilde{\mathbf{n}}_i$ the normal vectors of the corresponding nodes of the template, and $\mathbf{n}_i, \tilde{\mathbf{n}}_i$ the perpendicular vector to the first ones, the dot product $(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i$ should be similar to the dot product $(\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \mathbf{n}_i$, and the same occurs with the perpendicular vectors. It is modelled in the following terms:

Figure 5. Template for “A” character. Left: Points of the template and normal vectors. Right: Connections between neighbouring points.

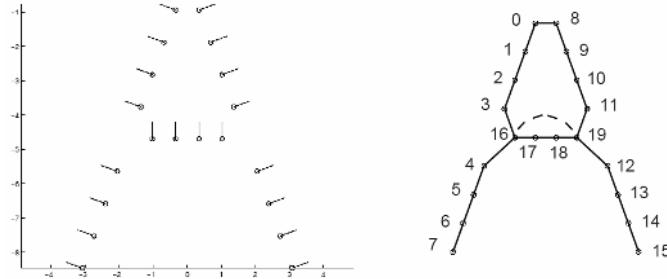
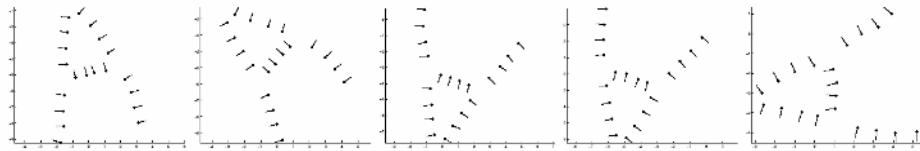


Figure 6. Variability of the “A” template



$$\begin{aligned} U_{ij}^A(\mathbf{q}_i, \mathbf{q}_j) &= [(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i - (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \mathbf{n}_i]^2 \\ U_{ij}^B(\mathbf{q}_i, \mathbf{q}_j) &= [(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i^\perp - (\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i) \cdot \mathbf{n}_i^\perp]^2 \end{aligned} \quad (28)$$

Then, the full interaction energy can be defined as:

$$U_{ij}(\mathbf{q}_i, \mathbf{q}_j) = \frac{1}{2} \{ K_{ij}^A U_{ij}^A(\mathbf{q}_i, \mathbf{q}_j) + K_{ij}^B U_{ij}^B(\mathbf{q}_i, \mathbf{q}_j) + K_{ij}^C U_{ij}^C(\mathbf{q}_i, \mathbf{q}_j) \} \quad (29)$$

The K coefficients define the strengths of the interactions, and $K_{ij} = 0$ for the pairs of nodes $\mathbf{q}_i, \mathbf{q}_j$ with no interaction. Higher values for these coefficients yield less deformable templates. These values can be chosen experimentally from stochastic samples (Figure 6).

The interaction energy can be symmetrized as $U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j) = U_{ij}(\mathbf{q}_i, \mathbf{q}_j) + U_{ji}(\mathbf{q}_j, \mathbf{q}_i)$. The shape prior if obtained from this symmetrized energy is as follows:

$$P(Q) = \frac{1}{Z} \prod_{i < j} e^{-U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j)} \quad (30)$$

In this equation Z is a normalization constant, and only $i < j$ nodes are considered for eliminating double-counting.

The image data $d(\mathbf{x})$, extracted from the raw greyscale image, consists of an edge map $I_e(\mathbf{x})$ and an orientation map $\phi(\mathbf{x})$. The edge map provides local evidence for the presence of an object boundary, and the orientation map indicates the orientation of these edges for each pixel of the image. In order to obtain this information, we can use the following filter based on a gradient operator:

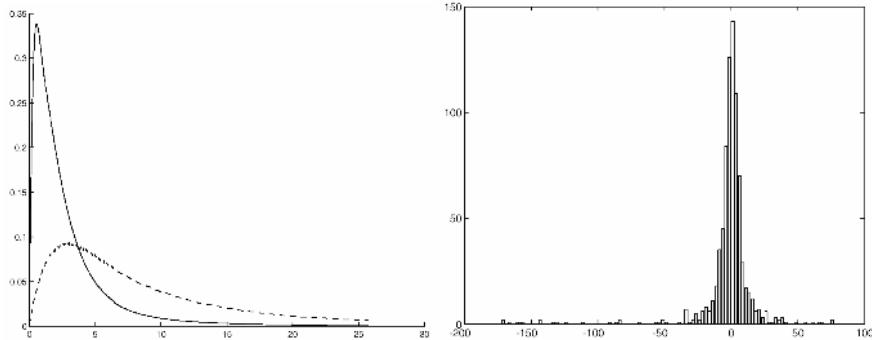
$$\begin{aligned} I_e(\mathbf{x}) &= |G * \nabla I(\mathbf{x})| \\ \phi(\mathbf{x}) &= \arctan(g_x / g_y), \text{ where } (g_x, g_y) = G * \nabla I(\mathbf{x}) \\ d(\mathbf{x}) &= (I_e(\mathbf{x}), \phi(\mathbf{x})) \end{aligned} \quad (31)$$

The likelihood model quantifies the edge strength to be either an ON or an OFF edge. If it's known beforehand whether a pixel is part of an edge (ON) or not (OFF), the probability of this pixel having the different possible edge strengths can be modelled as follows (see Figure 7):

$$\begin{aligned} P_{on}(I_e) &= P(I_e | \mathbf{x} \text{ ON edge}) \\ P_{off}(I_e) &= P(I_e | \mathbf{x} \text{ OFF edge}) \end{aligned} \quad (32)$$

Assuming that θ is the true normal orientation of a boundary, we can expect that the observed orientation of the edge in the image $\phi(\mathbf{x})$ is similar to either θ or $\theta + \pi$. It can be modelled as $P_{ang}(\phi | \theta)$, being this distribution periodic, with period π . For all pixels that are not edges, this distribution will be uniform, and it can be modelled as $U(\phi) = \frac{1}{2\pi}$. Therefore, the ON probability, if factorized into two terms, one of them dependent on the gradient strength and the other one on the divergence between the real and the estimated orientation, is:

Figure 7. Left: Example of probability distribution for $P_{on}(I_e)$ (dashed line) and $P_{off}(I_e)$ (solid line) (Right: Example of probability distribution for $P_{ang}(\phi | \theta)$)



$$P_{on}(d(\mathbf{x}) | \mathbf{q}_1 \dots \mathbf{q}_n) = P_{on}(I_e(\mathbf{x})) P_{ang}(\phi(\mathbf{x}) - \theta_i) \quad (33)$$

On the other hand, OFF probability only depends on gradient strength, assuming a uniform distribution for the orientation:

$$P_{off}(d(\mathbf{x}) | \mathbf{q}_1 \dots \mathbf{q}_n) = P_{off}(I_e(\mathbf{x})) U(\phi(\mathbf{x})) \quad (34)$$

Then, the likelihood model is defined as:

$$\begin{aligned} P(D | Q) &= \prod_{\text{all pixels } \mathbf{x}} P(d(\mathbf{x}) | \mathbf{q}_1 \dots \mathbf{q}_n) \\ P(d(\mathbf{x}) | \mathbf{q}_1 \dots \mathbf{q}_n) &= \begin{cases} P_{on}(I_e(\mathbf{x})) P_{ang}(\phi(\mathbf{x}) - \theta_i) & \text{if } \exists i : \mathbf{x} = \mathbf{x}_i, q_i = (\mathbf{x}_i, \theta_i) \\ P_{off}(I_e(\mathbf{x})) U(\phi(\mathbf{x})) & \text{otherwise} \end{cases} \end{aligned} \quad (35)$$

The posterior distribution will be:

$$P(Q | D) = \frac{P(Q) P(D | Q)}{P(D)} \quad (36)$$

Considering that there is not any and mapping to the same point in the image we have:

$$\begin{aligned} P(D | Q) &= \prod_{i=1}^N \frac{P_{on}(I_e(\mathbf{x}_i)) P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{P_{off}(I_e(\mathbf{x}_i)) U(\phi(\mathbf{x}_i))} \prod_{\text{all pixels}} P_{off}(I_e(\mathbf{x})) U(\phi(\mathbf{x})) \\ P(Q | D) &\propto P(Q) \prod_{i=1}^N \frac{P_{on}(I_e(\mathbf{x}_i))}{P_{off}(I_e(\mathbf{x}_i))} \frac{P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{U(\phi(\mathbf{x}_i))} \\ P(Q | D) &= \frac{1}{Z} \prod_i \psi_i(\mathbf{q}_i) \prod_{i < j} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \\ \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) &= e^{-U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j)} \\ \psi_i(\mathbf{q}_i) &= \frac{P_{on}(I_e(\mathbf{x}_i)) P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{P_{off}(I_e(\mathbf{x}_i)) U(\phi(\mathbf{x}_i))} \end{aligned} \quad (37)$$

We can calculate MAP (maximum a posterior) for obtaining the solution for our problem:

$$Q^* = \arg \max_Q P(Q | D) \quad (38)$$

But for this problem calculating the MAP marginal will be enough:

$$\begin{aligned}\mathbf{q}_i^* &= \arg \max_{\mathbf{q}_i} P(\mathbf{q}_i | D) \\ P(\mathbf{q}_i | D) &= \sum_{\mathbf{q}_1} \sum_{\mathbf{q}_{21}} \dots \sum_{\mathbf{q}_{i-1}} \sum_{\mathbf{q}_{i+1}} \dots \sum_{\mathbf{q}_N} P(Q | D)\end{aligned}\quad (39)$$

where the optimal state \mathbf{q}_i^* for a given point in the template has associated a pixel in the image where we are searching the template. Anyway, computing the marginal as described in Equation 39 is too expensive. An alternative is to use BP to compute an approximation to the marginal probabilities in time linear with number of nodes.

Belief Propagation

The template is represented with a MRF, and we want to obtain the marginal probabilities of the unknown variables \mathbf{q}_i (hidden states of the MRF). We can use BP (Yedidia, Freeman, & Weiss, 2002) for this purpose. If the MRF contains no loops, BP guarantees to find a correct solution. In certain domains, BP also works successfully with loops.

BP needs discrete variables, therefore we must quantize Q so that each \mathbf{q}_i belongs to a finite state space S . In the particular case of our 2D templates, this space will be the set of triples (x, y, θ) , where (x, y) can be the locations of the pixels with high enough edge strength on image lattice, and $\theta \in [0, 2\pi]$ can be $\phi(x, y)$ or $\phi(x, y) + \pi$ for each pixel (x, y) .

In order to calculate marginal probabilities, BP introduces message variables $m_{ij}(\mathbf{q}_i)$. The variable $m_{ij}(\mathbf{q}_i)$ will be a message sent from node i to node j , indicating the degree with which node i believes that node j should be in state \mathbf{q}_j . Given these message variables, the belief (estimation of the marginal $P(\mathbf{q}_i | D)$) of each node is computed as follows:

$$b_i(\mathbf{q}_i) = k \psi_i(\mathbf{q}_i) \prod_{j \in N(i)} m_{ji}(\mathbf{q}_i) \quad (40)$$

We can see that the belief for a node i is proportional to the product of the local evidence $\psi_i(\mathbf{q}_i)$ and all incoming messages, being k a normalization constant and $N(i)$, the nodes neighboring i , excluding i , as we can see in Figure 8 (left).

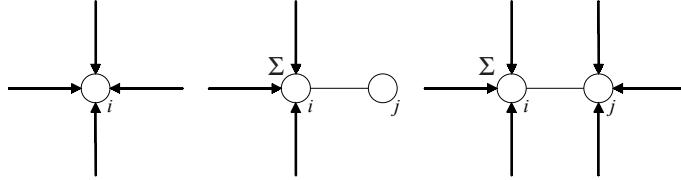
The messages are determined by the following rules:

$$m_{ij}(\mathbf{q}_j) = \sum_{\mathbf{q}_i} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \psi_i(\mathbf{q}_i) \prod_{k \in N(i) - j} m_{ki}(\mathbf{q}_i) \quad (41)$$

For computing the message to be sent from node i to j , we take the product over all messages going into node i , except the one coming from j , as is shown in Figure 8 (center).

These rules give exact beliefs for singly-connected MRF, that is, with no loops. In practical computation, it is possible to start with the nodes in the edge of the graph. In this case, a message is only computed when one node has available all messages necessary, that is, the messages from all neighboring nodes except the node we will send the message to. Therefore, it is convenient to start with nodes with only one connection.

Figure 8. Left: A diagrammatic representation of the BP belief Equation 40. Center: A diagrammatic representation of the BP message update rule, Equation 41. Right: A diagrammatic representation of the BP two-node belief Equation 44.



The whole computation takes a time period proportional to the number of links in the graph. This algorithm organizes “global” computation of marginal beliefs in terms of smaller local computations.

It’s important that there were no loops in the MRF, because in a loop all nodes have at least two connections, and it’s impossible to begin passing messages inside it with our procedure.

BP is defined in terms of belief equations and message update rules. It does not make reference to the topology of the graph. Then it can be implemented on a graph with loops. In this case it is possible to start with some initial set of uniform messages and iterate propagating messages, but these is not guaranteed to converge. With this procedure only approximate beliefs are ensured. In certain domains, it is empirically demonstrated that BP succeeds, but in principle we do not expect the algorithm to work well in graphs with loops.

In template-matching problems, in which there are loops, the messages are computed as follows:

$$\begin{aligned} m_{ij}^{(t+1)}(\mathbf{q}_j) &= \frac{1}{Z_{ij}} \sum_{\mathbf{q}_i} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \psi_i(\mathbf{q}_i) \prod_{k \in N(i) \sim j} m_{ki}^{(t)}(\mathbf{q}_i) \\ Z_{ij} &= \sum_{\mathbf{q}_i} m_{ij}^{(t+1)}(\mathbf{q}_i) \end{aligned} \quad (42)$$

where Z is a normalization factor, and (t) , $(t+1)$ are discrete time indices. Messages $m_{ij}^{(0)}(\mathbf{q}_i)$ are initialized to uniform values.

Updates can be done in parallel for all connected pairs of nodes i, j , or different subsets of pairs can be updated at different times in the case of the proposed focused message update.

The belief variable $b_i(\mathbf{q}_i)$ estimates marginal $P(\mathbf{q}_i | D)$, derived from message variables:

$$b_i(\mathbf{q}_i) = \frac{1}{Z'_i} \psi_i(\mathbf{q}_i) \prod_{k \in N(i)} m_{ki}(\mathbf{q}_i) \quad (43)$$

where Z'_i is a normalization factor for making $\sum_{\mathbf{q}_i} b_i(\mathbf{q}_i) = 1$.

Figure 9. Basic BP algorithm for template matching

```

BP TEMPLATE-MATCHING ALGORITHM
Compute the template:  $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N)$ 
Compute the compatibilities between nodes and local evidences: Eq. (37)

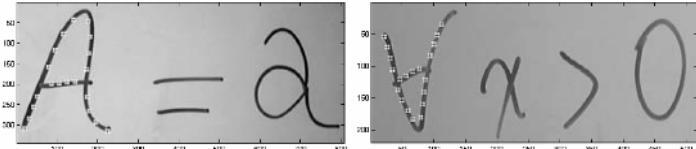
$$\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) = e^{-U_{ij}^{sym}(\mathbf{q}_i, \mathbf{q}_j)}$$


$$\psi_i(\mathbf{q}_i) = \frac{P_{on}(I_e(\mathbf{x}_i))P_{ang}(\phi(\mathbf{x}_i) - \theta_i)}{P_{off}(I_e(\mathbf{x}_i))U(\phi(\mathbf{x}_i))}$$


INITIALIZE messages (uniform);
 $t = 0$ 
 $m_{ij}^{(t)}(\mathbf{q}_j) = \frac{1}{|S|} \quad \forall \mathbf{q}_j \in S$ 
REPEAT
 $m_{ij}^{(t+1)}(\mathbf{q}_j) = \frac{1}{Z_{ij}} \sum_{\mathbf{q}_i} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \psi_i(\mathbf{q}_i) \prod_{k \in N(i) \sim j} m_{ki}^{(t)}(\mathbf{q}_i),$ 
being  $Z_{ij} = \sum_{\mathbf{q}_i} m_{ij}^{(t+1)}(\mathbf{q}_i)$ 
 $t = t + 1$ 
UNTIL CONVERGED;
 $b_i(\mathbf{q}_i) = \frac{1}{Z'_i} \psi_i(\mathbf{q}_i) \prod_{k \in N(i)} m_{ki}(\mathbf{q}_i), Z'_i$  is a norm. factor

```

Figure 10. Results for “A” template. Note that the template is rotation-invariant.



If the graph is a tree, then $b_i(\mathbf{q}_i)$ is guaranteed to be equal to the true marginal $P(\mathbf{q}_i | D)$ when update equations have converged to a fixed point.

It is convenient to introduce two-node marginal probabilities. Their corresponding two-node beliefs can be obtained analogously to the equation for one-node belief by (see Figure 8. Right image):

$$b_{ij}(\mathbf{q}_i, \mathbf{q}_j) = \frac{1}{Z} \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) \psi_i(\mathbf{q}_i) \psi_j(\mathbf{q}_j) \prod_{k \in N(i) \sim j} m_{ki}(\mathbf{q}_i) \prod_{l \in N(j) \sim i} m_{lj}(\mathbf{q}_j) \quad (44)$$

The computational cost of a message update is $|S|^2$. But the compatibility quantity $\psi_{ij}(\mathbf{q}_i, \mathbf{q}_j)$ is very small for many combinations. This sparsity can be exploited for considering only (x_j, y_j) sufficiently close to (x_i, y_i) . In Figure 9 we summarize the BP algorithm for template matching, and some results are shown in Figure 10.

Bethe Free Energies

There is a strong connection between computing the beliefs using BP and minimizing the so called Bethe free energy defined by:

$$F_{Bethe} = \sum_{(ij)} \sum_{\mathbf{q}_i, \mathbf{q}_j} b_{ij}(\mathbf{q}_i, \mathbf{q}_j) (E_{ij}(\mathbf{q}_i, \mathbf{q}_j) + \ln b_{ij}(\mathbf{q}_i, \mathbf{q}_j)) - \sum_i (n_i - 1) \sum_{\mathbf{q}_i} b_i(\mathbf{q}_i) (E_i(\mathbf{q}_i) + \ln b_i(\mathbf{q}_i)) \quad (45)$$

where $E_i(\mathbf{q}_i) = \ln b_i(\mathbf{q}_i)$, $E_{ij}(\mathbf{q}_i, \mathbf{q}_j) = -\ln \psi_{ij}(\mathbf{q}_i, \mathbf{q}_j) - \ln \psi_i(\mathbf{q}_i) - \ln \psi_j(\mathbf{q}_j)$ and n_i the number of neighbours of the i-th node. Actually, when there are no loops, the BP beliefs are the global minima of the Bethe free energy and, what is more important, a set of beliefs gives a BP fixed point in any graph if and only if they are local stationary points of the Bethe free energy. This theorem has been proved by taking Lagrange multipliers associated to the marginality and probability constraints over the beliefs.

$$\begin{aligned} L = & F_{Bethe}(b_i, b_{ij}) + \\ & + \sum_{(ij)} \sum_{\mathbf{q}_j} \lambda_{ij}(\mathbf{q}_j) [\sum_{\mathbf{q}_i} b_{ij}(\mathbf{q}_i, \mathbf{q}_j) - b_j(\mathbf{q}_j)] + \sum_{(ij)} \sum_{\mathbf{q}_i} \lambda_{ij}(\mathbf{q}_i) [\sum_{\mathbf{q}_j} b_{ij}(\mathbf{q}_i, \mathbf{q}_j) - b_i(\mathbf{q}_i)] + \\ & + \sum_{(ij)} \gamma_{ij} [\sum_{\mathbf{q}_i, \mathbf{q}_j} b_{ij}(\mathbf{q}_i, \mathbf{q}_j) - 1] + \sum_i \gamma_i [\sum_{\mathbf{q}_i} b_i(\mathbf{q}_i) - 1] \end{aligned} \quad (46)$$

These multipliers are closely connected with the messages used in the update rules. For instance, in the minimization algorithm as proposed in Yuille (2001), there are two loops: an outer loop for updating the beliefs given the Lagrange multipliers, and an inner loop for updating the latter multipliers.

- **Outer loop** (updates beliefs given multipliers): Bethe energy is reduced, with the update below, provided that the multipliers are chosen for satisfying the constraints:

$$\begin{aligned} b_{ij}(\mathbf{q}_i, \mathbf{q}_j; t+1) &= \phi_{ij}(\mathbf{q}_i, \mathbf{q}_j) e^{-\{\lambda_{ij}(\mathbf{q}_j) + \lambda_{ji}(\mathbf{q}_i) + \gamma_{ij}\}} \\ b_i(\mathbf{q}_i; t+1) &= \psi_i(\mathbf{q}_i) \left[\frac{b_i(\mathbf{q}_i; t)}{\psi_i(\mathbf{q}_i)} \right]^{q_i} e^{\{-\lambda_{ii}(\mathbf{q}_i) + \sum_k \lambda_{ki}(\mathbf{q}_i)\}} \end{aligned} \quad (47)$$

- **Inner loop** (constraint satisfaction): Relating the multipliers with the log of the messages we have the second loop. Until convergence:

$$\begin{aligned}
\gamma_{ij}(\tau+1) &= \log \sum_{\mathbf{q}_i, \mathbf{q}_j} \phi_{ij}(\mathbf{q}_i, \mathbf{q}_j) e^{-\{l + \lambda_{ji}(\mathbf{q}_j; \tau) + \lambda_{ji}(\mathbf{q}_i; \tau)\}} \\
\lambda_{ji}(\mathbf{q}_j; \tau+1) &= \frac{1}{2} \log \frac{\sum_{\mathbf{q}_j} \phi_{ij}(\mathbf{q}_i, \mathbf{q}_j) e^{\{-\lambda_{ji}(\mathbf{q}_i; \tau) - \gamma_{ij}(\tau)\}}}{\psi_j(\mathbf{q}_j) \left\{ \frac{b_j(\mathbf{q}_j; t)}{\psi_j(\mathbf{q}_j; t)} \right\}^{q_j} e^{\{q_j + \sum_{k \neq j} \lambda_{kj}(\mathbf{q}_j; \tau)\}}} \\
\lambda_{ji}(\mathbf{q}_i; \tau+1) &= \frac{1}{2} \log \frac{\sum_{\mathbf{q}_i} \phi_{ij}(\mathbf{q}_i, \mathbf{q}_j) e^{\{-\lambda_{ji}(\mathbf{q}_j; \tau) - \gamma_{ij}(\tau)\}}}{\psi_i(\mathbf{q}_i) \left\{ \frac{b_i(\mathbf{q}_i; t)}{\psi_i(\mathbf{q}_i; t)} \right\}^{q_i} e^{\{q_i + \sum_{k \neq i} \lambda_{ki}(\mathbf{q}_i; \tau)\}}}
\end{aligned} \tag{48}$$

GROUPING WITH JUNCTION DETECTION AND PATH SEARCHING

Contour-based segmentation can be informally posed in terms of finding the true locations in the image where the contours of objects lie. When using active contours and related approaches, contour dynamics embed a typically gradient-descent procedure. Such a procedure drives the contour, usually closed, to a stable, and hopefully optimal, set of locations associated to edges in the image. Sometimes, a more complex optimization approach like dynamic programming is used in order to deal with local optima. This approach is particularly useful when one knows the initial and final points of the contour and wants to find the points in between, for instance for extracting the details of a vessel tree while segmenting angiographic images (Figueiredo & Leitão, 1995). Going one step further, it is desirable to combine the detection of interesting points (for instance, those where the curvature of the contour changes dramatically) with the finding of the contours connecting them. This is the particular case of the so called perceptual grouping problem.

Suppose, for instance, that we want to obtain a geometric sketch of an indoor scene as a previous step to performing a more complex task like the estimation of our relative orientation with respect to the scene or even the recognition of the place from where the scene was taken. Indoor images contain meaningful features called junctions which provide useful local information about geometric properties and occlusions. The efficient and reliable extraction of such features is key to many visual tasks like depth estimation, matching, motion tracking and, particularly, segmentation. These requirements impose the design of both good junction models and statistical tests for reducing the number of false positives and negatives.

On the other hand, efficiency and reliability are also required for implementing the grouping strategy itself. Reliability precludes the use of greedy contour-finding methods because they are prone to local minima. However, the use of dynamic-programming strategies is prohibitive due to the quadratic complexity with the number of contour points. However, we should exploit the fact that under certain assumptions such a complexity can be reduced to linearity.

In this section we describe how to combine junction detection and grouping through connecting paths, provided that such paths exist, in order to obtain a geometric

sketch of the scene. First, we will describe junction modeling and detection and then we will describe a solution to the problem of connecting junctions.

Junction Classification

A generic junction model can be encoded by a parametric template:

$$\Theta = (x_c, y_c, r, M, \{\phi_i\}, \{W_i\}) \quad (49)$$

where (x_c, y_c) is the center, r is the radius, M is the number of wedges (sectors of near constant intensity), $\{\phi_i\}$ with $i=1, \dots, N$ are the wedge limits (supposed to be placed on the edge segments convergent in the center) and $\{W_i\}$, the intensity distributions associated with the wedges (see Figure 13).

Some practical assumptions will reduce the number of parameters to be estimated by a junction detector. First, potential junction centers (x_c, y_c) may be localized by a local filter, like the Plessey detector (Harris & Stephens, 1988) or, more recently, the SUSAN detector (Smith & Brady, 1997) which declares a corner when the intensity of the center is similar to that of a small fraction of points in the neighborhood. Second, although the optimal radius r may be found (Parida, Geiger, & Hummel, 1998) the cost of doing it is prohibitive for real-time purposes. This is why this parameter is assumed to be set by the user. Furthermore, in order to avoid distortions near the junction center, a small domain with radius R_{\min} around it should be discarded and then $r = R_{\max} - R_{\min}$ where R_{\max} is the scope of the junction.

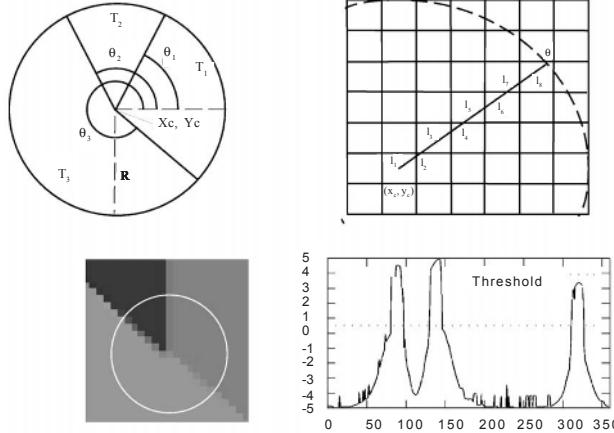
Given the latter simplifications, a junction classification method will focus on finding the optimal number of wedges M , the wedge limits $\{\phi_i\}$ and the wedge intensity distributions $\{W_i\}$. To that purpose, one may follow either a region-based or an edge-based approach, exploiting both of them by the fact that wedge intensity distributions are assumed to be near constant (that is, junctions are built on piecewise smooth areas in the image). In the region-based approach, the optimal position of a wedge limit is the equilibrium point between the neighbouring wedge regions which can be, in turn, fused into a greater one. However, in the edge-based method, edges emanating from the junction center are detected and thresholded. We have proposed these methods in Cazorla and Escolano (2003) and we have found that the second one yields a lower failure rate, although both are prone to over-segmentation. This is why we describe here the edge-based method.

Consequently, finding the wedge limits is addressed by analyzing the one-dimensional contrast profile (see Figure 11, right image) associated with the junction. Such a profile is estimated by computing, for each angle $\phi \in [0, 2\pi]$, the averaged accumulated contrast \tilde{I}_ϕ along the radius in such direction:

$$\tilde{I}_\phi = \frac{1}{r} \sum_{i=1}^N l_i \times I_e(x_i) \quad (50)$$

where $I_e(x_i)$ is the intensity contrast of the pixel x_i associated with segment l_i as is shown in Figure 11 (right), being N , the number of segments needed to discretize the radius along a given direction and not necessarily the same for all directions, as the reliability of the

Figure 11. (Top-left) Parametric model for junctions; (top-right) discrete accumulation of intensity along a direction; (bottom-left) example of an ideal junction; (bottom-right) intensity profile of the junction where each peak represents the location of a wedge limit.



junction detector depends on how “edgeness” is defined. As a starting point, we choose, $I_e(x) = |G * \nabla I(x)|$ that is, the Gaussian smoothed gradient (with typically unitary standard deviation). However, such a measure is finally embodied in the Bayesian edge model defined in Yuille and Coughlan (2000), and presented in the previous section. Consequently, the averaged accumulated contrast defined in Equation 50 is redefined as:

$$\tilde{I}_\phi = \frac{1}{r} \sum_{i=1}^N l_i \times \log \frac{P_{on}(d(x_i) | \phi^*)}{P_{off}(d(x_i))} \quad (51)$$

being $d(x) = (I_e(x), \phi(x))$ and the probabilities of being “on” and “off” the edge, that is $P_{on}(\cdot)$ and $P_{off}(\cdot)$, defined as before:

$$P_{on}(I_e(x) | \phi^*) = P_{on}(I_e(x)) P_{ang}(\phi(x) - \phi^*) \quad \text{and} \quad P_{off}(I_e(x)) = P_{off}(I_e(x)) U(\phi(x)) \quad (52)$$

where $P_{ang}(\phi(x) - \phi^*)$ is the probability of having the correct orientation and $U(\phi(x)) = 1/2\pi$ is the uniform distribution over orientations. In Figure 12 we represent the log-likelihood ratio and the magnitude and orientation of the gradient.

Given the 1D profile, junction classification simply consists of thresholding it properly. Furthermore, as the quality of the contrast profile depends on the correct localization of the junction center, we compensate for small localization errors (2-3 pixels)

by replacing the average \tilde{I}_ϕ by the median \hat{I}_ϕ . In Figure 13 we represent the resulting contrast profile. In practice, after thresholding we filter junctions with less than two wedges or with two wedges defining a quasi-straight line. We show some results in Figure 13. The parameters typically used were: $R_{\min} = 4$, $R_{\max} = 10$, that is $r = 6$, being the threshold $H = 0.5$.

Due to incorrect junction scopes or to bad localizations, the latter method may yield either false positives or false negatives (see Figure 13). However, some of these errors may be corrected by a proper grouping strategy along potential connecting edges between wedges of different junctions.

Connecting and Filtering Junctions

The grouping strategy relies on finding “connecting paths.” A connecting path P of length L rooted on a junction center (x_c, y_c) and starting from the wedge limit defined by ϕ is defined by a sequence of connected segments p_1, p_2, \dots, p_L with fixed or variable length. As points of high curvature are usually associated with corners, and these corners are placed at the origin or at the end of the paths, we assume that the curvature of these paths is smooth. For describing curvature, we define second-order orientation variables $\alpha_1, \alpha_2, \dots, \alpha_{L-1}$ where $\alpha_j = \phi_{j+1} - \phi_j$ is the angle between segments p_{j+1} and p_j . Then, following Yuille and Coughlan (2000), a connecting path P^* should maximize:

$$E(\{p_j, \alpha_j\}) = \sum_{j=1}^L \log \left\{ \frac{P_{on}(p_j)}{P_{off}(p_j)} \right\} + \sum_{j=1}^{L-1} \log \left\{ \frac{P_{\Delta G}(\alpha_{j+1} - \alpha_j)}{U(\alpha_{j+1} - \alpha_j)} \right\}. \quad (53)$$

It is straightforward to transform the cost function in Equation 53 into the typical cost function for snakes if we assume a Gibbs function whose exponent is given by Equation 53. Actually, the first term is related to the external energy whereas the second one is related to the internal energy (curvature). More precisely, the first term is the “intensity reward” and it depends on the edge strength along each segment p_j . Defining

Figure 12. (Top) Sample image and the value of the log-likelihood ratio for pixels; (bottom) magnitude (left) and orientation (right) of the gradient. In the case of orientation, grey is 0, white π and black is $-\pi$.

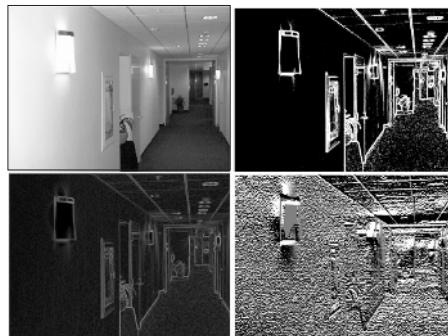
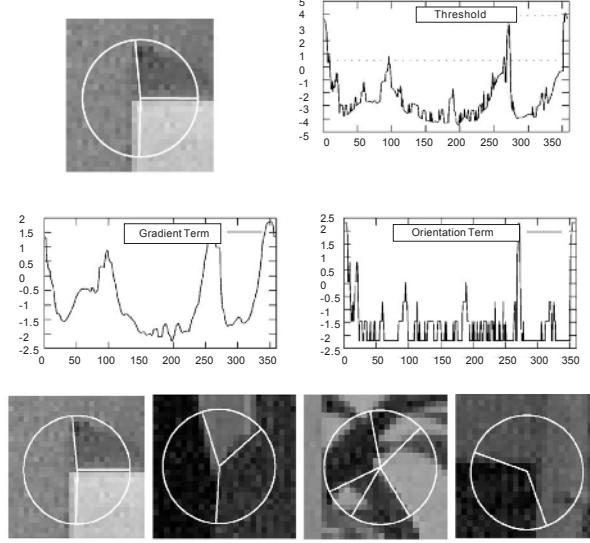


Figure 13. (Top and middle) Contrast profile using the Bayesian edge model. Peaks above the threshold represent suitable wedge limits; (bottom) some results.



the intensity reward of each segment of fixed length F in terms of the same edge model that was used to compute the contrast profile yields:

$$\log \left\{ \frac{P_{on}(p_j)}{P_{off}(p_j)} \right\} = \frac{1}{F} \sum_{i=1}^N l_i \times \log \frac{P_{on}(d(x_i) | \phi^*)}{P_{off}(d(x_i))} \quad (54)$$

Alternatively, $P_{on}(p_j)$ and $P_{off}(p_j)$ may be built on a non-linear filter depending on the gradient magnitude and also on the relative orientation of the segment with respect to the underlying edge.

On the other hand, the second term is the “geometric reward,” which relies on a first-order Markov chain on orientation variables which implements geometric smoothness by exponentially penalizing angular variations:

$$P(\alpha_{j+1} | \alpha_j) = P_{\Delta G}(\alpha_{j+1} - \alpha_j) \text{ and } P_{\Delta G}(\Delta \alpha_j) \propto \exp \left\{ -\frac{C}{2A} |\Delta \alpha_j| \right\} \quad (55)$$

where $\Delta \alpha_j = \alpha_{j+1} - \alpha_j$, modulates the rigidity of the path and $U(\alpha_{j+1} - \alpha_j)$ is the uniform distribution of the angular variation, defined to keep both the intensity and geometric terms in the same range. Such a choice of the geometric reward is dictated by our smoothness assumption but it is desirable to learn it from the data of the application

domain. As we will see later, the effectiveness of this reward depends on its departure from the uniform distribution.

Once the cost function is defined in Equations 53-55 its maximization is addressed by the Bayesian A* algorithm (Coughlan & Yuille, 1999). Given an initial junction center (x_c^0, y_c^0) and an orientation ϕ^0 , the algorithm explores a tree in which each segment p_j may expand Q successors. Although there are Q^N paths for path lengths of $N = L$, the Bayesian A* exploits the fact that we want to detect one target path against clutter, instead of taking the best choice from a population of paths. Then, the complexity of the search may be reduced by pruning partial paths with “too low” rewards. Then, the key element of Bayesian A* is the pruning rule. The algorithm finds the best path surviving to the pruning with an expected convergence rate of $O(N)$. In order to do so, the pruning relies on evaluating the averaged intensity and geometric rewards of the last L_0 segments of a path. These segments are called the “segment block” and the algorithm discards them when their averaged intensity or geometric reward is below a given threshold:

$$\frac{1}{L_0} \sum_{j=zL_0}^{(z+1)L_0-1} \log \left\{ \frac{P_{on}(p_j)}{P_{off}(p_j)} \right\} < T_I \text{ or } \frac{1}{L_0} \sum_{j=zL_0}^{(z+1)L_0-1} \log \left\{ \frac{P_{\Delta G}(\Delta \alpha_j)}{U(\Delta \alpha_j)} \right\} < T_G \quad (56)$$

T_I and T_G being respectively the intensity and geometric thresholds. These thresholds determine the “minimum averaged reward” required to survive the pruning. What is key in this context is that the thresholds are not arbitrary and they must satisfy the following conditions:

$$-D(P_{off} \| P_{on}) < T_I < D(P_{on} \| P_{off}) \text{ and } -D(U_{\Delta G} \| P_{\Delta G}) < T_G < D(P_{\Delta G} \| U_{\Delta G}) \quad (57)$$

denoting $D(\cdot \| \cdot)$ a distance between two distributions (arguments), the so called Kullback-Leibler divergence, defined as:

$$D(P_1 \| P_2) = \sum_{i=1}^M P_1(u_i) \log \frac{P_1(u_i)}{P_2(u_i)} \quad (58)$$

Both the intensity and geometric thresholds are typically chosen as close to their upper bounds as possible. The rationale of these conditions is summarized as follows: If P_{on} diverges from P_{off} , then T_I may have a high value and the pruning cuts many partial paths; otherwise, the pruning will be very conservative. The same reasoning follows for $P_{\Delta G}$ and $U_{\Delta G}$. More precisely, $D(P_{on} \| P_{off})$ quantifies the quality of the edge detector. Then, having large divergences means that the log-likelihood ratio will decide easily between being “on” or “off” the edge, and we will discover false paths sooner. Otherwise, it is hard to know whether a given segment is a good choice or not and the algorithm will wait longer before discarding a partial path. Similarly, $D(P_{\Delta G} \| U_{\Delta G})$ measures the departure from “geometric ignorance” (dictated by the uniform distribution). The more geometric knowledge, the easier to constrain the search.

The rationale on divergences is independent of the algorithm because there are fundamental limits for the task of discriminating the true task among clutter. There is an

order parameter K whose value determines whether the task may be accomplished (when it is positive) or not:

$$K = 2B(P_{on}, P_{off}) + 2B(U_{\Delta G}, P_{\Delta G}) - \log Q \quad (59)$$

being $B(.,.)$ the Bhattacharyya distance between two distributions:

$$B(P_1, P_2) = -\log \left\{ \sum_{i=1}^M P_1^{1/2}(u_i) P_2^{1/2}(u_i) \right\} \quad (60)$$

The order parameter depends on the quality of the edge detector and the quality of the geometric knowledge and determines \tilde{F} , the expected number of completely false paths having greater reward than the true paths; $K < 0$ implies $\tilde{F} \rightarrow \infty$, $K > 0$ implies $\tilde{F} = 0$, and for $K = 0$ there is a phase transition.

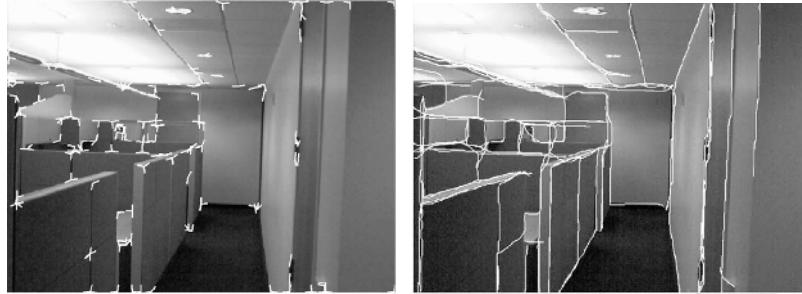
Although the $O(N)$ convergence is ensured, the existence of real-time constraints motivates the extension of the basic pruning rule by introducing an additional rule, though it is not admissible in terms of optimality: In practice we prime the “stability of long paths.” Long paths are more probable than shorter ones to be close to the target because they have survived to more prunes. We will also prune paths with length L_j when:

Figure 14. A* algorithm for path searching

```

BAYESIAN A* PATH SEARCHING
Consider junction center ( $x_c, y_c$ ) an orientation  $\phi$ ;
Generate Q successors and insert then in list OPEN;
WHILE true
    IF OPEN is empty
        IF length ( $P^*$ ) is small RETURN false;
        ELSE RETURN (true,  $P^*$ , final point candidate to junction);
    END
    Select and remove from list  $P \in \text{OPEN}$  minimizing Eq.(53)
     $E(\{p_j, \alpha_j\}) = \sum_{j=1}^L \log \left\{ \frac{P_{on}(p_j)}{P_{off}(p_j)} \right\} + \sum_{j=1}^{L-1} \log \left\{ \frac{P_{\Delta G}(\alpha_{j+1} - \alpha_j)}{U(\alpha_{j+1} - \alpha_j)} \right\}$ 
    IF length ( $P$ ) < length( $P^*$ )  $P^* \leftarrow P$ ;
    Prune OPEN by removing paths with length satisfying Eq. (61)
     $L_j > L_{best} - Z \times L_0$ 
    Expand  $P$ ;
    Insert each successor of  $P$  in OPEN if Eq. (56) is satisfied:
     $\frac{1}{L_0} \sum_{j=zL_0}^{(z+1)L_0-1} \log \left\{ \frac{P_{on}(p_j)}{P_{off}(p_j)} \right\} < T_l \text{ Or } \frac{1}{L_0} \sum_{j=zL_0}^{(z+1)L_0-1} \log \left\{ \frac{P_{\Delta G}(\Delta\alpha_j)}{U(\Delta\alpha_j)} \right\} < T_G$ 
END

```

Figure 15. Junction detection results (Left) and grouping results (Right)

$$L_{best} - L_j > Z \times L_0, \text{ that is, } L_j > L_{best} - Z \times L_0 \quad (61)$$

where L_{best} is the length of the best path so far and $Z \geq 0$ sets the minimum allowed difference between the best path and the rest of paths. For low Z , we introduce more pruning and consequently increase the risk of loosing the true path. When Z is high, shorter paths may survive. It is desirable to find (experimentally) a value for Z representing a trade-off between admissibility and efficiency. A summary of the path-searching algorithm is shown in Figure 14.

The Bayesian A* algorithm with stability pruning expands the search tree until it reaches the center (x_c^f, y_c^f) within a given neighborhood at the end of the selected path. Such searching is done by means of a “range tree.” The cost of building it is $O(J \log J)$, with J being the number of junctions whereas the cost of a query is logarithmic in the worst case.

After reaching a new junction, it is checked that the last segment of the path coincides with a wedge limit ϕ^f and, if so, this limit is labelled as “visited.” However, the search may finish without finding a junction, either when the search queue is empty, or at a “termination point” whose coordinates must be stored. In the first case, if the length of the path is below the block size, we assume that it corresponds to a false wedge limit. In the second case, we assume that we have discovered a potential junction. Then, our “local-to-global” grouping algorithm performs path searching from each non-visited limit and an edge may be tracked in both directions. Some results of the complete grouping algorithm are presented in Figure 15. More details about the algorithm and more complete experimental results may be found in Cazorla et al. (2002).

CONCLUSIONS

In this chapter we have reviewed some optimization-based solutions to segmentation problems. We have analyzed both the formulation of cost functions for the problems (region segmentation, deformable templates and grouping) and the optimization algorithms (simulated and mean-field annealing; jump-diffusion; stochastic gradient descent; optimization via Lagrange multipliers, and so on). We have devoted this chapter

to show the details of how the optimization techniques are formally built. We have chosen some representative cases in the literature. Sometimes the algorithms presented here are not suitable for practical implementations working in real time, for instance, in a mobile robot. However, from a methodological point of view, it is very interesting to show how to formalize the problems and find acceptable solutions to them. The formal simplifications are not arbitrary at all and obey an in-depth analysis of the formal properties of the energy functions and the minimization procedures. More precisely, the region segmentation method described in the first section introduces an important data-driven heuristic in order to avoid almost exhaustive search, although this search is theoretically enabled by the formal properties of jump-diffusion. This approach integrates bottom-up processes with top-down generative processes in order to accommodate both models to data and data to models.

In the second section we describe an efficient approach to model-based segmentation (deformable template matching). In this case there is a greedy application of the belief propagation equations in order to propagate local knowledge and find globally-consistent solutions. Although theoretically these equations do only grasp the best consistency when we have loops in the model connections, they work in practice. An in-depth analysis of message propagation yields additional simplifications. The connection between these equations and an optimization problem (Bethe energy minimization) are also presented in this part of the chapter.

Finally, the third segmentation problem covered is perceptual grouping by integrating feature extraction (junctions) and connection (path searching). A new theoretical cost function, which is a Bayesian alternative to the usual cost function for snakes, is exploited to perform contour-based segmentation with a linear computational cost. Here, interesting ideas from Bayesian inference and information theory yield efficient and robust tests for feature extraction and grouping.

Summarizing, the three solutions to segmentation-related problems presented here deploy interesting methodologies whose development will be key for the development of reliable and efficient solutions for segmentation problems.

REFERENCES

- Cazorla, M., & Escolano, F. (2003). Two Bayesian methods for junction classification. *IEEE Transactions on Image Processing*, 12(3), 317-327.
- Cazorla, M., Escolano, F., Gallardo, D., & Rizo, R. (2002). Junction detection and grouping with probabilistic edge models and Bayesian A* (pp. 1869-1881). (Reprinted from *Pattern Recognition*, 35, with permission from Elsevier.)
- Coughlan, J., & Yuille, A. L. (1999). Bayesian A* tree search with expected O(N) convergence for road tracking. In *EMMCVPR'99* (LNCS 1654, pp.189-200).
- Coughlan, J. M., & Ferreira, S. J. (2002). Finding deformable shapes using loopy belief propagation. In *ECCV 2002* (LNCS 2354, pp. 453-468).
- Figueiredo, M., & Leitão, J. (1995). A nonsmoothing approach to the estimation of vessel contours in angiograms. *IEEE Transactions on Medical Imaging*, 14, 162-172.
- Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 711-732.

- Grenander, U., & Miller, M. I. (1994). Representation of knowledge in complex systems. *Journal of the Royal Stat. Soc. Series B*, 56(4).
- Han, F., Tu, Z. W., & Zhu, S. C. (2004). Range image segmentation by an effective jump-diffusion method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9), 1138-1153.
- Harris, C. G., & Stephens, M. (1988, August 31- September 2). A combined corner and edge detection. In *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK (pp. 147-151).
- Murphy, K. P., Weiss, Y., & Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, UAI '99* (pp.67-475). San Francisco: Morgan Kaufmann Publishers.
- Parida, L., Geiger, D., & Hummel, R. (1998). Junctions: Detection, classification and reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7), 687-698.
- Robert, C. P., & Casella, G. (1999). Monte Carlo statistical methods. In *Springer texts in statistics*. New York: Springer-Verlag.
- Tu, Z. W., & Zhu, S. C. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5), 657-673.
- Yedidia, J. S., Freeman, W. T., Weiss, Y. (2002, January). Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium*, 8, 239-236.
- Yuille, A. L. (2001). An algorithm to minimize the bethe free energy. In *Proceedings of the EMMCVIP'01* (LNCS 2134, pp. 3-18).
- Yuille, A. L., & Coughlan, J. (2000). Fundamental limits of Bayesian inference: Order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2), 160-173.

Chapter III

Variational Problems in Image Segmentation and Γ -Convergence Methods

Giovanni Bellettini, University of Roma, Italy

Riccardo March, Italian National Research Council, Italy

ABSTRACT

Variational models for image segmentation aim to recover a piecewise smooth approximation of a given input image together with a discontinuity set which represents the boundaries of the segmentation. In particular, the variational method introduced by Mumford and Shah includes the length of the discontinuity boundaries in the energy. Because of the presence of such a geometric term, the minimization of the corresponding functional is a difficult numerical problem. We consider a mathematical framework for the Mumford-Shah functional and we discuss the computational issue. We suggest the use of the Γ -convergence theory to approximate the functional by elliptic functionals which are convenient for the purpose of numerical computation. We then discuss the design of an iterative numerical scheme for image segmentation based on the Γ -convergent approximation. The relation between the Mumford-Shah model and the Perona-Malik equation will be also discussed.

INTRODUCTION

The segmentation problem in computer vision consists in decomposing an image into regions which correspond to meaningful parts of objects in a visual scene. The image intensity has to be as uniform as possible inside each region, while sharp transitions take place across the boundaries. Every piece of boundary is an intensity edge and the related problem of edge detection looks for the location of the sharp transitions in intensity. Edge

detection requires a further linking process of the edges into global curves to achieve segmentation. The variational approach to image segmentation unifies edge detection and linking into a single energy minimization method.

Edges are considered as the locations where the modulus of the gradient of image intensity is locally maximum, so that edge detection requires the evaluation of derivatives of this intensity. Torre and Poggio (1986) showed that, because of the presence of noise, the numerical differentiation of an image is an ill-posed mathematical problem. Numerical stability of image differentiation, then, requires a previous regularizing operation that can be attained by means of convolution of the image intensity with a suitable kernel. Marr and Hildreth (1980) proposed the convolution with a Gaussian function in their theory of edge detection, and a rigorous justification of Gaussian filtering before differentiation is given by Torre and Poggio (1986). Morel and Solimini (1995) pointed out that there exists an energy functional associated with the Marr and Hildreth theory. This follows from the observation that the convolution of the image intensity with a Gaussian function is equivalent to the solution of the heat equation with the image as initial datum:

$$\frac{\partial u}{\partial t} = \Delta u, \quad u(x, 0) = g(x)$$

where $g(x)$ denotes the image intensity. The solution of the heat equation in the whole space is given by:

$$u(x, t) = (G_t * g)(x), \quad G_\sigma(x) = \frac{1}{4\pi\sigma} \exp\left(-\frac{x^2}{4\sigma}\right)$$

The heat equation is also the gradient flow of the Dirichlet functional:

$$E(u) = (1/2) \int_{\Omega} |\nabla u|^2 dx$$

with Neumann boundary conditions, where Ω is the image domain. The heat equation requires the choice of a stopping time which corresponds to the parameter σ of the Gaussian function G_σ . Such a parameter yields the spatial scale at which the edges have to be detected. A scale parameter can also be directly introduced in the differential equation by forcing the solution u to remain close to g :

$$\frac{\partial u}{\partial t} = \Delta u - \mu(u - g)$$

Now this partial differential equation is the gradient flow of the functional:

$$E(u) = (\mu/2) \int_{\Omega} (u - g)^2 dx + (1/2) \int_{\Omega} |\nabla u|^2 dx \quad (1)$$

where μ is a positive scale parameter. According to the Marr and Hildreth theory, the differential operator applied after the Gaussian filtering is the Laplacian, and the set K of edges is defined as the set of the zero-crossings of the Laplacian of the image intensity convolved with the Gaussian function:

$$K = \{x \in \Omega : (\Delta G_\sigma * g)(x) = 0\}$$

Torre and Poggio (1986) proved that, if $\nabla(\Delta G_\sigma * g)(x) \neq 0$ at any point x where $(\Delta G_\sigma * g)(x) = 0$, then the set of the zero-crossings is constituted of closed curves or curves that terminate at the boundary of the image. Hence important features, such as junctions between the boundaries of different regions, cannot be recovered. This is a consequence of the subdivision of the process of edge detection into two separate steps: linear filtering and nonlinear detection of the edges (zero-crossings of the Laplacian).

An alternative method to find the set K of boundaries after filtering, is the snake method introduced by Kass, Witkin, and Terzopoulos (1987), which looks for minimizers of the geometric functional:

$$E(K) = \lambda \int_K (1 + k^2) dl - \int_K |\nabla G_\sigma * g|^2 dl$$

where l denotes the arc length, k denotes the curvature of K and λ is a positive parameter (see also Aubert & Kornprobst, 2002). The main drawbacks of such an approach are again that only closed curves can be recovered and that the set K of edges is located after smoothing of the image by linear filtering with consequent distortion of the boundaries.

In order to overcome such drawbacks, the variational approach to segmentation yields a single nonlinear process, by simultaneously placing the boundaries and smoothing the image only off of the boundaries themselves (Chan & Vese, 2002; Mumford & Shah, 1989; Tsai, Yezzi, & Willsky, 2001). This can be achieved by inserting a geometric term in the functional (1):

$$E(u, K) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega \setminus K} |\nabla u|^2 dx + \alpha |K| \quad (2)$$

where $|\cdot|$ denotes the length measure and α is a positive parameter. The functional (2) is known as the Mumford-Shah functional (Mumford & Shah, 1989). By minimizing such a functional, the segmentation problem is formulated as the approximation of the image g by means of a piecewise smooth function u . The optimal function u is discontinuous across the set K and, outside K , u is required to be C^1 and close to g in the sense of the L^2 distance. The functional penalizes large sets K in order to avoid an overly fragmented segmentation. Mumford and Shah conjectured the existence of minimizing pairs (u, K) for the functional E , with the set K made up of a finite number of C^1 arcs.

The variational formulation overcomes the inconvenience of separate smoothing and detection processes at the cost of an increased computational complexity. Because of the presence of the geometric term $\alpha |K|$ in the energy, the numerical minimization of $E(u, K)$ is not a straightforward matter of finite differences and gradient descent. Such a

computational issue will be discussed in the present contribution, and it will be shown how it can be solved by means of approximation in the sense of Γ -convergence.

A related approach based on a single nonlinear process is given by the Perona-Malik theory (Perona & Malik, 1990) which considers a nonlinear version of the heat equation:

$$\frac{\partial u}{\partial t} = \operatorname{div}(f(|\nabla u|)\nabla u), \quad u(x,0) = g(x)$$

where f is a smooth, non increasing function on $[0,+\infty)$ with $f(0)=1$, $f(\xi)>0$ and $f(\xi)$ tending to zero at infinity. In regions where $|\nabla u|$ is small, the Perona-Malik equation behaves essentially as the heat equation, yielding a locally smooth solution; in the proximity of the image boundaries, where $|\nabla u|$ is large, the heat conductance becomes small and the smoothing process is inhibited. Hence, the Perona-Malik equation also attempts to smooth the image only off of the boundaries. The Perona-Malik equation is the gradient flow of the functional:

$$E(u) = \int_{\Omega} \psi(|\nabla u|) dx \quad \text{with } f(|\nabla u|) = \frac{\psi'(|\nabla u|)}{|\nabla u|} \quad (3)$$

and the function ψ is convex-concave. The relations between the Perona-Malik and the Mumford-Shah formulations will be discussed in the sequel.

A MATHEMATICAL FRAMEWORK FOR THE MUMFORD-SHAH FUNCTIONAL

In the following, Ω denotes a bounded open subset of the plane and we assume $g \in L^\infty(\Omega)$. In order to state an existence result of minimizers for the Mumford-Shah functional, it is necessary to enlarge the class of sets K from the curves with finite length to closed sets with a finite one-dimensional Hausdorff measure H^1 (which is just the generalization of length to highly irregular curves). The functional then becomes:

$$E(u, K) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega \setminus K} |\nabla u|^2 dx + \alpha H^1(K)$$

where $H^1(K)$ coincides with the usual length $|K|$ for smooth curves. The variational problem consists in the minimization of the functional $E(u, K)$ over closed sets $K \subset \Omega$ and $u \in C^1(\Omega \setminus K)$. Here ∇u denotes the classical gradient in $\Omega \setminus K$. The proof of existence of minimizers is obtained by resorting to a weak formulation of the variational problem: First the existence of a weak solution is proved, then the solution of the original problem is obtained by proving regularity properties of the weak solution. Besides the proof of the existence result, the weak formulation of the problem has a central role in the approximation by Γ -convergence, and hence in the computational theory. Weak solutions are looked for in a class of discontinuous functions, denoted by $SBV(\Omega)$, which is a special

class of functions of bounded variation introduced by De Giorgi and Ambrosio (Ambrosio, Fusco, & Pallara, 2000).

We say that a function $u \in L^1(\Omega)$ belongs to the class $SBV(\Omega)$ if the gradient Du (in the sense of distributions) is a measure which admits the decomposition:

$$Du = \nabla u \cdot dx + Ju \quad (4)$$

where ∇u here denotes the density of the regular part of the measure Du , and Ju is the jump part. The function ∇u is the gradient outside of the discontinuity set of u (called the approximate gradient). The measure Ju is concentrated (analogously to a Dirac δ measure) on the set of jumps of u , which will be denoted by S_u . The weak formulation of the energy functional is defined by:

$$F(u) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega} |\nabla u|^2 dx + \alpha H^1(S_u) \quad (5)$$

The set of boundaries is now implicitly defined as the discontinuity set of u , so that the weak functional depends only on the function u . It can be proved (Ambrosio, et al., 2000) that the variational problem:

$$(P) \quad \min \{F(u) : u \in SBV(\Omega)\}$$

admits a solution. However, the space SBV contains functions with a highly irregular discontinuity set. There exist examples of SBV functions such that the closure $\overline{S_u}$ of the discontinuity set has positive area (i.e., is a fat set). Since S_u has to represent the boundaries of the segmentation, this could be an undesirable property of the SBV space. Nevertheless, if $u \in SBV(\Omega)$ is a weak minimizer of F , then De Giorgi, Carriero, and Leaci (1989) proved that the set S_u is closed with finite length (hence null area), thus overcoming such a difficulty. Moreover, the function u is of class C^1 outside the set of jumps. Hence, if we set $K = \overline{S_u}$, it follows that the pair (u, K) minimizes the original functional E .

Finally further regularity theorems (Ambrosio, et al., 2000) show that if K is a minimizing set for the functional E , then for H^1 almost every point x in K there is a neighbourhood of x in which K is a C^1 arc. This is a step towards the proof of the Mumford-Shah conjecture.

THE COMPUTATIONAL ISSUE

The numerical solution of the variational problem requires the discretization of the energy functional. However, this is not a straightforward matter of finite difference/finite element discretization because of the presence of the geometric term $H^1(K)$, as it will be discussed in the following.

Let Ω be an open rectangle: Without loss of generality we may assume $\Omega = (0,1) \times (0,1)$. We follow the argument by Blake and Zisserman (1987), and we consider a lattice of coordinates (ih, jh) on Ω , where $h=1/N$ is the lattice spacing $0 \leq i \leq N$, $0 \leq j \leq N$, the

number N being an integer. We denote by $(u_{i,j})_{0 \leq i,j \leq N}$ and $(g_{i,j})_{0 \leq i,j \leq N}$ discrete versions of u and g , respectively, defined on the lattice. By approximating the partial derivatives of the function u with finite differences, a discrete version of the energy on the lattice is the following:

$$E_h = \mu h^2 \sum_{i,j} (u_{i,j} - g_{i,j})^2 + \sum_{i,j} \left\{ (1-l_{i,j})(u_{i+1,j} - u_{i,j})^2 + (1-m_{i,j})(u_{i,j+1} - u_{i,j})^2 \right\} \\ + \alpha h \sum_{i,j} (l_{i,j} + m_{i,j}),$$

where the arrays $(l_{i,j})_{0 \leq i,j \leq N}$ and $(m_{i,j})_{0 \leq i,j \leq N}$ denote boolean-valued matrices called line-processes: $l_{i,j}=1$ denotes the presence of a vertical discontinuity line of length h between pixel (i,j) and pixel $(i+1,j)$, while $l_{i,j}=0$ denotes continuity of u between such pixels. The line process $m_{i,j}$ has an analogous meaning for a horizontal discontinuity line between pixel (i,j) and pixel $(i,j+1)$.

The main drawback of such a discretization is that the resulting energy E_h is highly rotationally noninvariant, since it measures the total length of the discontinuity curves by means of an approximation of such curves with horizontal and vertical pieces of length h . As h tends to zero, such a measure converges (Chambolle, 1995) to a nonisotropic one-dimensional measure Λ (called the cab driver length) which satisfies the following inequalities for any set K :

$$H^1(K) \leq \Lambda(K) \leq \sqrt{2}H^1(K)$$

Then an algorithm based on the minimization of the discrete energy E_h yields segmentation boundaries made of small horizontal and vertical edges. On the other hand, the original energy E in the continuous setting is rotationally invariant and free from such an inconvenience.

The energy E_h has to be minimized with respect to both the variables $u_{i,j}$ and the line processes. The minimization can be done first with respect to the line processes (Blake & Zisserman, 1987): since $l_{i,j} \in \{0,1\}$ we have:

$$\min_{(l_{i,j})} \sum_{i,j} \left\{ (1-l_{i,j})(u_{i+1,j} - u_{i,j})^2 + \alpha h l_{i,j} \right\} = h^2 \sum_{i,j} \psi_h((u_{i+1,j} - u_{i,j})/h)$$

where the function ψ_h denotes the truncated parabola $\psi_h(\xi) = \min\{\xi^2, \alpha/h\}$.

It follows that the problem of minimizing E_h is equivalent to minimizing the following discrete energy with respect to the variables $u_{i,j}$ alone:

$$G_h = \mu h^2 \sum_{i,j} (u_{i,j} - g_{i,j})^2 + h^2 \sum_{i,j} \psi_h((u_{i+1,j} - u_{i,j})/h) + h^2 \sum_{i,j} \psi_h((u_{i,j+1} - u_{i,j})/h)$$

The nonconvex energy G_h can be associated to the following functional in the continuous setting:

$$G_\eta(u) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega} (\psi_\eta(\partial u / \partial x_1) + \psi_\eta(\partial u / \partial x_2)) dx$$

with $x=(x_1, x_2)$. Here the function ψ depends on the ratio $\eta = \alpha / h$ that has now to be considered as a single scale parameter. The rotational invariance of the functional can then be restored by setting:

$$G_\eta(u) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega} \psi_\eta(|\nabla u|) dx$$

If we now replace the truncated parabola with a nonconvex function ψ of the type of equation (3), we obtain just the Perona-Malik functional perturbed with the lower order term $\mu \int (u - g)^2 dx$. It can be proved that such a functional has not a global minimizer in the continuous setting (Chipot, March, & Vitulano, 2001), however it may possess many local minima which may be attained by solving the perturbed Perona-Malik equation.

Now it is relevant to find a way for constructing a family of computable functionals converging to the Mumford-Shah functional. If we consider the family of functionals G_η indexed by the parameter η , there is a difficulty in making the functionals $G_\eta(u)$ converge to $E(u, K)$. The difficulty is that we have to find a way to make a Lebesgue integral, such as $\int \psi_\eta(|\nabla u|) dx$, to converge to something which involves the one-dimensional Hausdorff measure. In the next section we show how to construct a family of functionals which involve only Lebesgue integrals, are numerically convenient and converge to the Mumford-Shah functional. The convergence takes place in the sense of Γ -convergence, which is a variational notion of convergence for functionals: minimizers of the approximating functionals converge to a SBV minimizer of the weak functional F in the L^2 metric.

Recently, Chan and Vese (2001, 2002) proposed a method to compute local minimizers of the Mumford-Shah functional by using level sets. An analogous approach was introduced by Tsai, Yezzi and Willsky (2001, 2002). If a curve C is the boundary of a subset $A \subset \Omega$, they represent C as the zero-level set of a Lipschitz function ϕ (called level set function):

$$\begin{cases} \phi(x) > 0 & \text{if } x \in A \\ \phi(x) < 0 & \text{if } x \in \Omega \setminus \bar{A} \\ \phi(x) = 0 & \text{if } x \in \partial A. \end{cases}$$

Let $\Theta(t)$ denote the Heavside function, that is, $\Theta(t) = 1$ if $t \geq 0$ and $\Theta(t) = 0$ if $t < 0$. If the curve C has finite length, the composition $\Theta \circ \phi$ is a function of bounded variation and the following equality holds:

$$H^1(C) = \int_{\Omega} |D(\Theta \circ \phi)|$$

Now consider k level set functions ϕ_j , $j=1,\dots,k$: The union of the zero-level sets of ϕ_j will represent the boundaries of the segmentation. Let $(\Theta \circ \phi_1, \dots, \Theta \circ \phi_k)$ denote the vector valued Heavside function whose components are only 1 or 0. The image domain Ω is then partitioned into $n = 2^k$ subsets A_i defined by:

$$A_i = \{x \in \Omega : (\Theta \circ \phi_1, \dots, \Theta \circ \phi_k) = I_i\}, \quad I_i \in \{0,1\}^k, \quad I_i \neq I_m \forall i \neq m$$

The partition $\Omega = \cup_i A_i$ defines a segmentation with boundaries $K = (\cup_i \partial A_i) \cap \Omega$. Chan and Vese (2002) write the Mumford-Shah energy for such a segmentation, but replacing the length term $H^1(K)$ with the sum of the length of the zero-level sets of ϕ_j :

$$\alpha \sum_{j=1}^k \int_{\Omega} |D(\Theta \circ \phi_j)| \quad (6)$$

Then they introduce a C^1 regularization Θ_ε of the Heavside function as ε tends to zero, and denoting $\delta_\varepsilon = \Theta'_\varepsilon$ (δ_ε converges in the sense of distributions to the Dirac δ distribution) they approximate the total length of the zero-level sets by means of the expression:

$$\alpha \sum_{j=1}^k \int_{\Omega} |\nabla(\Theta_\varepsilon \circ \phi_j)| dx = \alpha \sum_{j=1}^k \int_{\Omega} (\delta_\varepsilon \circ \phi_j) |\nabla \phi_j| dx$$

Hence the geometric term $H^1(K)$ is replaced by a Lebesgue integral involving differentiable functions, and that allows the local minimization of the energy by using gradient descent and finite difference discretization (Chan & Vese, 2002). However, expression (6) is not equal to the total length of the segmentation boundaries, since some parts of the curves will count more than once in the overall energy: some edges will have a different weight in the total energy. Moreover, the level set method is not able to reconstruct curves with free endpoints which are permitted by the minimizers of the Mumford-Shah functional. In the next section we propose to use an approximation in the sense of Γ -convergence due to Ambrosio and Tortorelli (1992) that allows, at least in principle, the recovery of both curves with free endpoints and the true length term in the energy.

APPROXIMATION BY Γ -CONVERGENCE

The approximation of functionals by means of Γ -convergence (Braides, 1998) is a useful tool for the numerical computation of minimizers of functionals involving geometric terms such as the Mumford-Shah functional. Let X be a separable metric space, let $Y \subseteq X$ be a dense subset and let $F_\varepsilon : Y \rightarrow [0, +\infty]$ be a sequence of functionals. We say that F_ε Γ -converges to $F : X \rightarrow [0, +\infty]$ as ε tends to zero, if the following inequalities are satisfied for every $v \in X$:

$$\forall v_\varepsilon \rightarrow v \quad \liminf_{\varepsilon \rightarrow 0^+} F_\varepsilon(v_\varepsilon) \geq F(v), \quad (7)$$

$$\exists v_\varepsilon \rightarrow v \quad \limsup_{\varepsilon \rightarrow 0^+} F_\varepsilon(v_\varepsilon) \leq F(v). \quad (8)$$

The functional F is called the Γ -limit, and when it exists, is unique. The meaning of inequality (7) follows. Let $\{\varepsilon_k\}$ be a sequence of positive numbers converging to zero; if the limit of the numerical sequence $\{F_{\varepsilon_k}(v_{\varepsilon_k})\}$ exists, it is greater than or equal to $F(v)$, otherwise, the limit of every convergent subsequence of $\{F_{\varepsilon_k}(v_{\varepsilon_k})\}$ is greater than or equal to $F(v)$. The meaning of inequality (8) is analogous.

The most important property of Γ -convergence for the calculus of variations is variational convergence, that is, convergence of the minimizers which is described in the following. If the functional F_ε admits a minimizer in Y for any ε , $\{v_\varepsilon\}$ is a sequence of minimizers of F_ε :

$$F_\varepsilon(v_\varepsilon) = \inf_Y F_\varepsilon \quad \forall \varepsilon > 0$$

and $\{v_\varepsilon\}$ converges to $v \in X$, with v a minimizer of F . Hence, minimizers of the functionals F_ε can be used to approximate minimizers of the Γ -limit F in the metric of the space X .

The weak formulation (5) of the energy is again useful to find the metric space X in which the Γ -convergence to the Mumford-Shah functional takes place, since such a metric is not available on the family of pairs (u, K) . In order to state the Γ -convergence result, it is necessary to introduce in the weak form of the energy an auxiliary variable s , which will turn out to be related to the discontinuity set S_u . The metric space X is defined by:

$$X(\Omega) = L^\infty(\Omega) \times \{s \in L^\infty(\Omega) : 0 \leq s \leq 1\}$$

endowed with the $[L^2(\Omega)]^2$ metric. The Γ -limit $F : X(\Omega) \rightarrow [0, +\infty]$ is then defined by:

$$F(u, s) = \begin{cases} \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega} |\nabla u|^2 dx + \alpha H^1(S_u) & \text{if } u \in SBV(\Omega), s \equiv 1, \\ +\infty & \text{elsewhere in } X(\Omega). \end{cases}$$

In Ambrosio and Tortorelli's theorem, the functional F is approximated by a sequence of elliptic functionals (Ambrosio & Tortorelli, 1992). The subset $Y \subset X$ is defined by:

$$Y(\Omega) = \{(u, s) \in X(\Omega) : u, s \in C^1(\Omega)\}$$

The approximating functionals $F_\varepsilon : Y(\Omega) \rightarrow [0, +\infty]$ are defined by:

$$F_\varepsilon(u, s) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega} (s^2 + \lambda_\varepsilon) |\nabla u|^2 dx + \alpha \int_{\Omega} [\varepsilon |\nabla s|^2 + (1/4\varepsilon)(1-s)^2] dx$$

where $\{\lambda_\varepsilon\}$ denotes a sequence of positive numbers converging to zero such that $\lim_{\varepsilon \rightarrow 0} \lambda_\varepsilon / \varepsilon = 0$. Hence, the geometric term $H^1(S_u)$ has been replaced by a quadratic, elliptic functional depending on the auxiliary variable s .

Ambrosio and Tortorelli (1990; 1992) showed that the family of functionals $\{F_\varepsilon\}$ Γ -converges to the functional F with respect to the $[L^2(\Omega)]^2$ metric. Moreover, they proved that if $(u_\varepsilon^*, s_\varepsilon^*)$ minimizes F_ε for any $\varepsilon > 0$, then the sequence $\{(u_\varepsilon^*, s_\varepsilon^*)\}$ admits a subsequence converging in $[L^2(\Omega)]^2$ to a pair $(u^*, 1)$, with $u^* \in SBV(\Omega)$. Hence, from the variational property of Γ -convergence, it follows that u^* minimizes $F(u, s)$, so that u^* is a weak minimizer of the Mumford-Shah functional. Then, by the regularity properties of weak minimizer discussed above, it follows that the pair $(u^*, \overline{S_{u^*}})$ minimizes the original functional $E(u, K)$.

Now we discuss the role of the auxiliary variable s : The function s controls the gradient $|\nabla u|$ and yields an approximate representation of the discontinuity set S_u . More precisely, the function s_ε^* , which minimizes F_ε , is close to 0 inside a tubular neighbourhood of the set S_{u^*} and is close to 1 outside. The tubular neighbourhood shrinks as ε tends to 0 and $|\nabla u_\varepsilon^*| \rightarrow +\infty$ in the neighbourhood allowing a set of jumps in the limit. Moreover, considering the limsup inequality (8) of Γ -convergence, we can understand how the length measure $H^1(S_u)$ is obtained in the limit as ε tends to zero. This phenomenon, which has also a relevant role in the design of numerical algorithms, is discussed in the following.

Ambrosio and Tortorelli (1992) proved that there exists a sequence $(u_\varepsilon, s_\varepsilon)$ converging to $(u, 1)$ in $[L^2(\Omega)]^2$ such that the inequality (8) is satisfied. With some simplifications, the sequence (s_ε) is constructed in the following way: The computation is based on the observation that, for small ε , the Hausdorff length of the set S_u can be expressed by means of the area of the tubular neighbourhood with radius ε of S_u :

$$H^1(S_u) \approx \frac{\text{area}(\{x \in \Omega : \text{dist}(x, S_u) < \varepsilon\})}{2\varepsilon}$$

The function s_ε is given by $s_\varepsilon(x) = \sigma_\varepsilon(\tau(x))$, with $\tau(x) = \text{dist}(x, S_u)$ in a tubular neighbourhood with infinitesimal radius a_ε of the set S_u . Outside of such a neighbourhood, $s_\varepsilon(x) = 1 - \xi_\varepsilon$, with ξ_ε infinitesimal as ε tends to zero. The function σ_ε is the solution of the Cauchy problem:

$$\frac{d\sigma_\varepsilon}{dt} = \frac{1}{2\varepsilon}(1 - \sigma_\varepsilon) \quad \sigma_\varepsilon(0) = 0 \quad (9)$$

and it is given by:

$$\sigma_\varepsilon(t) = 1 - \exp\left[-\frac{t}{2\varepsilon}\right] \quad (10)$$

From the condition $\sigma_\varepsilon(a_\varepsilon) = 1 - \xi_\varepsilon$, we have $a_\varepsilon = -2\varepsilon \ln \xi_\varepsilon$. Hence, if ε is small enough, the function s_ε makes a sharp transition from 1 to 0 in a tubular neighbourhood of the

discontinuity set S_u , and the sub-level sets $\{x : s_\varepsilon(x) < \delta\}$ with $\delta \in (0,1)$ are an approximate description of S_u .

By using the coarea formula (essentially, an integration with respect to curvilinear coordinates in the neighbourhood of S_u), the expression of the Hausdorff length for small ε , and the equality $|\nabla \tau| = 1$ almost everywhere, we have:

$$I_\varepsilon = \int_{\Omega} [\varepsilon |\nabla s_\varepsilon|^2 + (1/4\varepsilon)(1-s_\varepsilon)^2] dx = 2 \int_{S_u} dH^1 \int_0^{a_\varepsilon} [\varepsilon (d\sigma_\varepsilon/dt)^2 + (1/4\varepsilon)(1-\sigma_\varepsilon)^2] dt + O(\varepsilon)$$

from which, using (9) and (10), it follows:

$$I_\varepsilon = (1/\varepsilon) \int_{S_u} dH^1 \int_0^{a_\varepsilon} (1-\sigma_\varepsilon)^2 dt + O(\varepsilon) = H^1(S_u)(1/\varepsilon) \int_0^{a_\varepsilon} \exp(-t/\varepsilon) dt + O(\varepsilon)$$

Then, using the properties of the infinitesimals, we get:

$$\lim_{\varepsilon \rightarrow 0} I_\varepsilon = H^1(S_u) \lim_{\varepsilon \rightarrow 0} (1 - \xi_\varepsilon^2) = H^1(S_u)$$

It should be noted that the construction of the function s_ε given above captures asymptotically the essential form of the minimizer s_ε^* . Indeed, the construction depends on the function u only through the distance function τ , so that it is the same also for the SBV minimizer u^* . But the \liminf inequality (7) of Γ -convergence shows that, on any other sequence of pairs $(u_\varepsilon, s_\varepsilon)$ converging to $(u^*, 1)$, the limit of the energies F_ε is greater than or equal to $F(u^*, 1)$, so that s_ε yields the essential form of near minimizes of F_ε . This property is relevant for the design of algorithms, as will be discussed in the section on the numerical scheme.

By using Γ -convergence, the original variational problem (P) is then replaced by the family, indexed by ε , of approximated problems:

$$(P)_\varepsilon \quad \min \{F_\varepsilon(u, s) : (u, s) \in Y(\Omega)\}$$

The approximated problems are numerically more convenient because the functionals F_ε are elliptic, and the associated Euler equations admit a straightforward finite difference discretization. The Euler equations for the functionals F_ε form a nonlinear elliptic system given by:

$$\operatorname{div}((s^2 + \lambda_\varepsilon) \nabla u) = \mu(u - g) \tag{11}$$

$$\alpha \varepsilon \Delta s = s |\nabla u|^2 - \frac{\alpha}{4\varepsilon} (1-s) \tag{12}$$

with Neumann boundary conditions. In the sequel, a segmentation algorithm based on the numerical solution of such a system of Euler equations will be given.

A NUMERICAL SCHEME

We now give a numerical scheme for the solution of the Euler equations (11) and (12) by means of finite differences. The present algorithm improves the one presented by March (1992) and we give some partial convergence properties of the scheme. We assume that $\Omega = (0,1) \times (0,1)$. We use finite differences on a lattice of coordinates (ih, jh) , where $h = 1/N$ is the lattice spacing $0 \leq i \leq N$, $0 \leq j \leq N$. We denote by u_{ij} an approximation of $u(ih, jh)$, s_{ij} an approximation of $s(ih, jh)$ and g_{ij} a discrete version of g . We set $x = (x_1, x_2)$, and we use the finite difference approximations:

$$s^2 \frac{\partial u}{\partial x_1} \approx \frac{1}{h} s_{i,j}^2 (u_{i+1,j} - u_{i,j}), \quad \frac{\partial}{\partial x_1} \left(s^2 \frac{\partial u}{\partial x_1} \right) \approx \frac{1}{h^2} [s_{i+1,j}^2 (u_{i+2,j} - u_{i+1,j}) - s_{i,j}^2 (u_{i+1,j} - u_{i,j})]$$

By shifting the index i we have:

$$\frac{\partial}{\partial x_1} \left(s^2 \frac{\partial u}{\partial x_1} \right) \approx \frac{1}{h^2} [s_{i,j}^2 (u_{i+1,j} - u_{i,j}) - s_{i-1,j}^2 (u_{i,j} - u_{i-1,j})]$$

and by symmetrization,

$$\frac{\partial}{\partial x_1} \left(s^2 \frac{\partial u}{\partial x_1} \right) \approx \frac{1}{h^2} [s_{i+1,j}^2 (u_{i+1,j} - u_{i,j}) + s_{i-1,j}^2 (u_{i-1,j} - u_{i,j})]$$

which yields the following discrete version of the Euler equation (11):

$$\begin{aligned} \frac{1}{h^2} [(s_{i+1,j}^2 + \lambda_e)(u_{i+1,j} - u_{i,j}) + (s_{i-1,j}^2 + \lambda_e)(u_{i-1,j} - u_{i,j}) + (s_{i,j+1}^2 + \lambda_e)(u_{i,j+1} - u_{i,j}) \\ + (s_{i,j-1}^2 + \lambda_e)(u_{i,j-1} - u_{i,j})] - \mu(u_{i,j} - g_{i,j}) = 0. \end{aligned} \quad (13)$$

Analogously, the discrete version of the Euler equation (12) is:

$$\frac{4\alpha\epsilon^2}{h^2} (s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1} - 4s_{i,j}) - \frac{4\epsilon}{h^2} s_{i,j} |\nabla u|_{i,j}^2 + \alpha(1 - s_{i,j}) = 0, \quad (14)$$

where,

$$\frac{1}{h^2} |\nabla u|_{i,j}^2 = \frac{1}{4h^2} [(u_{i+1,j} - u_{i-1,j})^2 + (u_{i,j+1} - u_{i,j-1})^2]$$

In order to take into account the Neumann boundary conditions, fictitious points outside the boundary are introduced (that is, points with coordinates $i, j = -1$ or $i, j = N+1$). The boundary condition is imposed by requiring that the finite difference between a nodal value of u_{ij} (s_{ij}) on the boundary and the value of the corresponding fictitious node

vanishes. That determines the values of the fictitious nodes, and the finite difference equations (13) and (14) are then written also for the nodes on the boundary $\partial\Omega$.

If the array $(s_{i,j})_{0 \leq i,j \leq N}$ is kept fixed, the discrete equation (13) for u corresponds to a linear system for the $(N+1)^2$ unknowns $(u_{i,j})_{0 \leq i,j \leq N}$. The numerical solution of such a linear system can be computed by using an iterative method such as the conjugate gradient. The Jacobi and Gauss-Seidel iterative methods can also be used, but they converge slowly if used as stand-alone iterative methods. Nevertheless, if the Jacobi and Gauss-Seidel methods are used in conjunction with the multigrid method (Spitaleri, March, & Arena, 2001), the resulting iterative method can be very fast. Hence we now consider such iterative methods as building blocks of a segmentation algorithm. The iteration of the Jacobi method can be written explicitly:

$$u_{i,j}^{n+1} = \frac{\tilde{u}_{i,j}^n + \mu h^2 g_{i,j}}{\mu h^2 + (s_{i+1,j}^2 + s_{i-1,j}^2 + s_{i,j+1}^2 + s_{i,j-1}^2 + 4\lambda_e)} \quad (15)$$

where n denotes the iteration step and:

$$\tilde{u}_{i,j} = (s_{i+1,j}^2 + \lambda_e)u_{i+1,j} + (s_{i-1,j}^2 + \lambda_e)u_{i-1,j} + (s_{i,j+1}^2 + \lambda_e)u_{i,j+1} + (s_{i,j-1}^2 + \lambda_e)u_{i,j-1}$$

The Gauss-Seidel method could be analogously written. Let us consider the array $(u_{i,j})_{0 \leq i,j \leq N}$ as a vector U with $(N+1)^2$ components. The way we store the elements $u_{i,j}$ in the vector U is not important, provided it remains the same. Let G denote the vector corresponding to the image $g_{i,j}$. We denote by A the linear operator that maps $u_{i,j}$ into $\tilde{u}_{i,j}$, modified for the nodes on the boundary $\partial\Omega$ to take into account the Neumann boundary condition (that is, if $(ih, jh) \in \partial\Omega$ the terms in $\tilde{u}_{i,j}$ with an index equal to either -1 or $N+1$ are eliminated). Note that A is a $(N+1)^2 \times (N+1)^2$ matrix with null principal diagonal. We denote by D the diagonal matrix which corresponds to the multiplication of $u_{i,j}$ by:

$$s_{i+1,j}^2 + s_{i-1,j}^2 + s_{i,j+1}^2 + s_{i,j-1}^2 + 4\lambda_e$$

Again, if $(ih, jh) \in \partial\Omega$, every term with an index equal to either -1 or $N+1$ is eliminated together with a term λ_e from the corresponding diagonal element of D . Then, if the $s_{i,j}$ are kept fixed, the linear system of equations (13) can be written in matrix form:

$$[A - D - \mu h^2 I]U = \mu h^2 G \quad (16)$$

where I denotes the identity matrix. By the definition of the matrices A and D , we have:

$$|D_{k,k}| = \sum_{\substack{l=1 \\ l \neq k}}^{(N+1)^2} |A_{k,l}| \quad \forall k = 1, \dots, (N+1)^2$$

from which it follows that the matrix of the linear system (16) is strictly diagonally dominant, which implies that both the Jacobi (15) and the Gauss-Seidel iterative methods converge.

Analogously, if the array $(u_{i,j})_{0 \leq i,j \leq N}$ is kept fixed, the discrete equation (14) for s corresponds to a linear system for the $(N+1)^2$ unknowns $(s_{i,j})_{0 \leq i,j \leq N}$. The iteration of the Jacobi method is:

$$s_{i,j}^{n+1} = \frac{4\hat{s}_{i,j}^n + (h/\varepsilon)^2}{16 + (4/\alpha\varepsilon)|\nabla u|_{i,j}^2 + (h/\varepsilon)^2} \quad (17)$$

where $\hat{s}_{i,j} = s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1}$.

Let S be the vector corresponding to the array $(s_{i,j})_{0 \leq i,j \leq N}$. Consider now the five points difference operator which maps $s_{i,j}$ into:

$$s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1} - 4s_{i,j} \quad (18)$$

if (ih, jh) belongs to the interior of Ω . If $(ih, jh) \in \partial\Omega$, the Neumann boundary condition is taken into account by eliminating from the expression (18) the terms with an index equal to either -1 or $N+1$ together with a term $s_{i,j}$. The resulting linear operator is then decomposed into the sum of a matrix B with null principal diagonal, and a diagonal matrix $-M$. We denote by T the diagonal matrix which corresponds to the multiplication of $s_{i,j}$ by $|\nabla u|_{i,j}^2$. Then, if the $u_{i,j}$ are kept fixed, the linear system of equations (14) can be written in the form:

$$[4\alpha\varepsilon^2(B-M) - 4\varepsilon T - \alpha h^2 I]S = -\alpha h^2 Q \quad (19)$$

where Q denotes a vector whose components are all equal to one. By the definition of the matrices B and M we have:

$$|M_{k,k}| = \sum_{\substack{l=1 \\ l \neq k}}^{(N+1)^2} |B_{k,l}| \quad \forall k = 1, \dots, (N+1)^2$$

from which it follows that the matrix of the linear system (19) is strictly diagonally dominant, so that both the Jacobi (17) and the Gauss-Seidel iterative methods converge.

Using (10) and taking into account that the function σ_ε yields the essential form of near minimizers, we argue that a minimizer s_ε of F_ε is close to one at a distance of about 10ε from S_{u^*} , where u^* minimizes F . Then the minimizer s_ε makes a sharp transition from 1 to 0 in a tubular neighbourhood of S_{u^*} of width 20ε approximately. Then we have to choose the parameter ε in such a way that the discretization is able to sample such a transition: This is obtained if we have at least $mh = 20\varepsilon$, with m a suitable integer. In this sense the information obtained from the proof of the limsup inequality of Γ -convergence is useful to design the numerical scheme.

EXAMPLES OF COMPUTER EXPERIMENTS

In this section we give some computer examples using real images to illustrate the numerical feasibility of the Γ -convergent approximation. The iterative algorithm is started setting the array $(u_{i,j})_{0 \leq i,j \leq N}$ equal to the input image $(g_{i,j})_{0 \leq i,j \leq N}$, and setting $s_{i,j} = 1$, $i,j = 0, \dots, N$, which corresponds to the absence of discontinuities. For our simulation results we have scaled the data g so that $g_{i,j} \in [0,1]$ for any i, j . The problem of the choice of the weights μ and α of the functional has been discussed by Blake and Zisserman (1987), where the necessary details can be found: μ is a scale parameter and α measures the resistance to noise.

The numerical scheme has been implemented by alternating one iteration of the Gauss-Seidel method for the $u_{i,j}$ variables (keeping the $s_{i,j}$ fixed) with one Gauss-Seidel iteration for the $s_{i,j}$ variables (keeping the $u_{i,j}$ fixed).

The computational complexity of the algorithm is of order of N^2 arithmetic operations for iteration. With the above choice of the starting point of the algorithm, convergence is attained after about 40 total iterations (20 for $u_{i,j}$ and 20 for $s_{i,j}$). At an image resolution of 512x512 pixels, the execution time is about 6 seconds using a workstation with two Intel Xeon 3,0 Ghz processors (the code is implemented in Matlab with C compiler).

A quantitative comparison with a related method, such as the level set method by Chan and Vese, is not easy, but computational complexity and execution times are expected to be of the same order.

Figure 1. Image data g



Figure 2. The function s computed with $\epsilon = 1.4 \cdot 10^{-3}$

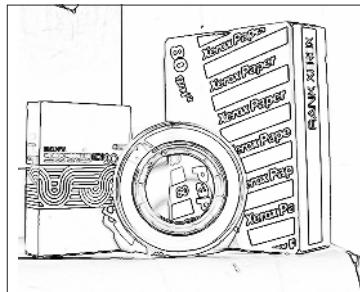


Figure 1 shows the image g of a still life scene. The image resolution is of 576x720 pixels and the brightness measurements are quantized into 256 levels. Figure 2 shows the reconstructed s function, where white corresponds to $s = 1$ and black corresponds to $s=0$. The values $\mu = 0.3$ and $\alpha = 3 \cdot 10^{-4}$ were used for the weights of the functional. The value $\varepsilon = 1.4 \cdot 10^{-3}$ was used for the Γ -convergence parameter.

Figure 3 shows another example of image g . The image resolution is of 384x512 pixels, with 256 grey levels. Figure 4 shows the reconstructed s function, obtained by using the values $\mu = 0.3$, $\alpha = 8 \cdot 10^{-4}$ and $\varepsilon = 2 \cdot 10^{-3}$ for the Γ -convergence parameter.

Figure 5 shows an image g of cells in a tissue. The image resolution is of 512x512 pixels, with 256 grey levels. Figure 6 shows the reconstructed s function, obtained by using the values $\mu = 0.3$, $\alpha = 8 \cdot 10^{-3}$ and $\varepsilon = 2 \cdot 10^{-3}$ for the Γ -convergence parameter.

Figure 3. Image data g



Figure 4. The function s computed with $\varepsilon = 2 \cdot 10^{-3}$

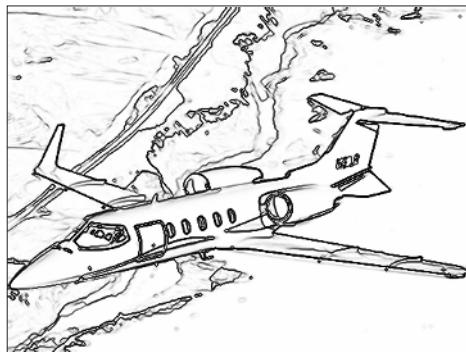
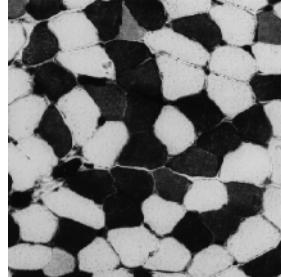
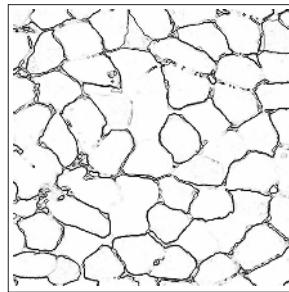


Figure 5. Image data g*Figure 6. The function s computed with $\epsilon = 2 \cdot 10^{-3}$* 

FUTURE TRENDS

The Mumford-Shah model has some drawbacks: It is unable to reconstruct crease discontinuities and yields the over-segmentation of steep gradients. The reconstruction of crease discontinuities is important in the segmentation of range images (Blake & Zisserman, 1987), where visible surfaces have to be reconstructed from two-dimensional range data. To overcome the above defects Blake and Zisserman (1987) introduced a second order functional which is given by:

$$E_1(u, K_0, K_1) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega \setminus (K_0 \cup K_1)} |\nabla^2 u|^2 dx + \alpha H^1(K_0) + \beta H^1(K_1 \setminus K_0)$$

where $|\nabla^2 u|^2$ denotes the squared Euclidean norm of the Hessian matrix of u , and α, β are positive weights. In this model K_0 represents the discontinuity set for u , and $K_1 \setminus K_0$ is the set of crease points (that is, the set where the function is continuous, but its gradient is discontinuous). The following family of functionals was recently proposed by Ambrosio, Faina, and March (2001) to approximate the functional in the sense of Γ -convergence:

$$\begin{aligned} F_{1,\varepsilon}(u,s,\sigma) = & \mu \int_{\Omega} (u-g)^2 dx + \int_{\Omega} (\sigma^2 + \lambda_\varepsilon) |\nabla^2 u|^2 dx + (\alpha - \beta) G_\varepsilon(s) + \beta G_\varepsilon(\sigma) \\ & + \xi_\varepsilon \int_{\Omega} (s^2 + \zeta_\varepsilon) |\nabla u|^\gamma dx, \end{aligned}$$

where:

$$G_\varepsilon(v) = \int_{\Omega} [\varepsilon |\nabla v|^2 + (1/4\varepsilon)(1-v)^2] dx$$

where $\lambda_\varepsilon, \xi_\varepsilon, \zeta_\varepsilon$ denote suitable infinitesimals and $\gamma \geq 2$. In this approximation the function σ is related to the set $K_0 \cup K_1$ and the function s is related to the set K_0 . Hence, a numerical scheme of the type previously discussed could be extended to such an approximation and applied to the segmentation of range images in order to recover surfaces with both jumps and creases. However, since the squared Hessian is now present in the functional, the biharmonic operator will appear in the associated Euler equations, making the convergence of the numerical scheme slower. An appropriate use of multigrid methods will then be essential in order to obtain good convergence rates.

Another field of application of Γ -convergence is the variational theory of Nitzberg-Mumford-Shiota of segmentation with depth (Nitzberg et al., 1993). This is a segmentation model that allows regions to overlap in order to take into account the partial occlusion of farther objects by those that are nearer. The solution of the variational problem yields both the boundaries of the objects in an image, with a reconstruction of the occluded boundaries, and the ordering of the objects in space.

The functional of Nitzberg-Mumford-Shiota is constructed in the following way. Let $\{R_1, \dots, R_n\}$ be a finite collection of sets that covers the image domain Ω , that is, $\bigcup_i R_i = \Omega$. The sets R_i may overlap, and they are endowed with a partial ordering $<$ that denotes occlusion: $R_i < R_j$ means R_i occludes R_j . The set:

$$R'_i = R_i \setminus \bigcup_{R_j < R_i} R_j$$

is the visible (i.e., not occluded) portion of R_i . A segmentation is then defined as an ordered set of overlapping regions, and it is denoted by $(\{R_i\}, <)$. The energy of the segmentation for a piecewise smooth approximation u of the image data g is given by Nitzberg et al. (1993):

$$E_2(u, \{R_i\}, <) = \sum_{i=1}^n \left(\mu \int_{R'_i} (u-g)^2 dx + \int_{R_i} |\nabla u|^2 dx + \alpha \int_{\partial R_i \setminus \partial \Omega} [1 + \psi(k)] dl \right)$$

where k denotes the curvature of ∂R_i and ψ is a positive, convex, even function. The functional is minimized with respect to the function u , the regions R_i and the ordering relation between the regions. For recent mathematical results about problems of this type, see Bellettini and March (2004).

A difficult aspect of working with the functional E_2 is having to deal with the curvature of the boundaries of the regions; it is hard to manage numerically the term:

$$\int_{\partial R_i \setminus \partial \Omega} [1 + \psi(k)] dl \quad (20)$$

In the case $\psi(k) = k^2$, an approach for the numerical minimization of the functional E_2 based on Γ -convergence has been recently proposed by Esedoglu and March (2003). The regions R_1, \dots, R_n are described by means of n functions z_1, \dots, z_n , which approximate their characteristic functions χ_{R_i} :

$$z_i(x) \approx \chi_{R_i}(x), \quad i = 1, \dots, n, \quad \chi_{R_i}(x) = \begin{cases} 1 & \text{if } x \in R_i \\ 0 & \text{if } x \notin R_i, \end{cases}$$

where the approximation takes place in the L^1 topology. Part of the boundary integral in the energy term (20) involves the length of the boundaries ∂R_i which can be approximated in the sense of Γ -convergence in a way similar to the Ambrosio-Tortorelli approximation:

$$\int_{\Omega} [\varepsilon |\nabla z_i|^2 + (1/\varepsilon) W(z_i)] dx \rightarrow c_1 \int_{\partial R_i \setminus \partial \Omega} dl \quad (21)$$

where convergence takes place in the sense of Γ -convergence as ε tends to zero, $W(t) = t^2(1-t)^2$ and c_1 denotes a positive constant depending only on the function W . This approximation, which holds for boundaries of sets, is known as the Modica-Mortola approximation (Modica, 1987).

Based on a conjecture by De Giorgi (1991), the curvature dependent part in the boundary integral (20) can be approximated as follows:

$$(1/\varepsilon) \int_{\Omega} [2\varepsilon \Delta z_i - (1/\varepsilon) W'(z_i)]^2 dx \rightarrow c_2 \int_{\partial R_i \setminus \partial \Omega} k^2 dl \quad (22)$$

where W' denotes the derivative of W and c_2 is another positive constant depending on W . An algorithm for the minimization of the energy E_2 based on the approximations (21) and (22), has been designed by Esedoglu and March (2003) in the case of piecewise constant functions u . However, the function $\psi(k)$ used in the original model of Nitzberg-Mumford-Shiota has linear growth, as $|k|$ tends to infinity in order to better reconstruct the corners along the boundaries ∂R_i . In this sense, the algorithm requires a further improvement.

Finally, all the Γ -convergence approximations can be modified by using nonconvex energies. For instance the term approximating the length of boundaries in the Ambrosio-Tortorelli approximation could be replaced by a functional of the type:

$$\int_{\Omega} [\varepsilon \varphi(|\nabla s|) + (1/4\varepsilon)(1-s)^2] dx$$

where the function φ is chosen nonconvex in order to recover functions s_e with sharper transitions between 0 and 1. An instance of φ function is given by:

$$\varphi(|\nabla s|) = \frac{|\nabla s|^2}{1 + \delta |\nabla s|^2}$$

with $\delta > 0$. Samson, Blanc-Feraud, Aubert and Zerubia (2000) proposed a variational model for image classification based on an approximation of the boundaries of the Modica-Mortola type, modified with a nonconvex φ function. The proof of the Γ -convergence of such an approximation has been given by Aubert, Blanc-Feraud and March (2004) using approximating functionals defined on spaces of finite elements, while the extension of such a result to the Ambrosio-Tortorelli approximation is the subject of current research (Aubert, Blanc-Feraud, & March, 2006).

CONCLUSIONS

We have discussed the Mumford-Shah variational method for image segmentation. The segmentation problem is approached by minimizing an energy functional that takes its minimum at an optimal piecewise smooth approximation to a given image. The functional to be minimized, which involves the computation of the length of the segmentation boundaries, does not admit a straightforward discretization by finite differences. We have proposed the use of the theory of Γ -convergence to approach this type of variational problem. The Γ -convergence allows the approximation of the Mumford-Shah functional by means of elliptic functionals which are numerically convenient. The Euler equations associated to the approximating functionals form a nonlinear system of elliptic equations which can be discretized by finite differences. An iterative numerical scheme can then be designed. The approach can be extended to further segmentation problems, such as surface recovery by segmentation of range images, and segmentation with reconstruction of the ordering of objects in space.

REFERENCES

- Ambrosio, L., Faina, L., & March, R. (2001). Variational approximation of a second order free discontinuity problem in computer vision. *SIAM Journal on Mathematical Analysis*, 32, 1171-1197.
- Ambrosio, L., Fusco, N., & Pallara, D. (2000). *Functions of bounded variation and free discontinuity problems*. Oxford University Press Mathematical Monograph.
- Ambrosio, L., & Tortorelli, V. M. (1990). Approximation of functionals depending on jumps by elliptic functionals via Γ -convergence. *Communication on Pure Applied Mathematics*, 43, 999-1036.
- Ambrosio, L., & Tortorelli, V. M. (1992). On the approximation of free discontinuity problems. *Bollettino Unione Matematica Italiana*, 6-B, 105-123.
- Andreu, F., Caselles, V., & Mazon, J. M. (2004). Parabolic quasilinear equations minimizing linear growth functionals. *Progress in Mathematics*, 223. Basel: Birkhauser.

- Aubert, G., & Kornprobst, P. (2002). *Mathematical problems in image processing*. New York: Springer.
- Aubert, G., Blanc-Feraud, L., & March, R. (2004). Γ -convergence of discrete functionals with nonconvex perturbation for image classification. *SIAM Journal on Numerical Analysis*, 42, 1128-1145.
- Aubert, G., Blanc-Feraud, L., & March, R. (2006). An approximation of Mumford-Shah energy by a family of discrete edge-preserving functionals. *Nonlinear Analysis: Theory, Methods, and Applications*, 61, 1908-1930. Retrieved from <http://www.iac.rm.cnr.it/~march/preprints.html>
- Barenblatt, G. I., Bertsch, M., Dal Passo, R., & Ughi, M. (1993). A degenerate pseudoparabolic regularisation of a nonlinear forward-backward heat equation arising in the theory of heat and mass exchange in stably stratified turbulent shear flow. *SIAM Journal on Mathematical Analysis*, 24, 1414-1439.
- Bellettini, G., & March, R. (2004). An image segmentation variational model with free discontinuities and contour curvature. *Mathematical Models and Methods in Applied Science*, 14, 1-45.
- Bellettini, G., & Fusco, G. (2004). The Γ -limit and the related gradient flow for singular perturbation functionals of Perona-Malik type. Retrieved from <http://cvgmt.sns.it/cgi/get.cgi/papers/belfusa/>
- Bellettini, G., Novaga, M., & Paolini, E. (2005). Global solution to the gradient flow equation of a nonconvex functional. Retrieved from <http://cvgmt.sns.it/cgi/get.cgi/papers/belnovpao05/>
- Blake, A., & Zisserman, A. (1987). *Visual reconstruction*. Cambridge, MA: The MIT Press.
- Braides, A. (1998). Approximation of free discontinuity problems. *Lecture Notes in Mathematics*, 1694. Berlin: Springer.
- Chambolle, A. (1995). Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations. *SIAM Journal on Applied Mathematics*, 55, 827-863.
- Chipot, M., March, R., & Vitulano, D. (2001). Numerical analysis of oscillations in a nonconvex problem related to image selective smoothing. *Journal of Computational & Applied Mathematics*, 136, 123-133.
- Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10, 266-277.
- Chan, T. F., & Vese, L. A. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50, 271-293.
- De Giorgi, E. (1991). Some remarks on Γ -convergence and least square methods. In G. Dal Maso, & G. F. Dell'Antonio (Eds.), *Composite media and homogenization theory* (pp. 135-142). Boston: Birkhauser.
- De Giorgi, E., Carriero, M., & Leaci, A. (1989). Existence theorem for a minimum problem with free discontinuity set. *Archive for Rational Mechanics and Analysis*, 108, 195-218.
- Esedoglu, S. (2001). An analysis of the Perona-Malik equation. *Communication on Pure Applied Mathematics*, 54, 1442-1487.

- Esedoglu, S., & March, R. (2003). Segmentation with depth but without detecting junctions. *Journal of Mathematical Imaging & Vision*, 18, 7-15.
- Fierro, F., Goglione, R., & Paolini, M. (1998). Numerical simulations of mean curvature flow in presence of a nonconvex anisotropy. *Mathematical Models & Methods in Applied Science*, 8, 573-601.
- Gobbino, M. (2003). *Entire solutions of the one-dimensional Perona-Malik equation*. Retrieved from <http://cvgmt.sns.it/cgi/get.cgi/papers/gob03/>
- Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1, 321-331.
- March, R. (1992). Visual reconstruction with discontinuities using variational methods. *Image and Vision Computations*, 10, 30-38.
- Marr, D., & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London, Series B*, 207, 187-217.
- Modica, L. (1987). The gradient theory of phase transitions and the minimal interface criterion. *Archive for Rational & Mechanical Analysis*, 98, 123-142.
- Morel, J. M., & Solimini, S. (1995). *Variational methods in image segmentation*. Boston: Birkhauser.
- Morini, M., & Negri, M. (2003). Mumford-Shah functional as Γ -limit of discrete Perona-Malik energies. *Mathematical Models & Methods in Applied Science*, 13, 785-805.
- Mumford, D., & Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communication on Pure Applied Mathematics*, 42, 577-684.
- Nitzberg, M., Mumford, D., & Shiota, T. (1993). Filtering, segmentation and depth. *Lecture Notes in Computer Science*, 662. Berlin: Springer.
- Perona, P., & Malik, J. (1990). Scale space and edge detection using anisotropic diffusion. *IEEE Transcript, PAMI*, 12, 629-639.
- Rudin, L., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60, 259-268.
- Samson, C., Blanc-Feraud, L., Aubert, G., & Zerubia, J. (2000). A variational model for image classification and restoration. *IEEE Transcript, Pattern Analysis & Machine Intelligence*, 22, 460-472.
- Spitaleri, R.M., March, R., & Arena, D. (2001). A multigrid finite difference method for the solution of Euler equations of the variational image segmentation. *Applied & Numerical Mathematics*, 39, 181-189.
- Torre, V., & Poggio, T. (1986). On edge detection. *IEEE Transcript, PAMI*, 8, 147-163.
- Tsai, A., Yezzi, Jr., A., & Willsky, A. S. (2001). Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transcript, Image Processing*, 10, 1169-1186.
- Yezzi, Jr., A., Tsai, A., & Willsky, A. S. (2002). A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication & Image Representation*, 13, 195-216.

APPENDIX: RELATION WITH THE PERONA-MALIK MODEL

In this section we discuss the Perona-Malik equation (Perona & Malik, 1990):

$$\frac{\partial u}{\partial t} = \operatorname{div}(f(|\nabla u|)\nabla u) \quad u(x,0) = g(x) \quad (23)$$

with Neumann boundary conditions and its relation with the variational approach. A typical nonlinear function f is:

$$f(\xi) = \frac{1}{1+|\xi|^2}$$

so that equation (20) is the gradient flow of the functional E in (3) with $\psi(\xi) = (1/2)\log(1+|\xi|^2)$.

In the following we discuss some properties of the Perona-Malik equation in the one-dimensional case, because the corresponding properties in two dimensions are not yet known. The qualitative properties of ψ which are responsible of the features of the model are the following:

ψ is strictly convex in $(-1,1)$ and strictly concave in $(-\infty,-1) \cup (1,+\infty)$

in addition, ψ is even and has a sublinear growth for $|\xi|$ large. The growth of ψ at infinity is important, since an easy computation shows that approximate step functions carry an energy of order $o(1)$. This in turn is responsible for the high degree of instability of equation (23) because of the presence, in any L^2 -neighbourhood of any initial datum, of staircase functions with zero energy (minimizers of E).

Due to the convex-concave character of ψ , equation (23) has a forward-backward parabolic character; indeed at all points (x,t) where $|\nabla u(x,t)| < 1$ (we call the set of these points the stable region of u), the equation is forward parabolic. On the other hand, at all points (x,t) where $|\nabla u(x,t)| > 1$ (unstable region of u), the equation is backward parabolic. It is therefore reasonable to expect that in the stable region of the initial datum g , the equation has a heat-type character, and the solution becomes smoother and smoother and its local maxima (resp. minima) decrease (resp. increase); in addition, still by the maximum principle, the gradient of u is expected to be confined in the region where ψ is convex. On the other hand, in the unstable regions of g (which we identify with the contours of the image) one could expect a sort of opposite behaviour, namely the norm of the gradient of u should become larger and larger (hence remains in the unstable region), thus enhancing the contours. Actually, a mathematical justification or a falsification of (some of) these behaviours is not available, since a reasonable notion of solution to (23) is still missing even in one dimension.

One of the interesting mathematical problems is indeed to define a local (and, next, a global) in-time solution of (23). In this respect we can mention the following aspects of the dynamics of (23) in one space dimension, some of which are observed in numerical

experiments. First of all, unless one is interested in pointwise smooth solutions to (23) (see Gobbino, 2003), a regularization of (23) and then a passage to the limit is needed in order to produce a solution.

Several different regularizations (and combinations of them) are possible, and in principle they could lead, in the limit (or, if the limit does not exist, in the limit of some converging subsequence for some initial data) to different solutions. Let us mention the approximate solutions obtained by discretizing in space (with a grid, the nodes of which are not allowed to move in space) (Fierro, Goglione, & Paolini, 1998) and in time (Eseedoglu, 2001), the approximate solutions obtained by adding to the equation a term of the form $-\eta^2 u_{xx}$ (Barenblatt, Bertsch, Dal Passo, & Ughi, 1993), those obtained by adding a fourth order perturbation of the form $-\eta^2 u_{xxxx}$ (Bellettini & Fusco, 2004) and those obtained using a system of two equations as in Nitzberg, Mumford and Shiota (1993).

Referring to the fourth order regularization method, numerical experiments and theoretical considerations lead to a notion of solution u with discontinuities (namely, a solution in the class of functions of bounded variation), since even a smooth initial datum can instantly produce a discontinuity in u , or in its space derivative. In addition, there is evidence of the existence of three time scales for the dynamics associated with the equation :

$$u_t = (f(|u_x|)u_x)_x - \eta^2 u_{xxxx} \quad u(x, 0) = g(x) \quad (24)$$

(with suitable boundary conditions) for $0 < \eta \ll 1$. The most interesting phenomenon appears in the first time scale, namely the experimental evidence of the quick formation of microstructures (of smoothed staircase type) in the unstable region of g . This phenomenon has been previously observed also in the analysis of other regularizations of (23) (Nitzberg et al., 1993; Barenblatt et al., 1993; Fierro et al., 1998). After the formation of microstructures, we enter the second time scale, and the evolution in the stable region has a heat-type character, while the interior of the unstable region does not evolve. The discussion of the interesting phenomena that appear at the interface between the stable and the unstable regions is beyond the scope of the present paper. Here we limit ourselves to mention that, at an interface point, solutions to (24) assume approximately zero Neumann boundary conditions from the stable side. In the last time scale the solution has a sort of staircase structure, and the vertical heights of the steps of the stairs have nonzero velocity and have a tendency to merge. There are theoretical elements (Bellettini & Fusco, 2004) to suppose that the evolution, in this last stage of the dynamics, is dictated by the L^2 -gradient flow of an energy E concentrated on staircase functions of the form:

$$u(x) = a_0 + \sum_i a_i \chi_{(x_i, l)}(x)$$

where x_i denote the jump points of u , χ_A denotes the characteristic function of the set A (i.e., $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ if $x \notin A$) and E has the form:

$$E(u) = \sum_i |a_i|^{1/2} \quad (25)$$

It should be noted that even minimizers of the Mumford-Shah functional can exhibit a staircase structure somewhere. However, that happens only in regions where a solution should be smooth, but with a gradient above a suitable threshold (the gradient limit, see Blake & Zisserman, 1987). In such regions a solution with multiple jumps might have the smallest Mumford-Shah energy. In view of the above mentioned observed one-dimensional behaviours, and due to the instability of equation (23) in concrete applications to computer vision, it seems reasonable to use the Perona-Malik scheme only for a short time, namely as an initial filter of the image g , and then use other methods to continue the segmentation procedure.

The Perona-Malik equation is a mathematically interesting example of gradient flow equation arising from an energy functional which has a convex part and unbounded concave parts. Other functions ψ can be considered, such as:

$$\psi(\xi) = \frac{1}{2}[1 - \exp(-\xi^2)] \quad \psi(\xi) = \frac{1}{2}[\arctan(\xi)]^2$$

Bellettini, Novaga and Paolini (2005) have shown the existence of a global solution (in one dimension) of the gradient flow of an energy functional, the integrand ψ of which has the form $\psi(\xi) = (1/2)\min(1, \xi^2)$. The advantage relies essentially on the fact that the instabilities due to microstructures are not anymore present; on the other hand, the disadvantage is that ψ is not smooth.

We have already observed that different scaling can lead from the Perona-Malik equation to gradient flow equations defined on functions with only one jump part; for instance, it happens (Bellettini & Fusco, 2004) that the singular perturbation functionals:

$$E_\varepsilon(u) = \int_{(0,1)} [\varepsilon^3 |u_{xx}|^2 + \frac{1}{\varepsilon \psi(1/\varepsilon)} \psi(u_x)] dx$$

Γ -converges up to a constant factor, to the functional in (25). Interestingly enough, the Perona-Malik functional is also strictly related to the Mumford-Shah functional; for instance, set $\psi_h(\xi) = \min(\xi^2, \alpha/h)$. Chambolle (1995) proved that the energy functionals:

$$E_h(u) = \mu h \sum_i (u_i - g_i)^2 + h \sum_i \psi_h(|u_{i+1} - u_i|/h) \quad (26)$$

defined on piecewise affine functions on a grid of the interval with uniform mesh size h Γ -converge, as h tends to zero, to the Mumford-Shah functional. In two dimensions the Γ -limit turns out to be an anisotropic version of the Mumford-Shah functional with the length term replaced by the cab driver length Λ (Chambolle, 1995). Another instance of a generalization of this result to two dimensions, showing once more the strict relation between the Perona-Malik functional and the Mumford-Shah functional, can be found in Morini and Negri (2003).

Finally, another equation often used in image denoising and restoration is the one arising as the gradient flow of the total variation functional $\int_{\Omega} |Du|$ defined on $BV(\Omega)$, namely:

$$\frac{\partial u}{\partial t} = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (27)$$

Rudin, Osher, and Fatemi (1992) proposed to obtain the denoised image u starting from the observed image g by solving the minimization problem:

$$\min \int_{\Omega} |Du| : \quad \int_{\Omega} (u - g)^2 dx \text{ given}$$

which in practice can be solved as:

$$\min \left\{ \int_{\Omega} |Du| + \frac{1}{2\lambda} \int_{\Omega} (u - g)^2 dx : u \in BV(\Omega) \right\}$$

for some Lagrange multiplier $\lambda > 0$. This formulation is at the basis of the total variation flow equation (27) as a restoration model in image processing; we refer the interested reader to Andreu, Caselles and Mazon (2004) for the mathematical analysis. Note that the total variation functional has a crucial advantage, since it is convex; the disadvantage is its lackness of differentiability at zero and its linear growth. For a given discretization, the discontinuities recovered by the total variation method appear less sharp with respect to the ones recovered by the Mumford-Shah functional. This is due to the fact that the Mumford-Shah functional is the Γ -limit of discrete energies with sublinear growth with respect to the gradient, such as for instance (26). Because of the sublinear growth, at a finite meshsize, sharper discontinuities pay less energy so that they are convenient in a minimize.

Chapter IV

A Graph-Based Image Segmentation Algorithm Using a Hierarchical Social Metaheuristic

Abraham Duarte, Rey Juan Carlos University, Spain

Angel Sanchez, Rey Juan Carlos University, Spain

Felipe Fernandez, Polytechnique University of Madrid, Spain

Antonio S. Montemayor, Rey Juan Carlos University, Spain

ABSTRACT

This chapter proposes a new evolutionary graph-based image segmentation method to improve quality results. Our approach is quite general and can be considered as a pixel- or region-based segmentation technique. What is more important is that they (pixels or regions) are not necessarily adjacent. We start from an image described by a simplified undirected weighted graph where nodes represent either pixels or regions (obtained after an oversegmentation process) and weighted edges measure the dissimilarity between pairs of pixels or regions. As a second phase, the resulting graph is successively partitioned into two subgraphs in a hierarchical fashion, corresponding to the two most significant components of the actual image, until a termination condition is met. This graph-partitioning task is solved as a variant of the min-cut problem (normalized cut) using a hierarchical social (HS) metaheuristic. As a consequence of this iterative graph bipartition stage, pixels or regions are initially merged into the two most coherent components, which are successively bipartitioned

according to this graph-splitting scheme. We applied the proposed approach to brightness segmentation on different standard test images, with good visual and objective segmentation quality results.

INTRODUCTION

Correct image segmentation is generally difficult to achieve and constitutes one of the most complex stages in image analysis. It usually represents a preliminary step for subsequent recognition and image understanding tasks. The segmentation problem consists of partitioning an image into its constituent semantically meaningful regions or objects (Gonzalez & Wood, 2002). The level of division depends on the specific problem being solved. This partition is accomplished in such a way that the pixels belonging to homogeneous regions with regard to one or more features (i.e., brightness, texture or colour) share the same label, and regions of pixels with significantly different features have different labels. Four objectives must usually be considered for developing an efficient generalized segmentation algorithm (Ho & Lee, 2003): *continuous closed contours, non-oversegmentation, independence of threshold setting and short computation time*. Specifically, the oversegmentation problem, which occurs when a single semantic object is divided into several regions, is a tendency of some segmentation methods, like watersheds (Haris, Efstathiadis, Maglaveras, & Katsaggelos, 1998; Hernández & Barner, 2000). Therefore, a subsequent region merging process is needed. In general, high-level knowledge of the input image would be useful in order to reduce the effect of incorrectly merged regions (Brox, 2001).

Many segmentation approaches have been proposed in the literature (Gonzalez & Wood, 2002; Parker, 1996; Sarkar et al., 2000; Sonka et al., 1999). The presented method can be considered as graph-based and pursues a high-level extraction of the image structures. Two kinds of graphs have been considered: pixel-based and region-based. The first approach represents the image as a weighted graph where nodes are the pixels in the original image and the edges together with their associated weights are defined using as local information the distance among pixels and their corresponding brightness values. The region-based graph approach requires an initial image oversegmentation (i.e., watershed transform) that produces a hierarchical top-down, region-based decomposition. To solve the segmentation problem, each pixel is assigned to a class or region by considering only local information (Gonzalez & Wood, 2002). This way, an image is represented by a simplified weighted undirected graph, called a modified region adjacency graph (MRAG). In the MRAG model, nodes are represented by the centres-of-gravity of each region resulting from the initial oversegmentation, and edges together with their associated weights are defined using the spatial distance between nodes, their corresponding brightness value and the corresponding region sizes. The MRAG structure is similar to the region adjacency graph (RAG) (Harris et al., 1998; Hernández & Barner, 2000; Sarkar et al., 2000) but MRAG also enables adding edges between pairs of nodes of non-adjacent regions.

Next, for both graph representations, a bipartition that minimizes the normalized cut value (Shi & Malik, 2000) for the image graph is computed. This process is successively repeated for each of the two resulting regions (image and subgraphs) using a binary splitting schema until a termination condition is met. The graph definition and the

application of a hierarchical social (HS) metaheuristic to efficiently solve the normalized cut problem is the core of the proposed method.

Metaheuristics are high-level general strategies for designing heuristic procedures (Glover & Kochenberger, 2002; Michalewicz & Fogel, 2000; Voss, 2000), they can also be considered as meta-strategies for neighbourhood searches to avoid local optima. The relevance of metaheuristics is reflected in their application for solving many different real-world complex problems, mainly combinatorial. Since the initial proposal of Glover about Tabu Search in 1986, many metaheuristics have emerged to design good general heuristic methods for solving different domain application problems. Genetic programming, GRASP, simulated annealing or ant colony optimization are some other well-known examples of metaheuristics. Their relevance is reflected in the recent publication of many books and papers on this research area (Glover & Kochenberger, 2002).

RELATED WORK

This section revises two segmentation approaches related to our work: metaheuristics-based and graph-cut based. The first approach consists of considering the segmentation task as an optimization problem in which an objective function is improved by means of a metaheuristic procedure. The second approach consists of transforming the image in a graph, where some graph cut techniques are applied.

Metaheuristics-Based Segmentation

Metaheuristics are general procedures successfully applied to a large diversity of very hard combinatorial problems. Surprisingly, compared to the amount of research undertaken on these optimization problems, relatively little work has been devoted to the application of these techniques to computer vision and image processing, despite of the potential advantages of robustness, quality and efficiency (Parker, 1996). Many image analysis tasks like image enhancement, feature selection and image segmentation may be effectively solved using metaheuristics (Poli, 1996).

Among these tasks, segmentation is, in general, one of the most difficult. Usually the standard linear segmentation methods are insufficient for a reliable object classification. The usage of some non-linear approaches, like neural networks or mathematical morphology, has usually provided better results (Sonka et al., 1999). However, the inherent complexity of many scenes (i.e., images with non-controlled illumination conditions or textured images) makes it very difficult to achieve an optimal pixel classification into regions, due to the combinatorial nature of the task. Metaheuristics-based segmentation has been focused on the use of evolutionary algorithms (Ho & Lee, 2003; Poli, 1996; Yoshimura & Oe, 1999) that have reported good performance in relation to more classical segmentation methods. A reduced number of papers using other metaheuristics for image segmentation have been reported. In general, unsupervised image segmentation is modelled as a clustering problem, which has been tackled using fuzzy algorithms (Ballerini, 2004; Kim, 2004) and ant colony optimization (ACO) metaheuristic (Ouadfel, 2002). Our approach for modelling and solving image segmentation as a graph-partitioning problem is related to Shi and Malik's work (2000). These authors use a computational technique based on a generalized eigenvalue problem for

computing the segmentation regions. Instead, we found that higher quality segmentation results can be obtained when applying a new evolutionary metaheuristic called hierarchical social (HS) algorithms (Duarte, 2004), through an iterative solution of a normalized cut problem.

Image Segmentation via Graph Cuts

The first image graph-based segmentation method used fixed neighbourhood structures and local measures in computing segmentation (Zahn, 1971). The method is based on the computing of the minimum spanning tree of the image graph, and it has also been successfully used in clustering applications (Ding, 2001). In general, high quality segmentation is obtained for simple images (synthetic), but for complex images, the results are not acceptable. Reference (Urquhart, 1982) proposes an edge weight normalization stage, which is not suitable to provide reasonable adaptive segmentation results.

Recent literature has witnessed two popular graph cut segmentation methods: the minimum cut (and their derivates) using graph cuts analysis (Shi & Malik, 2000; Veskrler, 2000; Wu et al., 1993) and energy minimization, using the max flow algorithm (Kolmogorov & Zabih, 2002; and Roy & Cox, 1998). More recently, a third major approach based on a generalization of the Swendsen-Wang method (Barbu & Zhu, 2005) has been proposed. In this chapter, we focus on the min-cut approach because, as stated in Shi and Malik (2000), this formulation obtains better results in a segmentation framework.

The min-cut optimization problem, defined for a weighted undirected graph $S = \{V, E, W\}$, consists of finding a bipartition of the set of vertices or nodes of the graph: $V = (C, C')$, such that the sum of the weights of edges with endpoints in different subsets is minimized. Every bipartition of the set of vertices V into C and C' ($V = C \cup C'$) is usually called a *cut* or *cutset* and the sum of the weights of edges, with a vertex in C and the other vertex in C' , is called *cut weight* or *similarity (ies)* between sets C and C' . For the considered min-cut optimization problem, the cut weight is given by:

$$\min_cut(S) = s(C, C') = \sum_{v \in C, u \in C'} w_{vu} \quad (1)$$

is minimized. In Karp (1972) it is demonstrated that the decision version of max-cut (dual version of min-cut problem) is NP-Complete. In this way, we need to use approximate algorithms for finding a high quality solution in a reasonable time.

The min-cut approach has been used by Wu and Leahy (1993) as a clustering method and applied to image segmentation. These authors search a partition of the image graph into k subgraphs such that the similarity (min-cut) among subgraphs is minimized. They pointed out that although the segmentation result for some images is acceptable, in general this method produces an oversegmentation because small regions are favoured. To avoid this fact other functions that try to minimize the effect of this problem are proposed (Ding et al., 2001). The function that must be optimized (minimized) and called min-max cut is:

$$\min_max_cut(S) = \frac{\sum_{v \in C, u \in C'} w_{vu}}{\sum_{v \in C, u \in C} w_{vu}} + \frac{\sum_{v \in C, u \in C'} w_{vu}}{\sum_{v \in C', u \in C} w_{vu}} = \frac{s(C, C')}{s(C, C)} + \frac{s(C', C)}{s(C', C')} \quad (2)$$

where the numerators of this expression are the same $s(C, C')$ and the denominators are the sum of the edge weights belonging to C or C' , respectively. It is important to remark that in an image segmentation framework, it is necessary to minimize the similarity between C and C' (numerators of Equation 2) and maximize the similarity inside C , and inside C' (denominators of Equation 2). In this case, the sum of edge weights between C and C' is minimized, and simultaneously the sums of edge weights inside of each subset are maximized. Shi and Malik (2000) proposed an alternative cut value, called *normalized cut*, which in general gives better results in practical image segmentation problems.

$$Ncut(S) = \frac{\sum_{v \in C, u \in C'} w_{vu}}{\sum_{v \in C, u \in C \cup C'} w_{vu}} + \frac{\sum_{v \in C', u \in C} w_{vu}}{\sum_{v \in C', u \in C \cup C'} w_{vu}} = \frac{s(C, C')}{s(C, C \cup C')} + \frac{s(C', C)}{s(C', C \cup C')} \quad (3)$$

HIERARCHICAL SOCIAL (HS) ALGORITHMS

This section presents the general features of a new evolutionary metaheuristic called hierarchical social (HS) algorithm. In order to get a more general description of this metaheuristic, the reader is pointed to references (Duarte, 2004; and Fernández et al., 2003). This metaheuristic has been successfully applied to several problems in a critical circuit computation (Fernández et al., 2003), scheduling (Duarte, 2004; Duarte et al., 2004), max-cut problems (Duarte et al., 2004) and top-down region-based segmentation (Duarte et al., 2004).

HS algorithms are mainly inspired in the hierarchical social behaviour observed in a great diversity of human organizations. The key idea of HS algorithms consists of a simultaneous optimization of a set of disjoint solutions. Each group of a society contains a feasible solution and these groups are initially randomly distributed to create a disjoint partition of the solution space. Using evolution strategies, where each group tries to improve its objective function in a cooperative fashion or competes with the neighbour groups, better solutions are obtained through the corresponding social evolution. In this social evolution, the groups with lower quality tend to disappear. As a result, the group objective functions are optimized. The process typically ends with only one group that contains the best solution found.

Metaheuristic Structure

For the image segmentation problem, the feasible society is modelled by the specified undirected weighted graph $S = \{V, E, W\}$, also called feasible society graph. The set of individuals are modelled by nodes or vertices V of the graph, and the set of feasible relations are modelled by edges E of the specified graph. The set of similarity relations are described by the weights W . As will be explained in the next section, nodes can represent the image pixels (pixel-based segmentation) or the initial presegmented regions (region-based segmentation). Weighted edges also model the similarity between these pixels or regions.

Figure 1. (a) Synthetic chess board image, (b) society partition in two groups

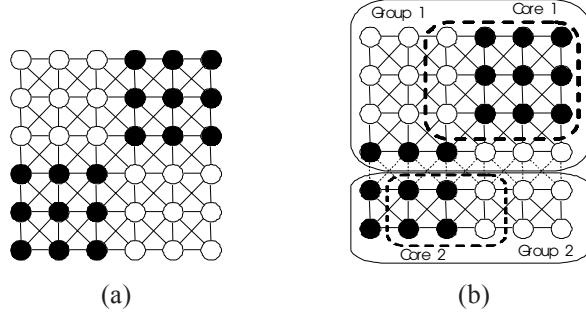


Figure 1a shows an example of the feasible society graph for a synthetic image (chess board) with two major black squares and two major white squares. In this image there are 36 nodes, which are connected using an eight-neighbourhood configuration.

The state of a society is modelled by a hierarchical *policy graph* (Duarte, 2004; Fernández et al., 2003). This graph also specifies a society partition composed by a disjoint set of groups $\Pi = \{g_1, g_2, \dots, g_m\}$, where each individual or node is assigned only to a group. Each group $g_i \subset S$ is composed of a set of individuals and active relations, which are constrained by the feasible society. The individuals of all groups cover the individuals of the whole society. Notice that each group contains exactly one solution.

The specification of the hierarchical policy graph is problem dependent. The initial society partition determines an arbitrary number of groups and assigns individuals to groups. Figure 1b shows a society partition example formed by two groups.

Each individual of a society has two objective functions: *individual objective function* $f1$ and *group objective function* $f2$ that are shared by all individuals of the same group. Furthermore, each group g_i is divided into two disjoint parts: *core* and *periphery*. The core determines the value of the corresponding group objective function $f2$ and the periphery defines the alternative local search region for each involved group.

In the image segmentation framework, the set of nodes of each group g_i is bipartitioned as: $g_i = (C_i, C'_i)$, where C_i represents the core or the group of nodes belonging to the considered cutset and C'_i represents the complementary group of nodes. The core edges are those that have their endpoints in C_i and C'_i . Figure 1b also shows an example of core for the previous considered partition. The core nodes of each group are delimited by a dotted dark line. For each group of nodes $g_i = (C_i, C'_i)$, the group objective function $f2(i)$ is given by the corresponding normalized cut $Ncut(i)$, referred to the involved group g_i :

$$f2(i) = NCut(i) = \frac{\sum_{v \in C_i, u \in C'_i} w_{vu}}{\sum_{v \in C_i, u \in C_i \cup C'_i} w_{vu}} + \frac{\sum_{v \in C_i, u \in C_i} w_{vu}}{\sum_{v \in C'_i, u \in C_i \cup C'_i} w_{vu}} = \frac{s(C_i, C'_i)}{s(C_i, g_i)} + \frac{s(C'_i, C_i)}{s(C'_i, g_i)} \quad (4)$$

$$\forall v \in g_i \quad f2(v, i) = f2(i) = NCut(i)$$

Figure 2. (a) Before intra-group movement, (b) after intra-group movement

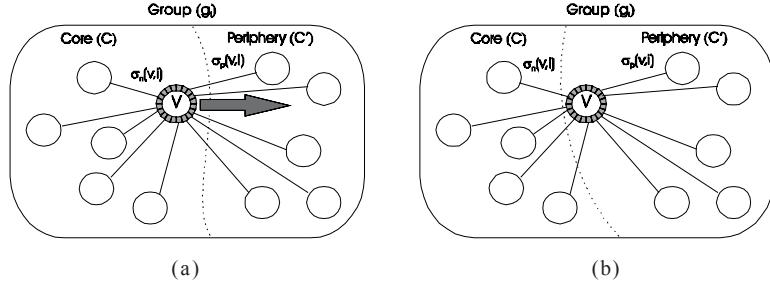
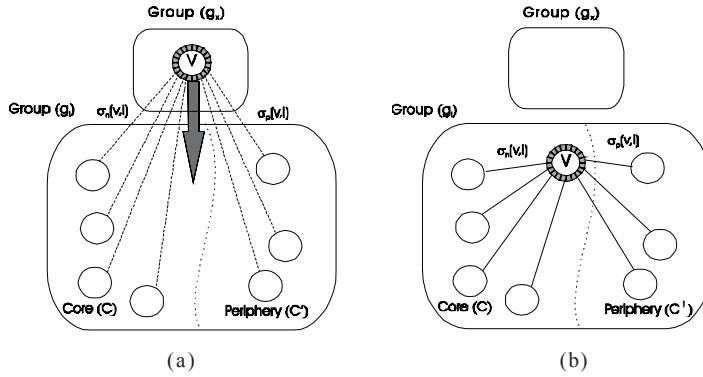


Figure 3. (a) Before inter-group movement, (b) after inter-group movement



where $g_i = C_i \cup C'_i$ and the weights w_{vu} are set to null for the edges that do not belong to the specified graph.

For each individual or node v , the individual objective function $fI(v,i)$ relative to each group g_p is specified by a function that computes the increment in the group objective function when an individual make a movement. There are two types of movements: *intra-group movement* and *inter-group movement*. In the intra-group movement there are two possibilities: the first one consists of a movement from C_i to C'_i , the second one is the reverse movement (C'_i to C_i). Figures 2a and 2b show the movements of an individual v which belong to C_i . In Figure 2a the moving individual (node) is highlighted in grey. The cut edges are also highlighted with thick lines. By the way, Figure 2b shows the group stage after the movement.

The inter-group movement is accomplished by individual v that belongs to a generic group g_x and wants to move from g_x to g_i . There are two possibilities: the first one consists of a movement form g_x to C_i , the second one consists of a movement from g_x to C'_i . As in the previous case, the moving individual and the cut edges are highlighted. This movement is illustrated in Figures 3a and 3b.

The next incremental formula shows the individual function computation of fI when the movement is of type $C_i \rightarrow C'_i$ (described by the function $C_to_C'(v,i)$):

$$f1(v,i) = C_to_C'(v,i) = \frac{s(C_i, C_i) - \alpha'(v,i) + \alpha(v,i)}{s(C_i, g_i) - \alpha'(v,i)} + \frac{s(C_i, C_i) - \alpha'(v,i) + \alpha(v,i)}{s(C_i, g_i) + \alpha(v,i)}$$

where $\alpha(v,i) = \sum_{u \in C_i} w_{vu}$ and $\alpha'(v,i) = \sum_{u \in C_i} w_{vu}$

(5)

This cut function has been derived taking into account the following considerations:

- Before the movement of individual $v \in g_i$, the cut edges were represented by $\alpha(v,i)$, and after the movement the cut edges are represented by $\alpha'(v,i)$. For this reason, the numerators of equation 5 must subtract $\alpha'(v,i)$ and sum $\alpha(v,i)$.
- Before the movement, $v \in g_i$ was contributing to the first denominator with $\alpha'(v,i) + \alpha(v,i)$ because this node belonged to the core. After the movement, the node is in the periphery so $\alpha'(v,i)$ must be subtracted from the first denominator.
- Before the movement, $v \in g_i$ was only contributing to the second denominator with $\alpha'(v,i)$, because this node belonged to the core. After the movement, the node is in the periphery, so the second denominator must be added the term $\alpha(v,i)$.

Notice that this expression can also be deduced from the Figures 3a and 3b and 4a and 4b. The previous expression allows for the selection of each individual v , the group which achieves the corresponding minimum value of $f1(v,i)$. For the other possible movements of individual v (respectively denoted as $C'_i \rightarrow C_i$, $g_x \rightarrow C_i$ and $g_x \rightarrow C'_i$) similar expressions can be obtained.

The HS algorithms here considered try to optimize one of their objective functions ($f1$ or $f2$) depending on the operation phase of the algorithm. During cooperative phase, each group g_i aims to improve independently the group objective function $f2$. During a competitive phase, each individual tries to improve the individual objective function $f1$; the original groups' cohesion disappeared and the graph partition is modified in order to optimize the corresponding individual objective function.

Metaheuristic Process

The algorithm starts from a random disjoint set of feasible solutions. Additionally, for each group an initial random cutset is derived. Groups are successively transformed through a set of evolution strategies. For each group, there are two main strategies: *cooperative strategy* and *competitive strategy*. Cooperative and competitive strategies are the basic search tools of HS algorithms. The first strategy can be considered as a local search procedure in which the quality of the solution contained in each group is autonomously (independently) improved in parallel. This process is maintained during a determined number of iterations (autonomous iterations). Moreover, the procedure works only with the individuals into a group. Figures 4a and 4b show an example of this kind of movement based on a chess board image graph (Figure 1a and 1b). Figure 4a shows a group partition (presented previously in Figure 1b), where the highlighted nodes correspond to the individuals that are performing an inter-group movement. In Figure 4b is shown the policy stage after the movement. As can be seen in this figure, the black and white nodes try to join in the same group, so almost all black nodes are regrouped

Figure 4. (a) Society before intra-group movement, (b) society after intra-group movement

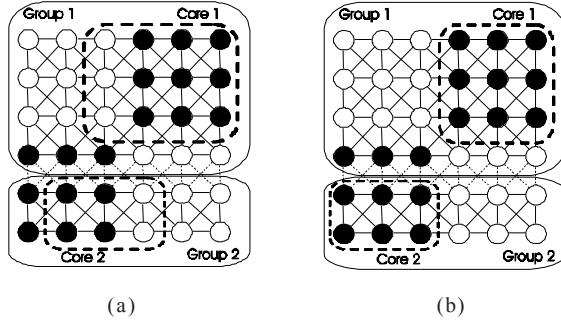
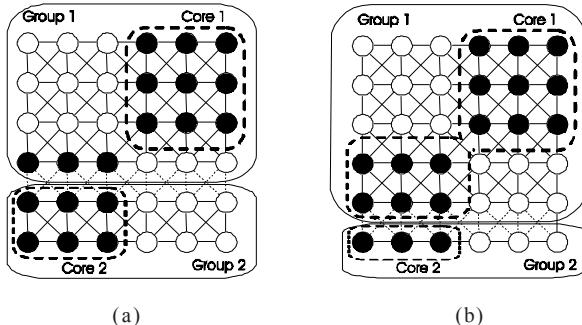


Figure 5. (a) Society before inter-group movement, (b) society after inter-group movement



in Core 1 and Core 2. In addition, almost all the white nodes are regrouped in the periphery. In this situation, the segmentation achieved by Group 1 is better than Group 2, because the first segmentation has separated the black nodes from the white nodes.

The competitive strategy can be considered as a constructive procedure and is oriented to allow the exchange of individuals among groups. Individuals of groups with higher group objective functions (lower quality), can change their groups during a competitive strategy in order to improve their respective individual objective functions. This way, the groups with lower quality tend to disappear because their individuals move to other ones with higher quality. Figures 5a and 5b summarise this behaviour. Figure 5a presents a group partition (previously presented in Figure 4a), where the nodes performing a movement are highlighted in grey. Figure 5b presents the society stage after the intra-group movement. As can be seen in this figure, the black nodes try to join to Core 1 because this is a better quality group.

Individuals of the best groups, with lowest group objective functions (higher quality), can move between core and periphery in order to improve their group objective

function. These strategies produce dynamical group populations, where group annexations and extinctions are possible.

GRAPH-BASED SEGMENTATION VIA NORMALIZED CUTS

Based on graph formulation, two main groups of methods are possible for the image segmentation problem (Shi & Malik, 2000; Gothandaraman, 2004): pixel-based methods, where each node represents to each pixel of the image and region-based methods, in which each node represents a connected set of pixels. A detailed review of these methods can be found in Gothandaraman (2004).

Pixel-Based Formulation

Pixel-based methods work at a very low level of detail, by grouping pixels based on a predefined similarity criterion. These methods construct an undirected weighted graph by taking each pixel as a node and connecting each pair of pixels by a weighted edge, which reflects the likelihood that these two pixels belong to the same object. At first glance, it can be considered that the graph is complete (there is an edge between every pair of pixels). Obviously, this approach is only practical for low resolution images (that is, images with spatial resolution up to 100x100 pixels), because the graph problem becomes intractable. To simplify this representation, only edges between pairs of spatially close pixels are considered (for instance, pixels up to radius of 20 pixels). In pixel-based methods, segmentation criteria are based on global similarity measures. In general, these methods are based on the partition of the graph by optimizing some cut value instead of merging the most similar adjacent regions.

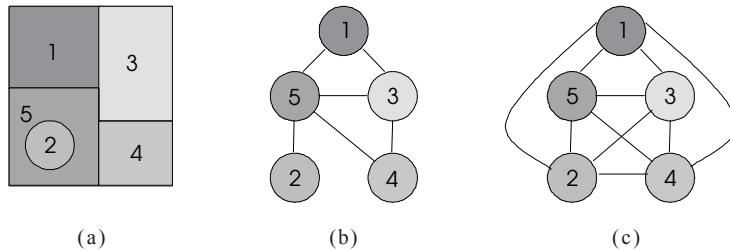
Given an image to be segmented, we first construct its simplified undirected image weighted graph $S = (V, E, W)$. This graph is defined with all the image pixels as nodes and setting the edge weights with a measure of spatial and grey level difference distances (similarity) between the corresponding endpoints. We can define the graph edge weight w_{ij} by connecting the two nodes i and j by the conditional function:

$$\text{if } (abs(x_i - x_j) < r_x) \text{ then } w_{ij} = e^{\frac{-(I_i - I_j)^2}{\sigma_i^2}} \cdot e^{\frac{-(x_i - x_j)^2}{\sigma_x^2}} \text{ else } w_{ij} = 0 \quad (6)$$

where r_x is an experimental threshold value, I_i is the grey level intensity of one pixel i and x_i is the spatial location of this pixel. The values of σ_i , σ_x and r_x are adjusted experimentally and in general they depend on the image features. Non-significant weighted edges, according to defined similarity criteria, are removed from the image graph.

The main problem of this approach is that corresponding graphs, for medium or large resolution images, become intractable. This fact is an immediate consequence of taking into account one node for each pixel. Notice that this approach is highly suitable for low resolution images because the image graph is relatively tractable. Moreover, the proposed method also achieves very accurate and sharp boundaries in segmented objects.

Figure 6. (a) Original image, (b) corresponding RAG of original image, (c) a possible MRAG



Region-Based Formulation

Some region-based methods, like watershed (Haris et al., 1998; Hernández & Barnes, 2000), can be used to simplify the graph structure. The oversegmented image may be modelled by means of the Region Adjacency Graph (RAG), a usual data structure for representing region neighbourhood relations in a segmented image (Sonka et al., 1999). In this graph, adjacent regions are merged in order to reduce the number of regions until a semantically meaningful segmentation is obtained. In general, in the merging process (bottom-up strategy), these methods take into account only local information (feature similarity, boundary continuity, etc.). For more complex images where there are occlusions or discontinuous objects, this approach does not always yield to an adequate segmentation results.

RAG is a weighted undirected graph, where each node represents a region of the oversegmented image and each weighted edge represents the degree of similarity between two adjacent regions (nodes). Figures 6a and 6b show a synthetic image and its corresponding RAG.

Methods based on RAG representation only consider local information for region merging (Haris et al., 1998; Hernández & Barner, 2000; Sarkar et al., 2000; Gothandaraman, 2004). Recently, it has been pointed out that for segmentation purposes it is important not only to consider low level knowledge of the image (coherence of brightness, colour, texture, etc.) but also the extraction of mid- and high-level knowledge about the image objects or the object models (Barbu & Zhu, 2005; Shi & Malik, 2000). Moreover, as stated in Felzenszwalb and Huttenlocher (2004) and Shi and Malik (2000), the image partitioning task is inherently hierarchical and it would be desirable to develop a top-down segmentation strategy which returns a hierarchical partition of the image instead of a flat partition one. Our approach shares this perspective and provides as its segmentation result an adaptable tree-based image bipartition where the first levels of decomposition correspond to major areas or objects in the segmented image. With this aim, we propose a new data structure, called modified region adjacency graph (MRAG), that takes advantage of both RAG and pixel-based representations. The MRAG structure is an undirected weighted graph $S = \{V, E, W\}$, where the set of nodes (V) represents the set of centres-of-gravity of each region, and the set of weighted edges correspond to non-adjacent regions (Figure 6c).

Some characteristics of the MRAG model that yield to advantages regarding to RAG are:

1. MRAG is defined once and it does not need any dynamic updating when merging regions;
2. The MRAG-based segmentation approach is hierarchical and the number of final regions is controlled by the user according to the required segmentation precision; and
3. The segmentation, formulated as a graph partition problem, leads to the fact that extracted objects are not necessarily connected.

Next, some implementation details about the construction of a MRAG structure are given:

- For each region $v_i \in V$ its spatial location x_i is computed as the *centre-of-gravity* of the pixels in the corresponding region resulting from the initial oversegmentation.
- If the resulting region is convex, the centre-of-gravity is inside of it, but if the region is concave, the centre-of-gravity can be outside. In this situation, the centre-of-gravity is placed in the corresponding location of the nearest pixel inside the region.
- The *cardinality*, or *size* $\|E_i\|$, of a region v_i is given by the number of pixels in that region, resulting from the initial oversegmentation.
- For each region $v_i \in V$ its mean intensity I_i is an arithmetic sum divided by the amount of pixels of that region.

The set of edge weights reflect the dissimilarity between each pair of related regions (nodes) v_i and v_j . These connected components can be adjacent or not, but if they are not, they are closer than a determined distance threshold r_x .

The edge weights $w_{ij} \in W$ are computed by the following conditional function:

$$\text{if } (|x_i - x_j| < r_x) \text{ then } w_{ij} = e^{\frac{-C_{ij}(I_i - I_j)^2}{\sigma_i^2}} \cdot e^{\frac{-C_{ij}(x_i - x_j)^2}{\sigma_x^2}} \text{ else } w_{ij} = 0 \quad (7)$$

Equation 7 is almost identical than equation 6 for a pixel-based approach, except that I_i is the grey level mean intensity of region i , and x_i is its centre-of-gravity. The values σ_i and σ_x and the threshold r_x have the same meaning. Notice that these parameters could be tuned in order to improve the segmentation results.

Finally, C_{ij} takes into account the cardinality of the regions i and j . This value is given by:

$$C_{ij} = \frac{\|E_i\| * \|E_j\|}{\|E_i\| + \|E_j\|} \quad (8)$$

where $\|E_i\|$ and $\|E_j\|$ are, respectively, the number of pixels in regions i and j . Non-significant weighted edges, according to defined similarity criteria, are removed from the image graph.

In the presented approach, the preliminary regions result from the initial image oversegmentation preprocessing. In this context, the segmentation problem can be

formulated as a graph bipartition problem, where the set V is partitioned into two subsets C and C' with high similarity among vertices inside each subset and low similarity among vertices of different subsets. As a starting hypothesis, it is assumed that each initial oversegmented region must be small enough in size with regard to the original image and does not have much semantic information.

Previous works (Shi & Malik, 2000; Veskler, 2000; Wu et al., 1993) pointed out that graph-based image segmentation is often sensitive to the choice of the edge weighting function. They used an exponential dependence among parameters, which is highly dependent on the choice of σ_x and σ_y . The proposed region-based method is much less sensitive to this selection, which is derived from the use of a cardinality factor C_{ij} .

METHOD OVERVIEW

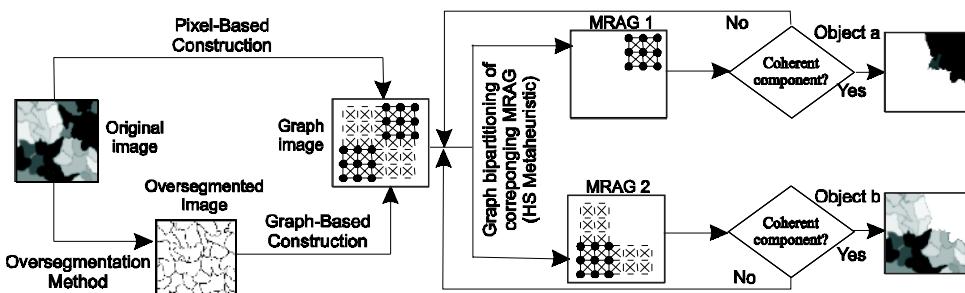
Pixel-based and region-based evolutionary segmentation approaches share the same basic structure. The only difference between both methods is that region-based methods produce an oversegmented image as a preprocessing stage; for example, applying standard watershed segmentation to the initial brightness image. So pixel-based methods use the original image as input and region-based methods use the oversegmented image.

Both methods construct the corresponding image-weighted graph for the input image. This graph is defined by representing each pixel or resulting region by a unique node and defining edges and corresponding edge weights as a measure of spatial location, grey level average difference and cardinality between the corresponding regions.

The final stage consists of iteratively applying the considered HS metaheuristic in a hierarchical fashion to the corresponding subgraph, resulting from the previous graph bipartition, until a termination condition is met. This condition is a trade-off between a required segmentation precision and efficiency. This stage itself constitutes an effective region merging for oversegmented images.

A graphical high level description of the algorithm is presented in Figure 7.

Figure 7. Block diagram of the proposed method



EXPERIMENTAL RESULTS

Computational experiments were evaluated in a 1.7 GHz Intel Pentium 4, 256 MB RAM. All algorithms were coded by the same programmer in C++ without code optimization. Regarding to the implemented HS algorithm, the selected number of groups is $Nodes/100$ and the number of autonomous iterations is 20.

The experimental results are divided into two parts: pixel-based and region-based approaches. In addition, for each approach, qualitative and quantitative results are given. As a quantitative parameter, $NCut$ value is used, as was suggested in Shi and Malik (2000). As a qualitative measure, we show several segmented images using the proposed methods.

Pixel-Based Segmentation Results

We compared the performance of the HS metaheuristic applied to a normalized cut problem with an adapted solution for the same problem using a standard genetic algorithm (Roy & Cox, 1998; Poli, 1996) as proposed by Dolezal et al. (1999) for the Max-Cut problem. Some main details of that metaheuristics implementation are the following: the initial population was 50 individuals, the maximum number of generations was 100 and the probability of crossover and mutation were $p_c = 0, 6$ and $p_m = 1/nodes$, respectively.

Both approaches were tested on several real and synthetic images. The segmentation results for the two real images are respectively shown in Figures 8 and 9.

Table 1 represents the comparative results between the genetic and HS algorithms for several real images (rows 1 and 2) and synthetic images (rows 3 and 4). First and second rows of Table 1 show experimental results for images of Figures 8 and 9. Its columns respectively represent the name and size of images, their corresponding image

Figure 8. Segmentation results for image Pout: (a) initial image, (f) structure of the segmentation tree, (b)-(j) resulting segmented regions according to the segmentation tree

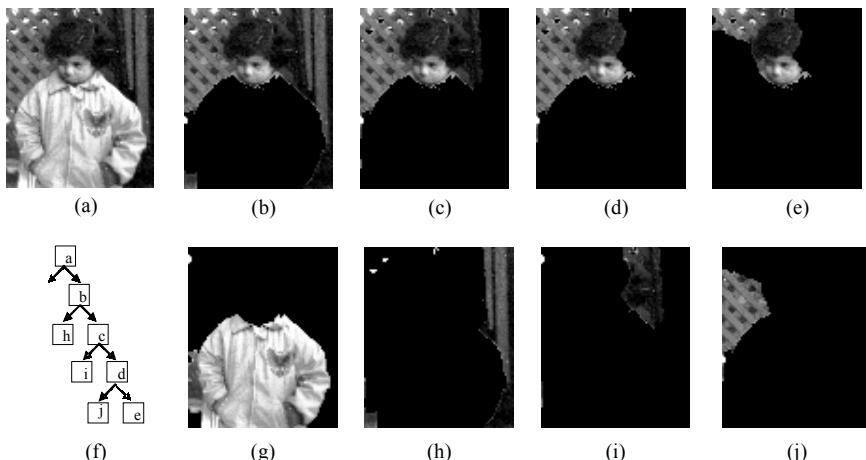


Figure 9. Segmentation results for image Hurricane: (a) initial image, (d) structure of the segmentation tree, (b)-(h) resulting segmented regions according to the segmentation tree

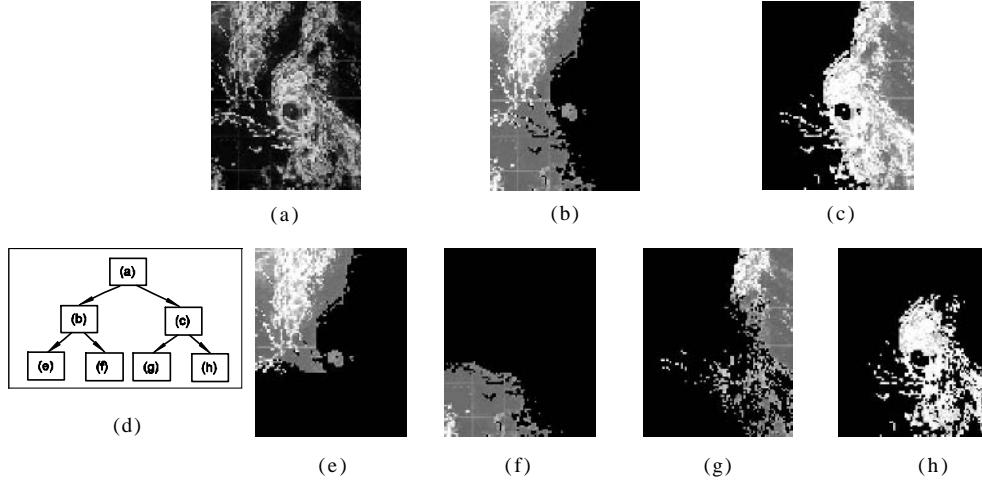


Table 1. Comparison between GA and HS for the first NCut bipartition value for 4 images

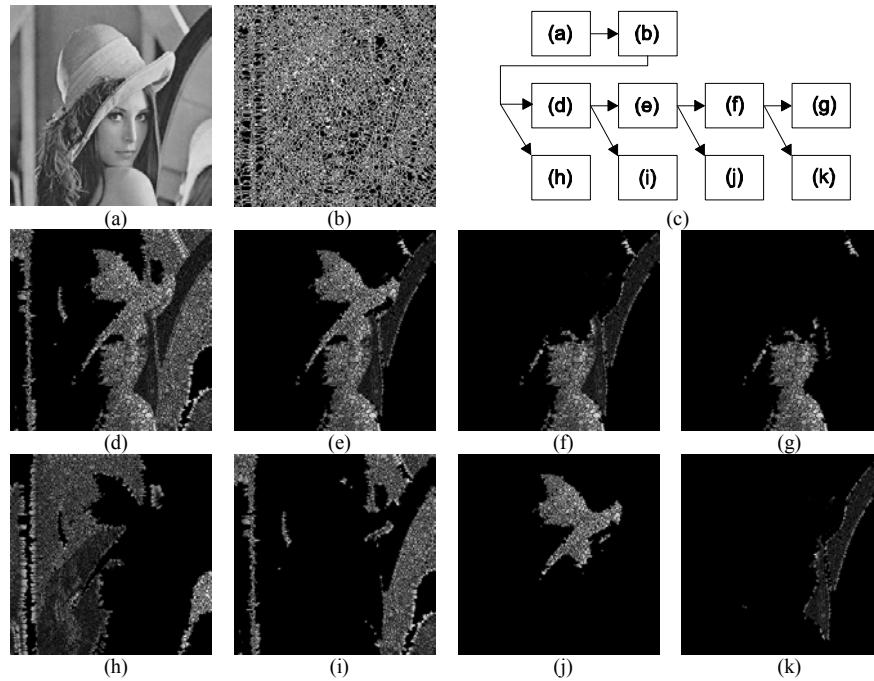
Image graph			Parameters			GA		HS	
Image	Nodes	Edges	σ^2_l	σ^2_x	r_x	NCut	Time	NCut	Time
Pout97x80	7760	251924	0.050	5	10	0.08921	2845	0.03149	716
Hurricane100x80	8000	1103388	0.007	15	15	0.16823	12221	0.02155	3168
Sint1_20x20	400	7414	0.030	10	4	0.08132	43	0.02886	< 1
Sint2_20x20	400	7414	0.030	10	4	0.09125	51	0.01802	< 1

Table 2. Image characteristics and quantitative results

Image	MRAG		MRAG Parameters			GA	HS
	Nodes	Edges	σ^2_l	σ^2_x	r_x		
Lenna256x256	7156	1812344	200	200	40	0.0765	0.0550
Pout256x256	1817	213964	200	200	60	0.2861	0.1734
Windsurf480x320	11155	1817351	200	200	35	0.0511	0.0396
Cameraman256x256	4181	460178	100	200	35	0.6662	0.0497

graphs (number of nodes and edges), the parameters that define the edges weights (s_p , s_x , r_x) and the results for the respective genetic and HS algorithms. For both algorithms, we show a segmentation quality result (for the first value of the $NCut$) and a computational result (execution time in seconds).

Figure 10. Segmentation results for image Lenna: (a) initial image, (b) watershed segmentation, (c) structure of the segmentation tree, (d)-(k) resulting segmented regions according to the segmentation tree

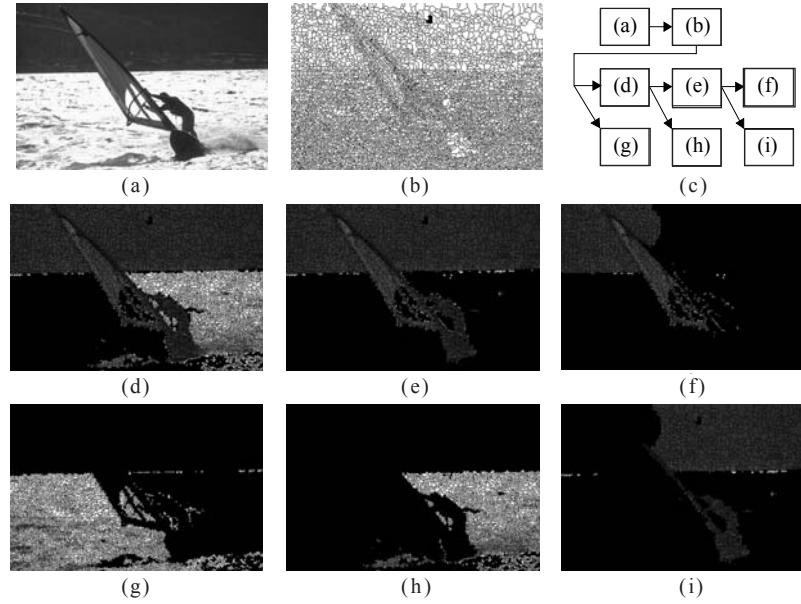


Region-Based Segmentation Results

Again, for a region-based segmentation approach, the performance of the HS metaheuristic applied to the normalized cut problem is compared to an adapted solution for the same problem using a standard genetic algorithm. Table 2 shows the comparative results obtained from standard test images. It reflects the name and resolution of the input image, the characteristics of the corresponding MRAG and the value of the first $NCut$ after the application of the HS metaheuristic.

The upper row of Figure 10a shows the input image (*Lenna*). Its corresponding oversegmented image, obtained by means of watershed algorithm, is presented in Figure 10b. The resulting segmentation tree (Figure 10c) gives a hierarchical view of the segmentation process. The original image is split into two parts (Figures 10d and 10h). Notice that the segmented objects are not connected. This property is especially interesting in images with partially occluded objects, noisy images, etc. The most important part (Figure 10d) is split again, obtaining the images presented in Figures 10e and 10i. As in the previous case, the most significant region (Figure 10e) is partitioned again, obtaining 10f and 10j. This process can be repeated until a determined minimum $NCut$ value is obtained or the process is stopped by the user. The segmented image is given by the union of the final components. The resulting objects correspond to the tree segmentation leaves.

Figure 11. Segmentation results for image Windsurf: (a) initial image, (b) watershed segmentation, (c) structure of the segmentation tree, (d)-(i) resulting segmented regions according to the segmentation tree



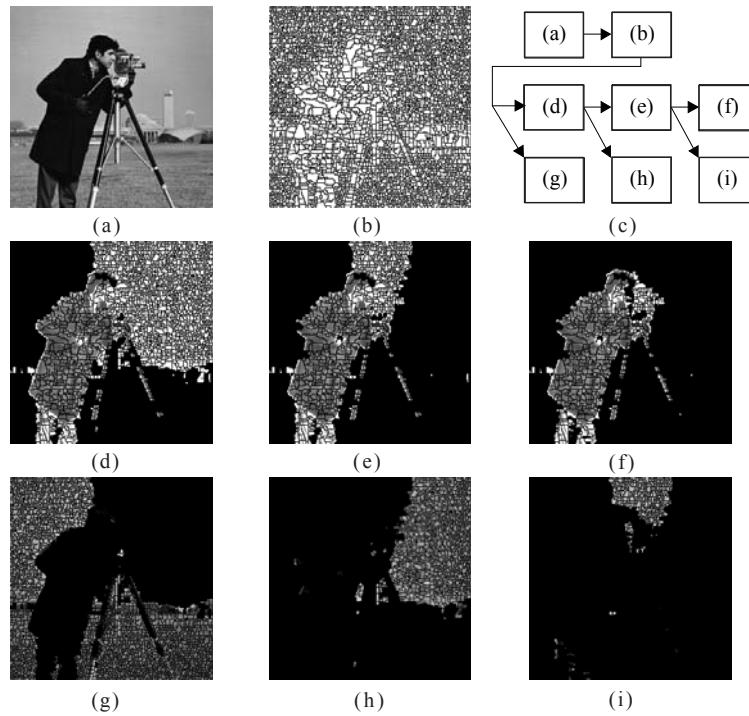
Note that during this hierarchical bipartition stage, the corresponding regions of resulting MRAG are appropriately merged. For the *Lenna* image, a high segmentation quality is achieved. Note that the images presented in Figures 10h and 10i could be also bipartitioned in order to achieve a more detailed segmentation.

The following tested image (Figure 11) is presented with the same format. In the first row appear the original image, its watershed transformation and its segmentation tree. The rest of the rows show successive bipartitions of the image in such a way that in each bipartition the most significant component is obtained through merging the corresponding MRAG regions.

Figure 12 presents the *cameraman*. The segmentation process of this image has a peculiarity relative to the *NCut*. Sometimes, segmentation contains spurious cuts that do not correspond to objects. An example of this phenomenon can be observed in Figure 12h, where the background is segmented into two regions. This fact occurs because *NCut* favours approximately equal size cuts. Sometimes, these spurious cuts do not affect the segmentation results, as in this case, because in the next cut the rest of the main information (in this case, the background and the cameraman) is extracted. If an important object has been split, or in posterior cuts the algorithm can not separate the objects, a different choice of the edge weights or metaheuristic parameters should be considered to improve the segmentation results.

Finally, notice that the first *NCut* value shown in the table offers a quantitative result of the segmentation quality (see the last column of Table 2). As a consequence, the best segmentation results are achieved for the *windsurf* image.

Figure 12. Segmentation results for image Cameraman: (a) initial image, (b) watershed segmentation, (c) structure of the segmentation tree, (d)-(i) resulting segmented regions according to the segmentation tree



The proposed method works very well when individual regions carry low semantic information (i.e., small initial regions compared to the complete image). This way, every pixel inside each initial region belongs also to the same object. For this reason, the output image resulting from the preprocessing stage needs to be highly oversegmented.

Another desirable property of the oversegmented image is that the resulting regions must be homogeneous in size. The use of exponential functions (Equation 7) for MRAG edges weights estimation amplifies the differences among regions (intensity, cardinality or location). Experimentally, we have obtained good results when the sizes of regions (pixels number) do not exceed in 200 those ones corresponding to smallest regions.

CONCLUSIONS

This paper introduces an HS metaheuristic as a graph-based segmentation technique to efficiently improve quality results. Two hierarchical approaches have been considered: pixel-based and region-based segmentation. Both methods have analogous behaviour. First, the input image is modelled as a weighted graph. A key point is that this

graph may establish neighbourhood pixels or regions to those that are not physically adjacent (weighted edges in the image graph). An important advantage of this graph design is that some partially occluded objects, resulting in more than one non-adjacent region in the image, could be incorrectly merged.

The image problem is now equivalent to minimize the *NCut* value in the corresponding graph. The HS metaheuristic was applied to exploit the power of competition and cooperation among pixels or different groups of regions in order to explore the solution space.

For a region-based approach, the input image is firstly oversegmented with a standard method, as a watershed algorithm. Next, the associated MRAG structure is built to model the segmentation problem as a graph. The region representation allows the processing of larger spatial resolution images than the pixel-based approach or any other typical graph-based segmentation method (Haris et al., 1998; Sarkar et al., 2000); and, as we have experimentally shown, that the HS algorithms provide an effective region merging method for oversegmentation problems, achieving high quality segmentation in an efficient way. An important advantage of the approach is that MRAG structure does not need to be updated when merging regions. Moreover, the resulting hierarchical top-down segmentation degree is adaptable to the complexity of the considered image and the application requirements.

The major advantage of using an *NCut* as group objective function in the HS metaheuristic is that the quality of the segmentation is very high. However, the efficiency of the method can be improved by decomposing the image at each level of the segmentation tree in more than two regions.

REFERENCES

- Ballerini, L., Bocchi, L., & Johansson, C. B. (2004). Imager segmentation by a genetic fuzzy C-means algorithm using colour and spatial information. *LNCS*, 3005, 260-269.
- Barbu, A., & Zhu, S-C. (in press). Graph partitioning by Swendsen-Wang cuts. *Journal of Pattern Recognition and Machine Intelligence*.
- Brox, T., et al. (2001). Multi-state region merging for image segmentation. In *Proceedings of the 22nd Symposium on Information and Communication Theory in Benelux* (pp. 189-196).
- Ding, C., He, X., Zha, H., Gu, M., & Simon, H. (2001, November 29-December 2). A min-max cut algorithm for graph partitioning and data clustering. In N. Cercone & T. Y. Lin (Eds.), *Proceedings of ICDM Conference*, San Jose, CA (pp. 107-114). Washington, DC: IEEE Computer Society.
- Dolezal, O., Hofmeister, T., & Lefmann, H. (1999). A comparison of approximation algorithms for the MAXCUT-problem. *Reihe CI* 57/99, SFB 531, Universität Dortmund.
- Duarte, A. (2004). *Algoritmos sociales jerárquicos: Una metaheurística basada en la hibridación entre métodos constructivos y evolutivos*. PhD thesis, Universidad Rey Juan Carlos, Madrid, Spain.
- Duarte, A., Fernández, F., & Sánchez, A. (2004, December 16-18). Software pipelining using hierarchical social metaheuristic. In A. Lotfi (Ed.), *Proceedings of the*

- International Conference on Recent Advances in Soft-Computing (RASC'04)*, Nottingham, UK (pp. 618-623). Nottingham, UK: Nottingham University Press.
- Duarte, A., Fernández, F., Sánchez, A., Sanz, A., & Pantrigo, J. J. (2004). Top-down evolutionary image segmentation using a hierarchical social metaheuristic. *LNCS*, 3005, 301-310.
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167-181.
- Fernández, F., Duarte, A., & Sánchez, A. (2003). A software pipelining method based on a hierarchical social algorithm. In *Proceedings of the 1st MISTA '03 Conference* (pp. 382-385).
- Glover, F., & Kochenberger, G. A. (2002). *Handbook of metaheuristics*. MA: Kluwer Academic Publishers.
- Gonzalez, R. C., & Woods, R. (2002) *Digital image processing* (2nd ed.). NJ: Prentice Hall.
- Gothandaraman, A. (2004). *Hierarchical image segmentation using the watershed algorithm with a streaming implementation*. PhD thesis, University of Tennessee, TN.
- Haris, K., Efstatiadis, S. N., Maglaveras, N., & Katsaggelos, A. K. (1998). Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. on Image Processing*, 7(12), 1684-1699.
- Hernández, S. E., & Barner, K. E. (2000). Joint region merging criteria for watershed-based image segmentation. *International Conference on Image Processing*, 2 (pp. 108-111).
- Ho, S. Y., & Lee, K. Z. (2003). Design and analysis of an efficient evolutionary image segmentation algorithm. *J. VLSI Signal Processing*, 35, 29-42.
- Karp, R. M. (1972). Reducibility among combinatorial problems. In R. Miller & J. Thatcher (Eds.), *Complexity of computer computations* (pp. 85-103). New York: Plenum Press.
- Kim, S. M., & Kim, W. (2004). An algorithm for segmentation gaseous objects on images. *LNCS*, 3005, 322-328.
- Kolmogorov, V. & Zabih, R. (2002). What energy functions can be minimized via graph cuts? In *ECCV*, 3 (pp. 65-81). Copenhagen, Denmark.
- Michalewicz, Z., & Fogel, D. B. (2000). *How to solve it: Modern heuristics* (2nd ed.). New York: Springer.
- Oquadfel, S., & Batouche, M. (2002). Unsupervised image segmentation using a colony of cooperating ants. *LNCS*, 2525, 109-116.
- Parker, J. R. (1996). *Algorithms for image processing and computer vision*. New York: John Wiley.
- Poli, R. (1996). Genetic programming for image analysis. In J. Koza (Ed.), *Genetic programming* (pp. 363-368). Stanford University, CA: MIT Press.
- Roy, S., & Cox, I. (1998, January 4-7). A maximum-flow formulation of the n-camera stereo correspondence problem. In L. Davis, A. Zisserman, M. Yachida, & R. Narasimhan (Eds.), *Proceedings of the International Conference on Computer Vision (ICCV '98)*, Bombay, India (pp. 492-502). Washington, DC: IEEE Computer Society.
- Sarkar, A., Biswas, M.K., & Sharma, K.M.S. (2000). A simple unsupervised MRF model Bbsed image segmentation approach. *IEEE Transaction on Image Processing*, 9(5), 801-811.

- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8), 888-905.
- Sonka, M., Hlavac, V., & Boyle, R. (1999). *Image processing, analysis and machine vision* (2nd ed.). London: Chapman & Hall.
- Urquhart, R. (1982). Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3), 173-187.
- Veskler, O. (2000, June). Image segmentation by nested cuts. In *Proceedings of IEEE CVPR Conference* (pp. 339-344).
- Voss, S. (2001). Meta-heuristics: The state of the art. In A. Nareyek (Ed.), *LNAI 2148* (pp. 1-23). New York: Springer.
- Wu, Z., & Leahy, R. (1993). Optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions Pattern Analysis & Machine Intelligence*, 15(11), 1101-1113.
- Yoshimura, M., & Oe, S. (1999). Evolutionary segmentation of texture image using genetic algorithms. *Pattern Recognition*, 32, 2041-2054.
- Zahn, C. T. (1971). Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Transactions on Computing*, 20, 68-86.

Chapter V

Modeling Complex Dynamic Systems for Image Segmentation

Liang Zhao, University of São Paulo, Brazil

ABSTRACT

Human perception is a complex nonlinear dynamics. Motivated by biological experimental findings, two networks of coupled chaotic elements for image segmentation are introduced in this chapter. In both models, time evolutions of chaotic elements that correspond to the same object in a given image are synchronized with one another, while this synchronized evolution is desynchronized with respect to time evolution of chaotic elements corresponding to other objects in the image. The first model is a continuous flow and the segmentation process incorporates geometrical information of input images; while the second model is a network of discrete maps for pixel clustering, accompanying an adaptive moving mechanism to eliminate pixel ambiguity. Computer simulations on real images are given.

INTRODUCTION

Scene segmentation is a fundamental process in computer vision. It is the ability to attend to some objects in a scene by separating them from each other and from their surroundings (von der Malsburg & Buhmann, 1992; Mohan & Nevatia, 1992). Pattern recognition and scene analysis can be substantially simplified if a good segmentation is available. Up to now, there has been growing interest in the development of new types of segmentation models inspired by biological systems. These models exploit parallel

architecture and have flexible implementation. By turns, traditional techniques are usually based on statistical data and employ serial processing, which suffer from low efficiency and the need for large computational power (Duda, Hart, & Stork, 2001; Jain, Murty, & Flynn, 1999; Pal & Pal, 1993).

In fact, evidence from physiological experiments has been accumulating with strong indications of the existence of synchronous rhythmic activities in different areas of the brain of some mammals, like cats and monkeys (Grey, König, Engel, & Singer, 1989; Murthy & Fetz, 1992). It has been suggested that this neuronal oscillation and synchronization have a role in solving feature binding and scene segmentation problems (König & Schillen, 1994). The processing through synchronous oscillations would be related to temporal coding: An object is represented by temporal correlation of firing activities of spatially distributed neurons (von der Malsburg & Schneider, 1986). Specifically, the solution of scene segmentation under this suggestion can be described by the following rule: The neurons which process different features of the same object have the same underlying frequency of oscillation (synchronization), while neurons which code different objects oscillate with different phases or at random (de-synchronization). This is the so-called *oscillatory correlation* (Terman & Wang, 1995). It has been successfully applied to segmentation of general images (Wang & Terman, 1997; Chen & Wang, 2002; Wang & Liu, 2002), range images (Liu & Wang, 1996) and medical images (Shareef, Wang, & Yagel, 1999). It is also used for motion determination (Cesmeli, Lindsey, & Wang, 2002) and perception (Chen & Wang, 2001). The main difficulty encountered in these kinds of models is dealing with two totally contrary factors at the same time: synchrony and desynchrony. The stronger the synchronization tendency among neurons, the more difficult it is to achieve de-synchronization and vice-verse. Segmentation solutions, then, are tradeoffs between these two tendencies. This situation is called *Synchrony-De-synchrony Dilemma* (Zhao, Macau, & Omar, 2000). The segmentation capacity (number of segments that can be extracted in a given image) is directly related to this dilemma, that is, the capacity will always be limited if the synchrony-desynchrony dilemma cannot be escaped from. This is because the enhancement of a synchronization or de-synchronization tendency will inevitably weaken the other. Usually de-synchronization is weakened because a coherent object should not be broken up. Segmentation capacity is decided exactly by the model's de-synchronization ability since it serves to distinguish one group of synchronized elements (an object) from another. Thus, the segmentation capacity decreases as the de-synchronization tendency is weakened.

In this chapter, we show how the synchrony-desynchrony dilemma can be avoided by applying spatio-temporal chaos and chaotic synchronization—the *Chaotic Oscillatory Correlation* (Zhao, Macau, & Omar, 2000; Zhao & Macau, 2001). Consequently, our oscillator networks have unbounded segmentation capacity. It is worth noting that a model's segmentation capacity is crucial in parallel processing, since segmented patterns may appear at the same instance. Hence, various objects would be indiscriminately grouped if the model's capacity were highly limited.

Pixel clustering tasks can be formulated as follows: Given an image, find the best partition of the pixel space into significant clusters (Duda, Hart, & Stork, 2001; Jain, Murty, & Flynn, 1999; Pal & Pal, 1993). Although in many cases image segmentation and pixel clustering are interchangeable techniques, there is a fundamental difference between them. The former emphasizes topological relations among objects in a given

image. For example, two objects of the same color at different locations should be segmented, while the latter tries to group homogenous pixels without considering geometrical information of images, that is, two or more spatially separated blocks with the same or very similar pixel values are always put into the same cluster. A similar work was shown in Rhouma and Frigui (2001), in which the authors proposed a model of pulse-coupled oscillators for data clustering. Each oscillator fires synchronously with all the others within its group, but the groups themselves fire with a constant phase difference. However, numerical simulations show that our model takes much less iterations to achieve clustering results. This is because the chaotic synchronization employed in our model can be very quickly achieved (Zhao & Macau, 2001). Thus, when the moving process is stabilized, data clusters are already extracted by each synchronized chaotic trajectory. Moreover, due to the model's parallel nature, all elements interact independently, thus the number of iterations needed to form compact groups increases very slowly as the amount of data becomes larger. This feature is especially attractive if the model is implemented in a parallel machine.

The rest of the chapter is organized as follows. Section 2 presents the basic concepts of chaos and chaotic synchronization. Synchronization analyses for networks of continuous flows and discrete maps are also given. Section 3 is devoted to describing the continuous model for image segmentation, while Section 4 presents a discrete model for pixel clustering. Section 5 concludes the chapter.

CHAOS AND CHAOTIC SYNCHRONIZATION

Chaos is a common phenomenon that occurs in many dynamical systems. It can be applied to the universe of almost all things: as large as celestial bodies; as small as atoms and molecules; as rapid as lasers; as slow as a pendulum (Ott, 2002). Qualitatively, chaos can be defined to be aperiodic-bounded dynamics in a deterministic system with sensitive dependence on initial conditions (Ott, 2002). Specifically,

- *Aperiodic* means that the same state is never repeated twice;
- *Bounded* means that on successive iterations the state stays in a finite range and does not approach $\pm \infty$;
- *Deterministic* means that there is a definite rule with no random terms governing the dynamics; and
- *Sensitive dependence on initial conditions* means that two nearby points will drift apart exponentially as time proceeds. This is an essential aspect of chaos.

Chaos describes a particularly complex type of dynamics in nonlinear dynamical systems. Chaotic behavior seems erratic and almost random as if the system is being influenced by noise or is extremely complex. However, it has been found that even very simple nonlinear systems demonstrate remarkably complex behavior. One of the key points is that chaotic behavior, although arising in purely deterministic systems, provides extraordinary problems of predictability. Predictability is impossible for long time term and difficult in short time term. This is a result of exponential divergence of nearby trajectories. The smallest change in the initial conditions (caused by noise/ imprecision) will cause the system to behave in a very different way, both quantitatively

Figure 1. Example of sensitive dependence on initial conditions of chaos. In this simulation, the bifurcation parameter $A = 4.0$. Trajectory represented by solid line runs from $x(0) = 0.2$. Trajectory represented by the dotted line runs from $x(0) = 0.20001$. Two trajectories drift apart just after 15 iterations.

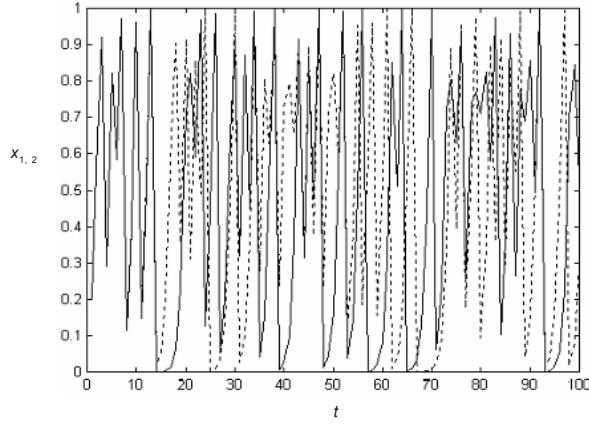
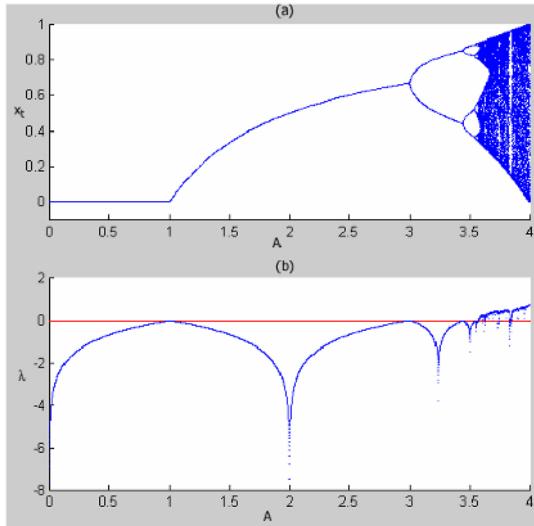


Figure 2. (a) Bifurcation diagram, and (b) Lyapunov exponents of logistic map



and qualitatively. Then, the long-term future behavior of a chaotic system is indeterminable, since we can never be certain of the exact value of the initial condition in any realistic system. Figure 1 shows that two chaotic trajectories from nearby initial conditions do not coincide but diverge from one another rapidly. The simulation is carried out on the logistic map $x(t+1) = Ax(t)(1-x(t))$, where A is the bifurcation parameter.

Figures 2a and 2b show the bifurcation diagram and the Lyapunov exponents of a logistic map. These are the most used methods to characterize chaos. The bifurcation diagram describes changes of dynamics in relation to system parameters. It provides a summary of the essential dynamics of the system. From Figure 2a, we see periodic behavior (columns of only one or a few points), quasi-periodic and chaotic behaviors (columns of many points), the period-doubling route to chaos, interior crisis, intermittency to chaos, periodic windows within chaotic region, etc. (for details, see Ott, 2002).

Lyapunov exponents measure the average rate of divergence between two trajectories as time goes to infinity. Positive values of Lyapunov exponents indicate chaotic dynamics and negative values indicate periodic dynamics. Figure 2b shows the Lyapunov exponents for the logistic map calculated by the following formula:

$$\lambda = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=t_0}^k \ln |f'(x(t))| = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=t_0}^k \ln |A(2x(t)-1)|$$

From this figure, we see that chaotic behavior appears when the parameter A is larger than 3.7, which matches well to the bifurcation diagram.

Because of the sensitive dependence property, chaotic systems seem to defy synchronization. However, it has been discovered that two or more chaotic systems can be synchronized by using a common driving signal or coupled together (Kurths, 2003; Pikovsky, 2001). Synchronization (including chaotic synchronization) is defined as the complete coincidence of the trajectories of the coupled individual dynamical systems in the phase space. In mathematical language, synchronization can be defined as: Given two dynamical systems with state variable vectors $\mathbf{x}(t), \mathbf{y}(t)$, respectively, the two systems are said to be synchronized if $|\mathbf{x}(t) - \mathbf{y}(t)| \rightarrow 0$, as $t \rightarrow \infty$.

Synchronization is an important concept in chaotic systems that has been extensively studied by researchers of applied science, such as electrical and mechanical engineering, biology and laser systems, etc. It also has been successfully used for information coding in communication systems (Kurths, 2003).

The following two subsections are devoted to analyzing the synchronization conditions for networks of chaotic flows and of coupled maps. The obtained results serve as theoretical bases for the image segmentation models presented in this chapter.

Synchronization in a Lattice of Chaotic Flows

Here, we consider the synchronization role in a 2D lattice of coupled continuous chaotic elements with Neumann boundary condition. Each element in the lattice is represented by a set of ordinary differential equations, which have a stable linear part perturbed by a bounded nonlinear function. Consider the following general system:

$$\dot{\mathbf{x}}_{i,j} = A(\mu)\mathbf{x}_{i,j} + f(t, \mathbf{x}_{i,j}, \mu) + k\Delta\mathbf{x}_{i,j} \quad (1)$$

where $1 \leq i \leq N$ and $1 \leq j \leq M$ are indexes of an element in the lattice, $\text{Re}(A(\mu)) < 0$, $\mu \in \Lambda$

a compact set, f is a nonlinear continuous function on $(t, \mathbf{x}_{i,j}, \mu)$, $\|f(t, \mathbf{x}_{i,j}, \mu)\| \leq L_1$, $\left\| \frac{\partial}{\partial \mathbf{x}_{i,j}} f \right\| \leq L_2$

$\left\| \frac{\partial}{\partial x_{i,j}} f \right\| \leq L_2$ for all i, j and L_1 and L_2 are positive constants. $\Delta \mathbf{x}_{i,j}$ is the coupling term and k is the coupling strength. In this case, each element is coupled to its four nearest neighbors. (It is quite straightforward to extend the analysis to a system where each element is coupled to its eight nearest neighbors):

$$\Delta \mathbf{x}_{i,j} = \mathbf{x}_{i,j-1} + \mathbf{x}_{i,j+1} + \mathbf{x}_{i-1,j} + \mathbf{x}_{i+1,j} - 4\mathbf{x}_{i,j}$$

The boundary condition is defined as:

$$\mathbf{x}_{i,0} := \mathbf{x}_{i,1}, \mathbf{x}_{i,M+1} := \mathbf{x}_{i,M}, \mathbf{x}_{0,j} := \mathbf{x}_{1,j}, \mathbf{x}_{N+1,j} := \mathbf{x}_{N,j}$$

We show the following theorem without proof. It gives a sufficient condition to obtain the synchronization of the system defined by equation (1).

Theorem 1. Consider the following synchronization function for equation (1):

$$V = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{M-1} [(\mathbf{x}_{i,j} - \mathbf{x}_{i,j+1})^T (\mathbf{x}_{i,j} - \mathbf{x}_{i,j+1})] + \frac{1}{2} \sum_{j=1}^M \sum_{i=1}^{N-1} [(\mathbf{x}_{i,j} - \mathbf{x}_{i+1,j})^T (\mathbf{x}_{i,j} - \mathbf{x}_{i+1,j})]$$

and assume the following synchronization condition:

$$k > \frac{L_2 + \rho + \|A\|}{4 \left(1 - \cos \frac{\pi}{P} \right)}$$

where $\rho > 0$ is a constant and $P = \max \{M, N\}$. Then:

$$V(t) \leq e^{-\rho t} V(0)$$

The proof of this theorem is in Zhao and Rodrigues (2005) and a similar theorem on synchronization condition in 1D array can be found in Zhao (2003).

This theorem provides sufficient conditions to obtain synchronization of the lattice. This means that any number of elements in such a system can be synchronized by offering a strong enough coupling strength k . The application of the theoretical result for image segmentation will be shown later in the chapter.

Synchronization in Globally Coupled Chaotic Maps

We consider a globally coupled map (GCM) composed of N chaotic maps:

$$x_i(t+1) = (1-\varepsilon)f(x_i(t)) + \frac{\varepsilon}{N} \sum_{j=1}^N f(x_j(t)) \quad (2)$$

where t is a discrete time step, i is the index number, N is the number of elements in the network, $f(x) = Ax(1-x)$ is the logistic map and ϵ is the coupling strength.

This model has triggered much interest in many areas. Kaneko studied this model and showed that it presents very rich and involved phenomena, such as clustering, attractors with dynamical tree structures and non-statistic behaviors, etc. (Kaneko, 1990; Ito & Kaneko, 2000; Ito & Kaneko, 2002; Ouchi & Kaneko, 2000). For our purposes here, we only consider the complete synchronized behavior.

The synchronized attractor $x_1 = x_2 = \dots = x_N = x$ lies along the one-dimensional diagonal in the N -dimensional space. The stability of the synchronized behavior can be evaluated from the product of the Jacobian matrix for equation (2) within this attractor:

$$\mathbf{J} = \prod_{t=1}^m \mathbf{J}_0 f'(x(t))$$

where

$$\mathbf{J}_0 = \begin{pmatrix} 1-\epsilon + \frac{\epsilon}{N} & \frac{\epsilon}{N} & \frac{\epsilon}{N} & \dots & \frac{\epsilon}{N} \\ \frac{\epsilon}{N} & 1-\epsilon + \frac{\epsilon}{N} & \frac{\epsilon}{N} & \dots & \frac{\epsilon}{N} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\epsilon}{N} & \frac{\epsilon}{N} & \frac{\epsilon}{N} & \dots & 1-\epsilon + \frac{\epsilon}{N} \end{pmatrix}_{N \times N}$$

The Lyapunov exponents are calculated by $\lambda_i = \lim_{m \rightarrow \infty} \left(\frac{1}{m} \ln |\mu_i| \right)$, where μ_i is i^{th} eigen value of $\mathbf{J} = (\mathbf{J}_0)^m \prod_{t=1}^m f'(x(t))$. After some algebra, we obtain $\mu_1 = \prod_{t=1}^m f'(x(t))$ and $\mu_i = (1-\epsilon)^m \prod_{t=1}^m f'(x(t))$, $i = 2, 3, \dots, N$. The eigenvector corresponding to μ_1 is given by $\left(\frac{1}{\sqrt{N}}, 1, 1, \dots, 1 \right)^T$. Then, the Lyapunov exponents are $\lambda_1 = \lambda_0$, $\lambda_i = \ln(1-\epsilon) + \lambda_0$, $i = 2, 3, \dots, N$, where λ_0 is the Lyapunov exponent of a single logistic map.

The eigenvector of μ_1 is related to the dynamics of the coupled system on the synchronized subspace, which will not affect the stability of synchronization. The synchronization stability is related to the values of the rest $N-1$ Lyapunov exponents, which accounts for perturbations that are orthogonal to the synchronization subspace embedded in the N -dimensional phase space, that is, the synchronized state is stable if the dynamics along the $N-1$ directions shrink. This is equivalent to the $N-1$ Lyapunov exponents being negative, that is, $\ln(1-\epsilon) + \lambda_0 < 0$. Then, the stability threshold for the synchronized state is $\epsilon_c = 1 - e^{-\lambda_0}$. If $\epsilon > \epsilon_c$, the synchronized state exists and is locally stable, while for coupling strength below the critical value, $\epsilon < \epsilon_c$, the synchronized state loses its stability and bifurcates to a multi-cluster state. For a single logistic map with $A = 4$, the Lyapunov exponent is $\lambda_0 = \ln 2$. Then, the critical coupling strength is $\epsilon_c = 0.5$.

The previous analysis shows that any number of elements in a GCM system can be synchronized by a strong coupling. Moreover, the synchronized trajectory is chaotic too. Later in the chapter, these results will be applied to pixel clustering.

IMAGE SEGMENTATION BY CHAOTIC OSCILLATOR NETWORK

In this section, a network consisting of locally coupled continuous chaotic flows is constructed for image segmentation.

Model Description

The model is a two dimensional network governed by the following equations:

$$\begin{aligned} \dot{x}_{i,j} &= -ax_{i,j} + G(cx_{i,j} + ey_{i,j} + I_{i,j} - \theta_x) + k \sum_{p=i-1}^{i+1} \sum_{q=j-1}^{j+1} H(|c_{p,q} - c_{i,j}| - \Theta)(x_{p,q} - x_{i,j}) \\ \dot{y}_{i,j} &= -bx_{i,j} + G(dx_{i,j} + fy_{i,j} - \theta_y) + k \sum_{p=i-1}^{i+1} \sum_{q=j-1}^{j+1} H(|c_{p,q} - c_{i,j}| - \Theta)(y_{p,q} - y_{i,j}) \\ G(v) &= \frac{1}{1 + e^{-(v/T)}} \\ H(v) &= \begin{cases} 1 & \text{if } v \leq 0 \\ 0 & \text{if } v > 0 \end{cases} \end{aligned} \quad (3)$$

where (i, j) is a lattice point with $1 \leq i \leq N$ and $1 \leq j \leq M$; (p, q) is used as the index of the nearest neighbors of (i, j) ; $x_{i,j}$ and $y_{i,j}$ are dynamical variables of element (i, j) , which will be described below; $c_{i,j}$ is the pixel value of element (i, j) ; and k is the coupling strength. $H(v)$ is a Heaviside function and Θ is a threshold value.

Without the coupling terms, equation (3) is a Wilson-Cowan neural oscillator (Baird, 1986), where a and b are decay parameters (positive numbers) of x and y , respectively; c and f are self-excitatory parameters; e is the strength of coupling from the inhibitory unit y to excitatory unit x (it is a negative value to assure that the variable y acts as inhibitory). The corresponding coupling strength from x to y is given by d ; θ_x and θ_y are thresholds of unit x and y , respectively; $G(v) \in [0, 1]$ is a sigmoid function with T defining its steepness; and I is an external stimuli. If I is a constant, no chaos can appear since it is a two-dimensional continuous flow (Ott, 2002). In order to get a chaotic oscillator, the external stimuli is defined as a periodic function: $I(t) = A\cos(t)$, where A is the amplitude of the driving signal. We see that the interaction terms vanish when the oscillators are synchronized. Thus, the synchronous trajectory remains once the synchronization state is achieved.

The segmentation strategy is described below. Considering a scene image containing p non-overlapped objects, the network is organized such that each element corresponds to a pixel of the image. The parameters can be chosen so that the stimulated oscillators (receiving a proper input, corresponding to a figure pixel) are chaotic. The unstimulated oscillators (receiving zero or a very small input, corresponding to a

background pixel) remain silent ($x_{ij} = 0$). If the similarity of pixel values between two neighbors, say (i,j) and (p,q) , is beyond a predefined threshold Θ , the Heaviside function returns 1 and the coupling between them is maintained. Otherwise, the coupling between (i,j) and (p,q) is cut. If each group of connected, stimulated oscillators are synchronized, then each object is represented by a synchronized chaotic orbit, namely X_1, X_2, \dots, X_p . The dynamics of each synchronized chaotic orbit is qualitatively similar to a single uncoupled oscillator under this coupling scheme (Kurths, Boccaletti, Grebogi, & Lai, 2003; Pikovsky, Rosenblum, & Kurths, 2001). Thus, X_1, X_2, \dots, X_p can be considered as a series of orbits generated by a chaotic system from the same or different initial conditions. Due to the sensitive dependency on initial condition, which is the fundamental characteristic of chaos, if we give different (or random) small perturbations to each trajectory of X_1, X_2, \dots, X_p , i.e., $X_1+d_1, X_2+d_2, \dots, X_p+d_p$, all these chaotic orbits will be exponentially distant from each other after a while. On the other hand, the synchronization state would not be destroyed by a small perturbation if it is an asymptotically stable state. In this way, all the objects in the scene image can be separated. From the above description, one can see that the segmentation mechanism is irrespective to the number of objects in a given scene. Thus, our model has unbounded segmentation capacity. Computer simulations show that objects in a given scene can be separated (their resulting synchronized chaotic trajectories are distinct) even without the perturbation mechanism.

Computer Simulations

Here, the amplitude of external stimulation A is considered as a bifurcation parameter. Other parameters are held constant at: $a = 1.0, b = 0.01, c = 1.0, d = 0.6, e = -2.5, f = 0.0, \theta_x = 0.2$ and $\theta_y = 0.15, T = 0.025, \Theta = 25$ and $c_{ij} \in \{0, 1, 2, \dots, 255\}$. With these parameter settings, chaotic dynamics can appear for each uncoupled oscillator (Zhao, Macau, & Omar, 2000). To show our segmentation method, each stimulated oscillator (corresponding to a figure pixel) receives an external input with amplitude $A = 1.2$. Unstimulated oscillators (corresponding to a background pixel) have $A = 0.0$. So, each stimulated oscillator will be chaotic, while those without stimulation will remain silent.

Now, we discuss the simulation results of the model using the input image shown by Figure 3. This image consists of 20 objects. These patterns are simultaneously presented to a network, where each pixel is represented by an oscillator. The initial

Figure 3. Original gray-level image for segmentation

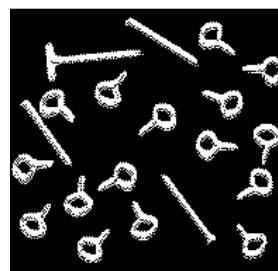
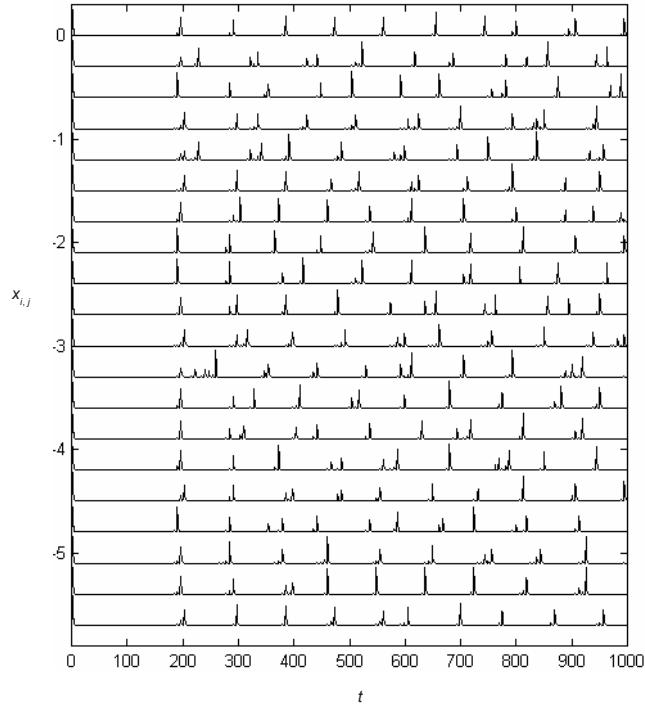


Figure 4. Each synchronized chaotic trajectory corresponds to a segment of image shown by Figure 3



conditions of all oscillators on the grid are set randomly. The coupling strength is $k = 12.0$. Because of such a strong coupling, all connected similar pixels can be synchronized (see Section 2.2). Thus, the elements corresponding to each object in the image form a synchronized chaotic trajectory. Figure 4 shows the temporal activities of the oscillator blocks. We see the appearance of twenty distinct synchronized chaotic orbits, each representing an object.

From Figure 4, we see that different groups of oscillators can reach the active phase at the same time. Therefore, usually we cannot correctly extract objects by only evaluating their activities at an instant. A formal method for extracting objects is to calculate cross-correlation among elements in a time window. However, this is computationally demanding because a large number of elements are involved. Here, we propose a simple method: We have seen from Figure 4 that different chaotic trajectories may cross a predefined Poincaré section simultaneously at some instants. However, because of the sensitive dependency property of chaos, they cannot keep together for a long time. Actually, they separate quickly once they have approached. Then, these chaotic trajectories can be distinguished by observing their second, third or more crossing of the Poincaré section. If a set of oscillators crosses the Poincaré section simultaneously several successive times, they are considered to be a segmented object. For this purpose, a proper time interval is chosen within which each chaotic trajectory

can cross the Poincaré section several times (three or four times are enough). In this way, one object is extracted in each time interval.

PIXEL CLUSTERING BY CHAOTIC OSCILLATOR NETWORK

In this section, a network consisting of coupled chaotic maps for pixel clustering is presented.

Model Description

The general scheme of pixel clustering based on chaotic oscillatory correlation theory can be described as follows: Each element in the network is represented by a chaotic map, which corresponds to an image pixel. When an image is supplied to the network, the elements self-organize according to a predefined similarity criterion, such that each group of elements representing a pixel cluster will be coupled together, while those couplings among different pixel clusters will be eliminated. Consequently, all elements belonging to the same pixel cluster will be synchronized, that is, each pixel cluster will be represented by a synchronized chaotic trajectory. Any number of such chaotic trajectories is already distinguished from each other following the high sensitivity to initial conditions and the dense properties of chaos (Zhao, Macau, & Omar, 2000; Zhao & Macau, 2001).

However, because of ambiguous pixels that stay between various cluster centers in the feature space, the described approach cannot perform general pixel clustering tasks. Usually, elements in the network representing ambiguous pixels may belong to various groups and make them interconnected. Consequently, it is impossible to obtain a synchronized trajectory for each cluster. In order to solve this problem, a mechanism to adaptively move pixel values is introduced (Zhao, Carvalho, & Li, 2004), where each element in the network receives forces from a group of other elements within a certain similarity level. These forces make similar pixels more compact, dissimilar pixels more distinct and ambiguous pixels join the group imposing the greatest forces on it, leaving other groups. In this way, the mechanism of chaotic synchronization can work correctly. Hence, the model is composed of two components: a network of adaptively coupled chaotic maps and a pixel-moving mechanism. The former extracts each pixel cluster and the latter eliminates ambiguous pixels.

The proposed model is a one-dimensional array composed of N chaotic maps, each corresponding to a pixel in a given image, that is, the 2-D input image is first arranged in a 1-D array. Specifically, it is governed by the following equations:

$$x_i(t+1) = (1-\varepsilon)f(x_i(t)) + \frac{\varepsilon}{M_i(t)} \sum_{j=1}^N z_{ij}(t)f(x_j(t)) \quad (4)$$

$$z_{ij}(t+1) = \beta z_{ij}(t) + (1-\beta)H\left(e^{-\alpha\|e_j(t)-e_i(t)\|} - \theta\right) \quad (5)$$

$$c_{ik}(t+1) = \begin{cases} 0 & \text{if } c_{ik}(t) + \eta F_{ik}(t) \leq 0 \\ c_{ik}(t) + \eta F_{ik}(t) & \text{if } 0 < c_{ik}(t) + \eta F_{ik}(t) < 1 \\ 1 & \text{if } c_{ik}(t) + \eta F_{ik}(t) \geq 1 \end{cases} \quad (6)$$

$$\mathbf{F}_i(t) = \frac{\sum_{j \in \Delta_i(t)} \frac{\mathbf{c}_j(t) - \mathbf{c}_i(t)}{\|\mathbf{c}_j(t) - \mathbf{c}_i(t)\|} e^{-\alpha \|\mathbf{c}_j(t) - \mathbf{c}_i(t)\|}}{M_i(t)} \quad (7)$$

$$j \in \Delta_i(t), \text{ if } H\left(e^{-\alpha \|\mathbf{c}_j(t) - \mathbf{c}_i(t)\|} - \theta\right) = 1 \quad (8)$$

$$H(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (9)$$

where i, j are element indexes, $i, j = 1, 2, \dots, N$; $f(x) = Ax(1-x)$ defines a logistic map with A being the bifurcation parameter; and $\mathbf{c}_i(t) = (c_{i1}(t), c_{i2}(t), \dots, c_{iK}(t))$ represents the attribute vector of the i th pixel at iteration t , with K being the number of attributes. Specifically, for a gray-level image, $K = 1$ and c_i is a scalar value representing the intensity of the i^{th} pixel; for a color image, $K = 3$ and $c_i(t)$ is a vector representing the three-color components of the i^{th} pixel; $\mathbf{c}_i(0)$ is the original value of the i^{th} pixel; and $H(v)$ is a Heaviside function. The term $e^{-\alpha \|\mathbf{c}_j(t) - \mathbf{c}_i(t)\|}$ is a Gaussian function, which results in a value between 0 and 1, and has its stiffness controlled by the parameter α and $\|\bullet\|$ is the Euclidean norm. The parameter θ is a threshold, which shifts the Heaviside function. As θ becomes higher, the possibility of the Heaviside function returning the value 1 is reduced. The parameter β controls the integration rate, with $0 \leq \beta < 1$. A low value for β results in rapid adjustments to the returned value of the Heaviside function; a high β value leads to smooth adjustments. $\Delta_i(t)$ is a local pixel set around pixel i ; $\mathbf{F}_i(t) = (F_{i1}(t), F_{i2}(t), \dots, F_{iK}(t))$ represents the total force imposed upon the pixel i from all pixels in $\Delta_i(t)$ at iteration t ; and $M_i(t)$ is

the number of pixels in $\Delta_i(t)$. The vector $\frac{\mathbf{c}_j(t) - \mathbf{c}_i(t)}{\|\mathbf{c}_j(t) - \mathbf{c}_i(t)\|}$ defines the force direction imposed on pixel i from pixel j . Finally, η controls the adjustment rate of $\mathbf{c}_i(t)$.

Without the variables $z_{ij}(t)$ and $\mathbf{c}_i(t)$, equation (4) is equivalent to a Globally Coupled Map (GCM), where all elements in the system can be synchronized by providing the coupling strength $\varepsilon > 0.5$ (See Section 2.3). These two variables incorporate into the system the ability of adaptive pixel clustering.

The pixel clustering process can be described as follows: Initially, the coupling strength ε is set to a value larger than 0.5, such that all elements coupled together will be synchronized. Next, an image vector $\mathbf{c}_i(0)$, $i = 1, 2, \dots, N$, is input to the system. As the system runs, the variable $z_{ij}(t)$ approaches the return value of the Heaviside function in equation (5). The Heaviside function returns a value 1 if the similarity between $\mathbf{c}_i(t)$ and $\mathbf{c}_j(t)$ is beyond a threshold value, which can be adjusted by the parameters α and θ ;

otherwise, it returns a value 0. Actually, the term $H\left(e^{-\alpha\|\mathbf{c}_j(t)-\mathbf{c}_i(t)\|} - \theta\right)$ defines a region $\Delta_i(t)$ of pixel values, from which all pixels j are considered similar to pixel i . Consequently, the coupling between elements i and j is maintained when $z_{ij}(t)$ approaches 1, while the connection is removed when $z_{ij}(t)$ approaches 0. As the system evolves, elements with similar features will be coupled together, and thus form a synchronized chaotic trajectory due to the strong coupling strength. At the same time, elements with very different features will be distributed into different synchronized groups, since there is no connection between them. Because all these chaotic trajectories are generated locally (within a group of coupled elements), they will have distinct temporal activities from each other. This happens because any two chaotic trajectories exponentially drift apart, even if they have only a very small difference at any instant. In this way, the model has unbounded clustering capacity.

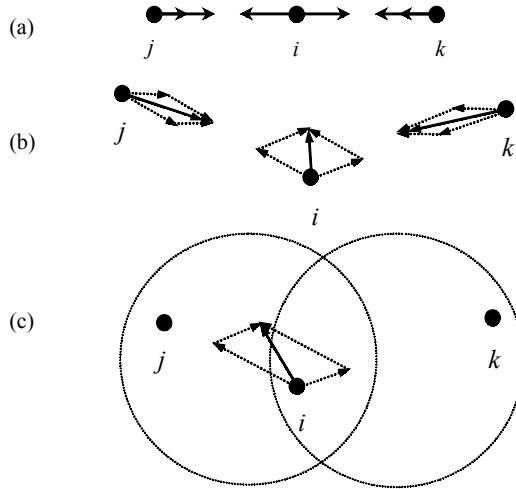
It must be observed that very often some pixel values are grouped into more than one cluster. This ambiguity problem can be solved by an adaptive process defined by equations (6), (7) and (8).

Initially, $\mathbf{c}_i(0)$ is the original value of i th pixel. As the system evolves, each pixel i receives a force of attraction $\mathbf{F}_i(t)$ from a set of similar pixels in $\Delta_i(t)$. This force drives the element to move toward the group with higher similarity. Consider the following three typical cases:

- Figure 5a shows that the elements j and k are on the opposite sides of element i and the distances from j to i and from k to i are the same. In this case, j receives a sum of two forces to the right, so it will move to the right. In a similar way, k will move to the left. However, the two forces received by the element i have equal magnitudes and opposite directions, so they are cancelled. This means that i remains at its original position. Successive iterations will make the three elements increasingly closer. If the moving rate η is sufficiently small, they will match; otherwise, the three elements will oscillate, confined to a small region.
- Figure 5b shows three elements not on the same line, with different distances from j to i and from k to i . From this figure, it is possible to see that the total force received by each element drives it toward the others.
- Figure 5c shows a situation where the element i is in two groups at the same time and the elements j and k are separated in two different groups by hard thresholding. In this case, it is possible to see that i will approach the group of j , finally leaving the group of k , since the force imposed on i by j is larger than that imposed by k . Thus, the ambiguity situation of i can be eliminated.

Without the adaptive moving mechanism, ambiguous pixels, working like bridges, often make dynamical elements from different groups interconnected. In this way, the elements belonging to different groups may synchronize together. Considering again Figure 5c, suppose that elements j and k belong to different clusters. Without moving element i , the elements j and k would be coupled together through i . Then, we may accidentally get synchronized trajectory between j and k . As a result, j and k could not be distinguished.

Figure 5. Illustration of interaction among elements



Generally speaking, elements at the center of a data group will move slowly, since the majority of the forces imposed on them are cancelled, while off-center elements will move toward their respective centers rapidly. Ambiguous elements (those that receive forces from more than one group) will leave other groups and fix themselves in only one group, which has the strongest attraction to them. Thus, successive iterations will decrease pixel value distances between similar pixels (decrease intra-class distance) and increase the distance between very different pixels (increase inter-class distance). If the movement rate η is sufficiently small ($\eta \rightarrow 0$), each group of $c_i(t)$ will approximate and be confined into a small region.

In comparison with conventional pixel clustering techniques, this model offers the following interesting characteristics:

1. It is not necessary to know the number of pixel clusters beforehand;
2. The model has unbounded capacity of pixel clustering, that is, any number of chaotic trajectories, each representing a pixel cluster, can be distinguished; and
3. The mechanism of adaptive modification of pixel values makes the model robust enough to cluster ambiguous elements.

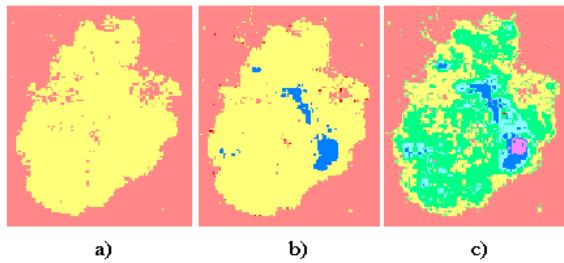
Computer Experiments

Computer simulation results presented in this section indicate that the clustering process is not very sensitive to the parameters β and η , but is strongly dependent on the parameters T and θ . The parameter ε should be kept larger than 0.5 in order to guarantee the synchronization among elements from the same group (Kaneko, 1990; Zhao & Macau, 2001). The parameter α controls the stiffness of the Gaussian function in Equation (5) and θ is a hard threshold. A small value of α smoothes the difference between pixels, resulting in a small number of pixel clusters, while a large value of α amplifies the difference, leading

Figure 6. Original image (104×123) for pixel clustering



Figure 7. Clustering results: (a) 2 clusters, $\theta = 0.55$; (b) 4 clusters, $\theta = 0.7$; (c) 9 clusters, $\theta = 0.86$



to a large number of pixel clusters. A large value of θ reduces the coupling between elements, generating more clusters, while a small value of θ results in a large coupling range, consequently producing less pixel clusters. In practical image processing applications, details of the original image can be amplified by setting a large value to θ . On the other hand, if only skeletons of the objects are needed, a small value can be set.

In order to show the complete pixel clustering process performed by the proposed model, consider Figure 6, which shows the human skin structure as an input image. The image is encoded by using RGB. Here, the hierarchical representation effect is obtained by only changing θ . Other parameters are held constant at $a = 4.0$, $\epsilon = 0.6$, $\alpha = 0.8$, $\beta = 0.1$ and $\eta = 0.01$. Figure 7 shows the pixel clustering results in three different resolutions. As previously observed, when θ is small, a clustering with few pixel groups is obtained. Figure 7(a) shows the 2 clusters produced. In this case, one can see that the background and the object are separated. As θ increases, clustering with more pixel groups is achieved. Figure 7(b) and 7(c) show cases where 4 and 9 clusters are produced, respectively. In these figures, it is possible to see some detail within the object.

Figure 8(a) shows the time series of all elements in the network corresponding to the clustering result of Figure 7(a). Initially, the trajectories of each group of elements are twisted together. This occurs because some elements belong to more than one cluster and, consequently, different groups are interconnected. As the system evolves, all such elements leave their least attractive group and go to their most attractive one. In this way,

Figure 8. Evolution of chaotic maps corresponding to the simulation shown by Figure 7(a): (a) 100 time units; (b) a section of (a)

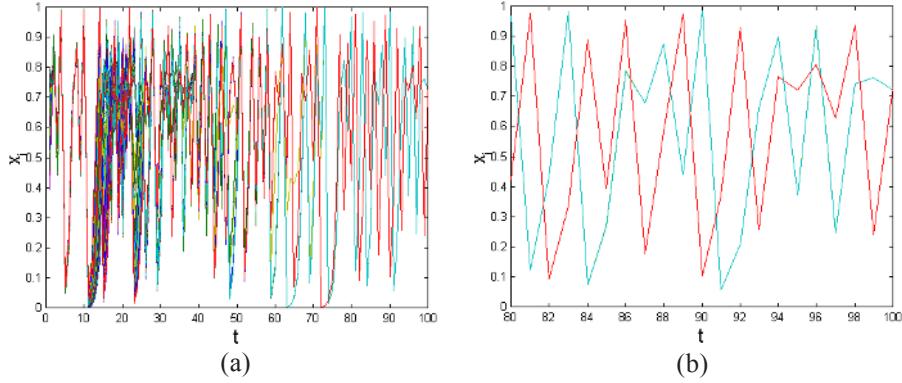
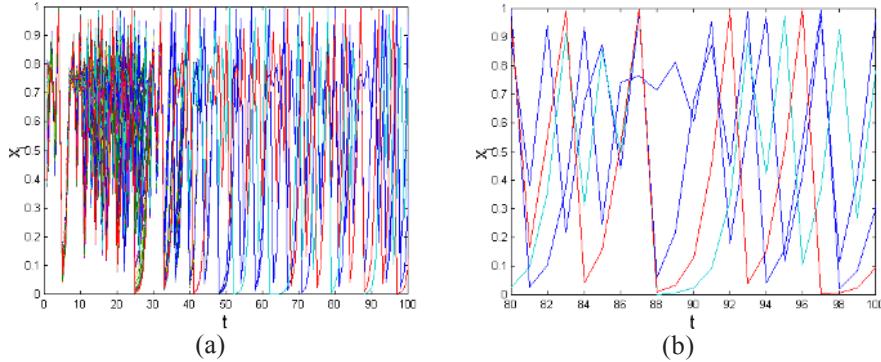


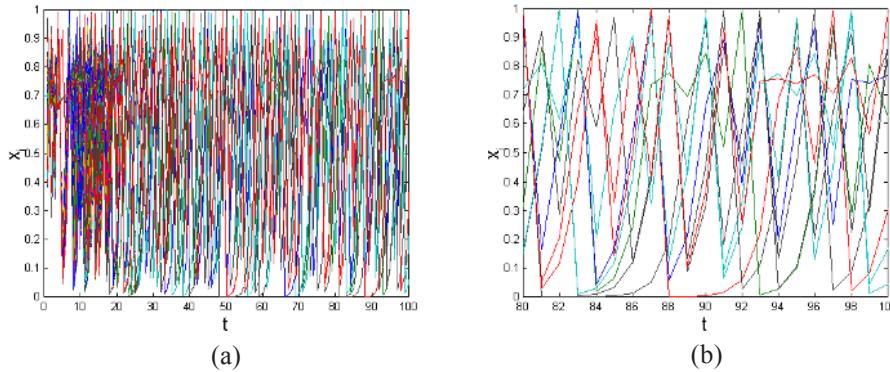
Figure 9. Evolution of chaotic maps corresponding to the simulation shown by Figure 7(b): (a) 100 time units; (b) a section of (a)



overlaps among groups are eliminated and, consequently, pixel clusters are correctly formed. From Figure 8(b), a section of Figure 8(a), the formation of two distinct trajectories, each representing a pixel cluster, can be seen. Figure 9 shows the formation of four synchronized trajectories corresponding to Figure 7(b) and Figure 10 shows the formation of nine synchronized trajectories corresponding to Figure 7(c).

With a predefined similarity function, data points in the feature space are mapped to the coupling structure of the model. The couplings among distant data points (under the similarity measure) are cut down, while the connections among nearby data points are maintained. Ambiguity is solved by the data moving mechanism. In this way, the model represents consistently the inherent structure of input data. Consequently, significant clustering results can be generated. As described in the text, there are five parameters in the model. Three of them (β , η and ϵ) can be easily fixed for almost all applications. Thus, we need to tune only two parameters (θ and T). In general, there is

Figure 10. Evolution of chaotic maps corresponding to the simulation shown by Figure 7(c): (a) 100 time units; (b) a section of (a)



not a unique representation in data clustering problems, that is, a same data set can be interpreted by different meaningful clustering results. By using this model, one can get hierarchical meaningful results by tuning a very small number of parameters. Moreover, due to the model's self-organizing feature, together with parameter tuning, neither the cluster number nor the guessing of initial cluster centers is needed. Thus, combinatorial search can be avoided.

CONCLUSIONS

Because of the sensitive dependency on initial conditions of the chaotic time evolution and the fact that a chaotic trajectory is dense in its invariant set, any number of chaotic trajectories can be easily distinguished; on the other hand, due to their deterministic nature, chaotic trajectories can be synchronized. Thus, chaos is a suitable solution to form data groups by synchronization, while at the same time, to distinguish one group from another by desynchronization. Naturally, this idea can be applied not only to image segmentation, but also to solve more general problems, such as data clustering.

It is also worth noting that synchronization is a temporal feature of neuron activities. It is stimuli-evoked and irrespective of initial conditions. Thus, synchronized oscillation is a suitable solution to code temporal information and thus can be used in video segmentation or motion detection.

Network is ubiquitous in our everyday life (Newman, 2003). The advantage of using network structure for image segmentation is that it can obtain not only overall statistical properties of images, but also local features through interactions among elements of the network. Moreover, the evolution of dynamical systems as network elements permits control and other adaptive mechanisms to be applied during segmentation. Thus, the theory of nonlinear dynamics together with the theory of complex networks may make a considerable contribution to the study of image processing. The work presented here can be considered as an endeavor toward this direction.

ACKNOWLEDGMENT

The author would like to thank FAPESP, a Brazilian Fund Agency, for its support.

REFERENCES

- Baird, B. (1986). Nonlinear dynamics of pattern formation and pattern recognition in the rabbit olfactory bulb. *Physica 22D*, 150-175.
- Cesmeli, E., Lindsey, D. T., & Wang, D. L. (2002). An oscillatory correlation model of visual motion analysis. *Perception & Psychophysics*, 64, 1191-1217.
- Chen, K., & Wang, D. L. (2001). Perceiving geometric patterns: From spirals to inside/outside relations. *IEEE Transactions on Neural Networks*, 12, 1084-1102.
- Chen, K., & Wang, D. L. (2002). A dynamically coupled neural oscillator network for image segmentation. *Neural Networks*, 15, 423-439.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: John Wiley & Sons.
- Grey, C. M., König, P., Engel, A. K., & Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338, 334-337.
- Ito, J., & Kaneko, K. (2000). Self-organized hierarchical structure in a plastic network of chaotic units. *Neural Networks*, 13, 275-281.
- Ito, J., & Kaneko, K. (2002). Spontaneous structure formation in a network of chaotic units with variable connection strengths. *Physical Review Letters*, 88, 028701(1-4).
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3), 264-323.
- Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and coding in a network of chaotic elements. *Physica D*, 41, 137-172.
- König, P., & Schillen, T. B. (1994). Binding by temporal structure in multiple feature domains of an oscillatory neuronal network. *Biological Cybernetics*, 70, 397-405.
- Kurths, J., Boccaletti, S., Grebogi, C., & Lai, Y-C. (2003). Chaos (Focused issue), 13(126).
- Liu, X., & Wang, D. L. (1999). Range image segmentation using a LEGION network. *IEEE Transactions on Neural Networks*, 10, 564-573.
- Mohan, R., & Nevatia, R. (1992). Perceptual organization for scene segmentation and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6), 616-635.
- Murthy, V. N., & Fetz, E. E. (1992). Coherent 25- to 35-Hz oscillations in the sensorimotor cortex of awake behaving monkeys. *Proceedings of the National Academy of Science USA*, 89 (pp. 5670-5674).
- Newman, M. E. J. (2003). The structure and function of complex networks. *Society for Industrial & Applied Mathematics Review*, 45(2), 167-256.
- Ott, E. (2002). *Chaos in dynamical systems* (2nd ed.). New York: Cambridge University Press.
- Ouchi, N. B., & Kaneko, K. (2000). Coupled map with local and global interactions. *Chaos*, 10, 359-365.

- Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26(9), 1277-1294.
- Pikovsky, A., Rosenblum, M., & Kurths, J. (2001). *Synchronization—A universal concept in nonlinear sciences*. New York: Cambridge University Press.
- Rhouma, M. B. H., & Frigui, H. (2001). Self-organization of pulse-coupled oscillators with application to clustering. *IEEE Transactions on Neural Networks*, 23, 180-195.
- Shareef, N., Wang, D. L., & R. Yagel (1999). Segmentation of medical images using LEGION. *IEEE Transactions on Medical Imaging*, 18, 74-91.
- Terman, D., & Wang, D. L. (1995). Global competition and local cooperation in a network of neural oscillators. *Physica D*, 81, 148-176.
- von der Malsburg, Ch., & Buhmann, J. (1992). Sensory segmentation with coupled neural oscillators. *Biological Cybernetics*, 67, 233-242.
- von der Malsburg, Ch., & Schneider, W. (1986). A neural cocktail-party processor. *Biological Cybernetics*, 54, 29-40.
- Wang, D. L., & Liu, X. (2002). Scene analysis by integrating primitive segmentation and associative memory. *IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics*, 32, 254-268.
- Wang, D. L., & Terman, D. (1997). Image segmentation based on oscillatory correlation. *Neural Computation*, 9, 805-836.
- Zhao, L. (2003). Robust chaotic synchronization for scene segmentation. *International Journal of Modern Physics B*, 17(22), 4387-4394.
- Zhao, L., Carvalho, A. C. P. L. F., & Li, Z-H. (2004). Pixel clustering by adaptive moving and chaotic synchronization. *IEEE Transactions on Neural Networks*, 15(5), 1176-1185.
- Zhao, L., & Macau, E. E. N. (2001). A network of globally coupled chaotic maps for scene segmentation. *IEEE Transactions on Neural Networks*, 12(6), 1375-1385.
- Zhao, L., Macau, E. E. N., & Omar, N. (2000). Scene segmentation of the chaotic oscillator network. *International Journal of Bifurcation and Chaos*, 10(7), 1697-1708.
- Zhao, L., & Rodrigues, H. M. (in press). Chaotic synchronization in 2D lattice for scene segmentation, *IEEE Trans. Neural Networks*.

Section III: Video Segmentation

Chapter VI

Joint Space-Time-Range Mean Shift-Based Image and Video Segmentation

Irene Yu-Hua Gu, Chalmers University of Technology, Sweden

Vasile Gui, Technical University Timisoara, Romania

ABSTRACT

This chapter addresses image and video segmentation by using mean shift-based filtering and segmentation. Mean shift is an effective and elegant method to directly seek the local modes (or, local maxima) of the probability density function without the requirement of actually estimating it. Mean shift is proportional to the normalized density gradient estimate, and is pointing to the local stationary point (or, local mode) of the density estimate at which it converges. A mean shift filter can be related to a domain filter, a range filter or a bilateral filter depending on the variable setting in the kernel, and also has its own strength due to its flexibility and statistical basis. In this chapter a variety of mean shift filtering approaches are described for image/video segmentation and nonlinear edge-preserving image smoothing. A joint space-time-range domain mean shift-based video segmentation approach is presented. Segmentation of moving/static objects/background is obtained through inter-frame mode-matching in consecutive frames and motion vector mode estimation. Newly appearing objects/regions in the current frame due to new foreground objects or uncovered background regions are segmented by intra-frame mode estimation. Examples of image/video segmentation are included to demonstrate the effectiveness and robustness of these methods. Pseudo codes of the algorithms are also included.

INTRODUCTION

There has been an increasing interest in object-based video segmentation largely due to multimedia applications, MPEG video coding standards, content-based image retrieval and representation, virtual reality and video surveillance and tracking (Koprinska, 2001; Li, 2003; <http://www-vrl.umich.edu>). It is desirable that image and video segmentation generate partitions of images consisting of semantically meaningful entities such as regions, objects (e.g., balls, flowers, persons or cars) or object parts. While simple objects are often segmented automatically by grouping pixels with similar or homogeneous low-level image features, segmentation of complex objects consisting of multiple regions often requires some kind of supervised segmentation, as there is a gap between the low-level image features and high-level semantics and human perception. This means that additional information is required, either provided by the users or by higher level modules, to make semantic inferences from the low-level results.

Compared with 2D image segmentation, video segmentation has additional features related to object motion (e.g., homogeneity in speed, direction and acceleration). Exploiting these features in the pixel region, or object level, often makes the video segmentation and tracking less challenging (Stauffer, 2000; Li, 2004).

There exists a large amount of survey literature for image/video segmentation techniques (Fu, 1981; Koprinska, 2001; Zhang, 1996; Cheng, 2001). Segmentation methods are often divided into categories although the division rationale varies. A common way to categorize segmentation methods is according to the source of attributes which the segmentation is based upon, e.g., pixel-based, region-based, content-based, edge-based, object-based, semantic-based and many more. Pixel-based segmentation methods use low-level image features such as color, intensity, texture, motion, optical flow, depth and disparity from each individual pixel. Region-based segmentation methods employ multi-scale shape, edges, polygon boundaries and texture statistics for extracting regions of interest (Pauwels, 1999). Object-based methods employ features associated with individual objects. These features can be extracted from different levels (Li, 2003 & 2004; Pan, 2000; Meier, 1998). Frequently, the boundaries among different categories of methods may become blurred, especially when hybrid methods are applied. One may also classify segmentation methods based on whether the method is fully automatic or semiautomatic. Typical examples of semiautomatic methods include a user-interactive process, e.g., active contours (or snakes) and image retrieval. In active contour-based segmentation, a user is usually required to select some initial points close to the contour of a region (Kass, 1987; Xu, 1998). In hybrid multi-layer image retrievals, a user may add some high-level semantics and make choices from the low-level segmentation results (Zhang, 2004; Wang, 2004; Sun, 2003). A user-interactive process usually can achieve better segmentation with the input of high-level semantics and intelligent choices from the human perception.

Another way is to divide segmentation methods into categories according to whether or not a (underlying) model is employed. For example, Gaussian mixture models (GMMs) are frequently used for video object segmentation and tracking. To estimate the unknown parameters in the model, one may use expectation maximization (EM) or Bayesian learning (Stauffer, 2000; Li, 2004). Stochastic modeling, such as Markov random field (MRF)/Gibbs distributions and Bayesian inferences, is frequently used for the segmentation of textures and complex images (Geman, 1984; Zhu, 1998). MRFs are

known to be equivalent to Gibbs distributions, which are associated with an energy minimization problem. A variety of energy minimization-based segmentation methods can be found in the literature (see <http://www.emmcvpr.org>). Most of these stochastic methods require probability density estimates. They are robust in performance, although often computationally intensive. Sometimes, physical models can be employed for image segmentation, for example, thermal models for different materials can be exploited for segmenting target objects and background clutter in infrared images (Lundberg, 2001) and reflection models can be utilized for color image segmentation (Klinker, 1988). Apart from using models, non-model-based methods are still widely used, especially for unsupervised segmentation. Examples include adaptive clustering, for example, possibilistic C-means, fuzzy C-means and their variants (Jain, 1998; Dave, 1997). These unsupervised clustering techniques usually require some validation of the segmented regions as a post-processing. Region growing for segmentation may enable the incorporation of *a priori* information, for example, seeded region growing for supervised video object segmentation (Adams, 1994). Seeded region growing is related to the morphologically inspired watershed segmentation, where seeds are defined as pixels representing the local minima of a morphological gradient (Vincent, 1991). It starts with a major image simplification step aimed at eliminating unimportant details. This will lead to relatively few regions with more regular boundaries. These boundaries are often related to real image edges that are desirable in image segmentation. Region merging is commonly used as a post-processing to reduce the number of regions generated from the watershed segmentation. Starting from an over-segmented image, adjacent regions are successively merged by using statistical and edge information (Zhu, 1996). Since the watershed segmentation uses morphological processing combined with region growing and edge detection techniques, the segmented regions usually well preserve the image edges despite the over-segmentation. More techniques exist for edge-preserving smoothing and segmentation that may lead to semantic meaningful segmentation. Among them, mean shift, bilateral filter and nonlinear diffusion are becoming an increasingly important set of segmentation methods due to their powerful paradigm of nonlinear processing (Cheng, 1995; Tomasi, 1998; Van de Weijer, 2001; Comaniciu, 2002; Barash, 2002; Pollak, 2000; DeMenthon, 2002). As the main theme of this chapter is to address a variety of mean shift filters, some related background of mean shift filters will be further reviewed in this section.

A large part of the recent segmentation work is concentrated on videos. For video segmentation, the methods can be divided into three types according to the strategies applied: In the first type, segmentation starts from the spatial-domain followed by tracking along the temporal direction. The tracking is usually based on the temporal coherence between some spatial regions (Moscheni, 1998; Wren, 1997; Khan, 2001). In the second type, trajectories of points, or regions of interest, are extracted based on the motion similarity along the temporal direction. Trajectories may then be grouped, for example, according to similarity in motion, constraints in the spatial location and the lengths of trajectories (Megret, 2002), in the models (Torr, 1998) or by K-means clustering (Allmen, 1993). In the third type, segmentation is directly applied to the 3D spatiotemporal pixel volume of an image sequence, so that the evidence of similarity is collected in the joint domains without favoring one dimension over another (DeMenthon, 2002; Greenspan, 2004).

Related Work

Mean shift for mode seeking and clustering was initially proposed by Fukunaga and Cheng (Fukunaga, 1975; Cheng, 1995), followed by some major developments and extensions made by Comaniciu and Meer (Comaniciu, 2002). Since then, there is much new development and reported work on mean shift theories and various applications to image and video segmentation (Van de Weijer, 2001; Comaniciu, 2002; Wang, 2004; DeMenthon, 2002). One attraction of mean shift is the statistical basis and its association with the density estimate (Huber, 1981). Often, we are only interested in finding the local modes (or, local maxima) rather than the entire density distribution (e.g., when using the maximum *a posteriori* criterion). Mean shift provides an effective and elegant approach to directly estimate the local modes (maxima) without the requirement of actually estimating the density function. When considering statistical modelling of an image whose features are described by a non-parametric multi-mode probability density function (estimated empirically using kernel functions), mean shift is a powerful tool to seek the local modes in the density function. Mean shift-based image/video segmentation is based on the fact that pixels in the same region share some similar modes. Depending on the feature setting, regions with different types of similarity (e.g., homogeneity in intensities, colors or texture feature attributes) can be estimated.

Recent studies have shown that mean shift is related to nonlinear diffusions and bilateral filtering (Barash, 2002). In physics, a diffusion process governs the transport of heat, matter or momentum leading to an ever increasing equalization of spatial concentration differences. Analogous to this, blurring an image to a controllable degree is considered as being generated by a diffusion process in a scale space. Image intensities are the concentration of a scalar property (Weickert, 1998). In the nonlinear diffusion proposed by Pollak (Pollak, 2000), filter kernel values are inversely proportional to the magnitude of the intensity gradient in a currently processing location. Nonlinear diffusion is known to be able to yield sharp boundaries that separate homogeneous regions, however the computational cost is usually very high. A bilateral filter, on the other hand, combines a domain filter and a range filter so that both the geometrical closeness and the photometric similarity of an image can be taken into account during image filtering and segmentation (Tomasi, 1998). By defining different range features, a bilateral filter may obtain image segmentation based on the homogeneity of different image attributes. While a mean shift filter is related to both the bilateral filter and nonlinear diffusion, it has an additional flexibility since the data under the kernel change during the mean shift iterations.

Much of the recent development in mean shift filter is motivated by the multimedia applications. For example, video segmentation may be performed by firstly applying a 2D mean shift segmentation, followed by motion tracking and spatial temporal integration (Comaniciu, 2000). In video paintbox, 2D mean shift segmentation is performed on some key image frames. Associations are then created between segments according to the color, shape and location of segments to obtain video segmentation and painting (Collomosse, 2005). However, such spatial segmentation followed by temporal tracking may lead to favouring one dimension over another. Apart from this, the 2D mean shift segmentation may yield rather noisy results along the temporal direction. Alternatively, video segmentation may employ a mean shift filter to video volume so as to mitigate these problems. DeMenthon and Megret (DeMenthon, 2002) proposed to use seven-dimen-

sional (7D) feature vectors including color, time, motion and position-related features. Each video volume is then considered as a collection of three feature vectors, and clustered to obtain a consistent moving object by using a hierarchical mean shift filter. Further extension was proposed by Wang (Wang, 2004), where anisotropic kernels are applied to the mean shift filter, and the criterion of spatio-temporal consistent motion of objects is applied. To obtain good performance, user specification of video semantics may also be added. For example, in the video toonering, a user may indicate the boundary points of a region using the results from a low-level segmentation in some key frames. These boundaries can then be used to create 3D unions of mean shift segments that belong to a region (Wang, 2004). While mean shift segmentation to video volume seems somewhat attractive, the computational cost increases significantly as the size of data sets under video volume becomes large. If we consider that the joint probability density function (pdf) of the image sequence is independent, or nearly independent between the spatial and the temporal domain, the joint pdf for a 3D video volume becomes the product of two pdfs. Hence, the mean shift segmentation of video volumes can be simplified without favouring one dimension over another. We shall describe a novel joint space-time-range mean shift filter for video segmentation that is inspired by such an idea. The combined process of inter-frame mode matching and intra-frame mode estimation in the joint space-time-range domain enables the segmentation of moving objects/clutter as well as prevents the newly appeared objects/clutter from removing the occlusion.

In the remaining part of the chapter, we first describe the basic theory of mean shift. Mean shift is shown to be proportional to the normalized density gradient estimate, and is pointing to the stationary point (mode) of the density estimate at which it converges. Two types of commonly used kernels are described. We then describe the settings of different kernel variables, which relate a mean shift filter to a domain filter, a range filter and a bilateral filter, respectively.

We then describe two main applications of mean shift-based filters: image/video segmentation and nonlinear edge-preserving image smoothing. Segmentation can be considered as the partition of an image into geometrically connected regions, each of which shares a certain type of homogeneity or similarity. Depending on the definition of homogeneity or similarity, e.g., pixel intensities, statistical attributes or color attributes of pixels, one may segment an image according to a designed purpose. Examples are included for 2D color image segmentation in the joint space-range domain. For video segmentation, we describe a novel method that is based on the joint space-time-range adaptive mean shift filtering and segmentation. Segmentation of moving/static objects/background is achieved through inter-frame mode-matching in consecutive frames and motion vector mode estimation. Newly appearing objects / regions due to new foreground objects or uncovered background are segmented by intra-frame mode estimation. The method is shown to be effective in video segmentation. Finally, some video segmentation examples are included to demonstrate the method.

BASIC THEORY

One way to characterize an image (or image feature space) is to use probability distributions. As a pdf(probability density function) may have many modes (i.e., multiple local maxima) and unknown shape, we may use a non-parametric method to estimate the

pdf empirically. In many applications, instead of estimating the exact density function we may only need to estimate the associated local modes (i.e., the local maxima) of the function, or the density gradient. Mean shift is an effective and elegant approach to finding the modes of density estimate without the requirement of actually estimating the function. The mean shift vector is found to be proportional to the normalized density gradient and is pointing towards the direction of maximum increase in the density (i.e., towards the local mode). Mean shift can be used in many applications, e.g., nonlinear edge-preserving smoothing filtering, image segmentation and clustering, apart from being used for seeking modes and estimating probability density functions.

Mean Shift for Finding the Mode of Density Estimate

Let a given set of feature vectors be $S = \{\mathbf{x}_i, i = 1 \dots n\}$, where \mathbf{x}_i is a L-dimensional feature vector. Assuming the estimated kernel density has the following form:

$$\hat{p}_K(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K_H(\mathbf{x} - \mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^n \frac{1}{n|\mathbf{H}|^{1/2}} K(\mathbf{H}^{-1/2} d(\mathbf{x}, \mathbf{x}_i, \mathbf{H})) \quad (1)$$

where K is the L-dimensional kernel, and \mathbf{H} denotes the bandwidth matrix of size L, and d is a distance function between data (or, feature) vectors \mathbf{x} and \mathbf{x}_i . For example, one may define the distance function as $d(\mathbf{x}, \mathbf{x}_i, \mathbf{H}) = |\mathbf{H}|^{-1/2} \|\mathbf{x} - \mathbf{x}_i\|$. For *radial symmetric* kernels and Euclidean distance or L_2 norm, the kernel density estimate in equation (1) can be described in the following general form:

$$\hat{p}_K(\mathbf{x}) = \frac{1}{nh^L} \sum_{i=1}^n K\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) = \frac{c_k}{nh^L} \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (2)$$

where $\mathbf{H} = h^2 \mathbf{I}$, and c_k is a normalization constant. It is worthwhile to mention that a radial symmetric kernel assumes an isotropic multivariate \mathbf{x} . One can set different bandwidths for different components of \mathbf{x} to cope with the anisotropic feature space. The mode of the estimated density function $\hat{p}_K(\mathbf{x})$ can be obtained by taking the gradient to equation (2) and setting it to zero:

$$\nabla \hat{p}_K(\mathbf{x}) = \frac{2c_k}{nh^{L+2}} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) k'\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (3)$$

where $k'(\cdot)$ is the first derivative of $k(\cdot)$. Set $G(\mathbf{x}) = c_g g(\|\mathbf{x}\|^2)$, $K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2)$, where g (or, k) is called the *profile* of G (or K), and:

$$g(x) = -k'(x) \quad (4)$$

The kernel K is called to be the *shadow* of kernel G when equation (4) is satisfied.

Substituting the shadow kernel into equation (3) and denoting $\hat{p}_G(\mathbf{x}) = \frac{c_g}{nh^L} \sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$, it follows that:

$$\nabla \hat{p}_K(\mathbf{x}) = \frac{2}{h^2} \frac{c_k}{nh^L} \sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) - \frac{2\mathbf{x}c_k}{h^2 c_g} \hat{p}_G(\mathbf{x}) \quad (5)$$

After some manipulations, equation (5) can be put in the form:

$$\frac{1}{2} h^2 c \frac{\nabla \hat{p}_K(\mathbf{x})}{\hat{p}_G(\mathbf{x})} = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (6)$$

where $c = \frac{c_g}{c_k}$ is a constant. The right hand side of equation (6) is defined as isotropic *mean shift* (or, in short, mean shift in this chapter) and is denoted as:

$$m_G(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (7)$$

where G indicates that the mean shift is computed with the kernel G. Substituting equation (7) into equation (6) yields:

$$m_G(\mathbf{x}) = \frac{1}{2} h^2 c \frac{\nabla \hat{p}_K(\mathbf{x})}{\hat{p}_G(\mathbf{x})} \quad (8)$$

It is important to mention that equation (8) implies that the mean shift computed with kernel G is proportional to the normalized density gradient estimate obtained with kernel K, where K is the shadow kernel of G. Mean shift is a vector that can be considered as the steepest ascent. Its magnitude adaptively decreases to guarantee the convergence, and its direction points to the nearest stationary point (local mode) of the density function estimate. Equation (7) gives the formula for computing mean shift once a kernel is selected. It is also important to notice that equation (8) implies that one may find the modes of $\hat{p}(\mathbf{x})$ without actually knowing $\hat{p}(\mathbf{x})$ itself. The mode can be obtained by finding the convergence points of the mean shift using equation (7).

Iterative Algorithm for Mean Shift and Mode Seeking

One can use an iterative algorithm to compute the mean shift in equation (8). Let $S = \{\mathbf{x}_i, i=1,2,\dots,n\}$ be the given data/feature set, and $\{\mathbf{y}_j, j=1,2,\dots\}$ be the successive centre locations of kernel G originally set as \mathbf{x} . Let $\mathbf{y}_1 = \mathbf{x}$, then the j -th mean shift vector is computed iteratively from the $(j-1)$ -th vector as follows:

$$m_G(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\frac{\|\mathbf{y}_j - \mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^n g\left(\frac{\|\mathbf{y}_j - \mathbf{x}_i\|^2}{h}\right)} - \mathbf{y}_j \quad (9)$$

The magnitude of the above mean shift will converge to zero, which is equivalent to monotonically climbing the hill of the estimated density function until it reaches the maximum point, or mode. Assume we have $\lim_{j \rightarrow \infty} \mathbf{y}_j = \mathbf{y}_c$ after the mean shift converges. Then \mathbf{y}_c is the point where the kernel density estimate $\hat{p}(\mathbf{x})$ reaches its local mode $\hat{p}(\mathbf{y}_c)$. The algorithm is summarized in Table 1.

Kernel Selection

Kernel selection is an essential issue, especially for mean shift-based filtering. In image processing applications, mean shift-based filtering is mainly used as a nonlinear edge-preserving smoothing filter. The filter can be considered as a nonlinear lowpass filter. Two types of kernels are frequently used for computing the mean shift. One is the Epanechnikov kernel that can be obtained from using the minimum mean integrated square error (MISE) criterion:

Table 1. Pseudo codes for mean shift iteration and mode seeking

Algorithm 1: Iterative algorithm for computing the mean shift. Given data / feature set $S = \{\mathbf{x}_i, i=1,2,\dots,n\}$ Step 1: Set the initial centre location of the kernel as $\mathbf{y}_1 = \mathbf{x}$ Step 2: $j \leftarrow j + 1$: compute the new centre: $\mathbf{y}_{j+1} = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\frac{\ \mathbf{y}_j - \mathbf{x}_i\ ^2}{h}\right)}{\sum_{i=1}^n g\left(\frac{\ \mathbf{y}_j - \mathbf{x}_i\ ^2}{h}\right)}$ Step 3: Compute the mean shift $m_G(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j$ Step 4. Repeat Step 2 and Step 3 until the mean shift converges (i.e., $\ m_G(\mathbf{y}_j)\ = \ \mathbf{y}_{j+1} - \mathbf{y}_j\ < \varepsilon$). Step 5: Set the mode of the density estimate at x as: $\mathbf{y}_c \leftarrow \mathbf{y}_{j+1}$.	
---	--

$$K_E(\|\mathbf{x}\|^2) = \begin{cases} \frac{(L+2)}{2c_L}(1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where c_L is the volume of the unit L-dimensional sphere. Comparing equation (10) with equation (2), we obtain the so-called profile $k(x)$ as:

$$k_E(x) = \begin{cases} 1-x & \text{if } 0 \leq x \leq 1 \\ 0 & x > 1 \end{cases} \quad (11)$$

where K_E is the shadow of G_E whose profile is:

$$g_E(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & x > 1 \end{cases} \quad (12)$$

That is, the shadow kernel for G_E with a profile g_E , which is a uniform kernel within a L-dimensional unit sphere, is the Epanechnikov kernel.

Another type of kernel is the multivariate Gaussian kernel:

$$K_N(\|\mathbf{x}\|^2) = \frac{1}{(2\pi\sigma^2)^{L/2}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \quad (13)$$

The profile of the Gaussian kernel in equation (13) is:

$$k_N(x) = \exp\left(-\frac{x}{2}\right) \quad (14)$$

Obviously the shadow kernel for a Gaussian kernel is also a Gaussian kernel. To reduce the computation, a kernel with an infinite length (e.g., Gaussian kernel), is often truncated to yield a kernel with finite support. If the truncated length is correspondent to the kernel bandwidth h , then the error introduced by the truncation is negligible.

Isotropic and Anisotropic Kernels

It is notable that a radial symmetric kernel (i.e., all feature components have the same type of kernel and bandwidth) is associated with a mean shift of an isotropic kernel. However, kernels are in general non-symmetric along different dimensions of the data space. Such a mean shift filter is associated with an anisotropic kernel (Wang, 2004). When the data has a non uniform variance, using an anisotropic kernel mean shift filter is equivalent to using an isotropic kernel mean shift filter with a linear transformation (for data pre-whitening) and the inverse transformation inserted before and after the filter. To limit the scope of this chapter, we will concentrate on kernels that are radial symmetric. However, it can be straightforward to formulate an anisotropic kernel once the type of

kernel, for example, a Gaussian kernel, is given and the components of feature vector are assumed to be independent.

Kernel Bandwidths

Once the type of kernel is selected, there is one free parameter—the kernel bandwidth—to be chosen. The selection of kernel bandwidth is not a trivial task. The bandwidth can be selected according to the scale of the estimate. Scale selection is directly related to the resolution, for example, of semantic meaningful entities desired by the users, and is therefore application-dependent. Further, the bandwidths of a kernel may vary in different parts of the data space. One may choose the bandwidth automatically according to some pre-selected criteria, for example, minimum (Asymptotic) MISE (Mean Integrated Square Error), adaptive bandwidth, minimum intra-versus inter-cluster distances, or using data-driven methods for selecting bandwidths (Singh, 2003; Dementhon, 2001; Abramson, 1982).

Feature Vectors

A feature vector \mathbf{x} may consist of domain and range components, that is, $\mathbf{x} = [\mathbf{x}^d \ \mathbf{x}^r]$. Domain features are variables, for example, the pixel coordinates $\mathbf{x}^d = [s_x \ s_y]^T$. Range features are functions of variables, for example, the magnitudes of color image components $\mathbf{x}^r = \mathbf{f}(\mathbf{x}^d) = [R(\mathbf{x}^d) \ G(\mathbf{x}^d) \ B(\mathbf{x}^d)]^T$. Depending on the applications, we may define different feature vectors \mathbf{x} for the mean shift.

MEAN SHIFT-BASED FILTERING AND SEGMENTATION OF 2D IMAGES

Mean Shift Filtering

Instead of using mean shift to find the mode of density estimate in the previous section, mean shift can be used as a nonlinear filter that preserves sharp changes and smoothes out small variations. There is a clear association between a mean shift-based filter and a bilateral filter. Depending on the setting of the feature vector, we can see that a mean shift-based filter can be equivalent to a domain filter, a range filter or a bilateral filter.

Spatial mean shift filters: When choosing the feature vector as the spatial coordinates $\mathbf{x} = \mathbf{x}^d = [s_x \ s_y]^T$ of image pixels, and setting $g(\mathbf{x}) = g_d(\mathbf{x}^d)$ and the bandwidth as $h = h_d$ in equation (7), we obtain the equation for computing a spatial mean shift.

Range mean shift filters: Let the feature vector contain the range feature, $\mathbf{x} = \mathbf{x}^r = \mathbf{f}(\mathbf{x}^d)$, for example, intensity values of color image pixels $\mathbf{x} = \mathbf{f}(\mathbf{x}^d) = [R(\mathbf{x}^d) \ G(\mathbf{x}^d) \ B(\mathbf{x}^d)]^T$, where $\mathbf{x}^d = [s_x \ s_y]^T$, and set $g(\mathbf{x}) = g_r(\mathbf{f}(\mathbf{x}^d))$ and $h = h_r$ in equation (7), to obtain the equation for computing a range mean shift.

Joint space-range mean shift filters: Let the feature vector contain both the domain and the range variables, $\mathbf{x}=[\mathbf{x}^d \ \mathbf{f}(\mathbf{x}^d)]^T$, and set $g(\mathbf{x})=g_d(\mathbf{x}^d)g_r(\mathbf{f}(\mathbf{x}^d))$ with the corresponding h_d and h_r in equation (7), to obtain the equation for computing a joint space-range mean shift. If a Gaussian kernel is selected, then $g(\mathbf{x})$ becomes:

$$g_N(\|\mathbf{x}\|^2) = \exp\left(-\frac{\|\mathbf{x}^d\|^2}{2\sigma_d^2}\right) \exp\left(-\frac{\|\mathbf{f}(\mathbf{x}^d)\|^2}{2\sigma_r^2}\right) \quad (15)$$

where $2\sigma_d^2$ and $2\sigma_r^2$ are the domain and range bandwidths, respectively.

Examples of Nonlinear Edge-Preserving Image Smoothing

Edge-preserving image smoothing (or, enhancement) is one of the basic applications of mean shift filtering. In addition, it is also a building block for mean shift-based segmentation of 2D images and 3D video. To obtain edge-preserving image smoothing, we employ a joint space-range mean shift filter. By slightly modifying the algorithm in Table 1, we can obtain the mean shift filtered image pixels. Let the feature vector be defined as $\mathbf{x}=[\mathbf{x}^d \ \mathbf{f}(\mathbf{x}^d)]^T$, where $\mathbf{x}^d=[s_x \ s_y]^T$ and $\mathbf{x}^r=\mathbf{f}(\mathbf{x}^d)$. Let \mathbf{x} be drawn from the original 2D image and initially located at the centre of the kernel and $\{\mathbf{y}_j=[\mathbf{y}_j^d \ \mathbf{f}(\mathbf{y}_j^d)]^T, j=1,2,\dots\}$ be the successive centre locations of the kernel after mean shift iterations. The original 2D image has a corresponding set of feature vectors $\{\mathbf{x}_i, i=1,2,\dots,n\}$. The algorithm for space-range mean shift image filtering can be described by the pseudo codes in Table 2.

A joint space-range mean shift filter takes into account both the geometrical closeness and photometric similarity in an image. Such a mean shift filter can be used as a nonlinear edge-preserving smoothing filter when the range is set to be the image intensity. If the differences of pixel intensities are small, the mean shift filter acts as a lowpass filter in a local image region. However, if the intensity differences are large (e.g., around edges), then the range filter kernel is close to zero value, hence no filtering is actually applied to these pixels. This is more obvious when observing the example in Figure 1, where a space-range mean shift filter with a Gaussian kernel was applied to the 2D data. From Figure 1, one can see that the noise in the original signals was smoothed out while the sharp changes (similar to image edges) were preserved.

Figure 2 shows the results from applying a nonlinear edge-preserving smoothing filter to 2D images by using a joint spatial-intensity mean shift filter with a Gaussian kernel. One can observe from the second and third columns of Figure 2 that increasing the kernel bandwidth leads to loss of more details although the image edges are largely maintained. Changing kernel bandwidth allows for the maintenance of edges with different sharpness while smoothing the details in images.

Table 2. Pseudo codes for joint space-range mean shift filtering

Algorithm 2: Joint Space-Range Mean Shift Filtering of 2D Image.

Define the feature vector for mean shift filter as $\mathbf{x} = \begin{bmatrix} \mathbf{x}^d & \mathbf{f}(\mathbf{x}^d) \end{bmatrix}^T$, where $\mathbf{x}^d = \mathbf{s} = \begin{bmatrix} s_x & s_y \end{bmatrix}^T$;

Given: original image $\mathbf{f}(\mathbf{x}^d)$, or, corresponding feature set $S = \left\{ \begin{bmatrix} \mathbf{x}_i^d & \mathbf{f}(\mathbf{x}_i^d) \end{bmatrix}^T, i=1,2,\dots,n \right\}$

For each pixel in the original image $\mathbf{f}(\mathbf{x}^d)$, do:

Step 1: $j=1$, set $\mathbf{y}_1 = \mathbf{x}$;

Step 2: $j \leftarrow j+1$; calculate

$$\mathbf{y}_{j+1} = \frac{\sum_{l=1}^n \mathbf{x}_l g_d \left(\frac{\|\mathbf{y}_j^d - \mathbf{x}_l^d\|^2}{h_d} \right) g_r \left(\frac{\|\mathbf{f}(\mathbf{y}_j^d) - \mathbf{f}(\mathbf{x}_l^d)\|^2}{h_r} \right)}{\sum_{l=1}^n g_d \left(\frac{\|\mathbf{y}_j^d - \mathbf{x}_l^d\|^2}{h_d} \right) g_r \left(\frac{\|\mathbf{f}(\mathbf{y}_j^d) - \mathbf{f}(\mathbf{x}_l^d)\|^2}{h_r} \right)} \quad (16)$$

Step 3: calculate mean shift $m(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j$;

Step 4: repeat Steps 2 and 3, until the above mean shift converges, $\|m(\mathbf{y}_c)\| < \varepsilon$,

where $\mathbf{y}_c = \begin{bmatrix} \mathbf{y}_c^d & \mathbf{f}(\mathbf{y}_c^d) \end{bmatrix}^T$;

Step 5: Set the filtered image pixel value as $\mathbf{F}(\mathbf{x}^d) = \mathbf{f}(\mathbf{y}_c^d)$;
(i.e., only replace the range value)

End;

Output the mean shift filtered image $\mathbf{F}(\mathbf{x}^d)$.

Figure 1. Nonlinear edge-preserving smoothing by using a joint spatial-intensity mean shift filter with a Gaussian kernel: (a) The synthetic image $f(x, y)$ contains a sharp step change and random noise; (b) resulting image $F(x, y)$ obtained from mean shift filtering. The kernel parameters were set to be $\sigma_d=5$ and $\sigma_r=50$

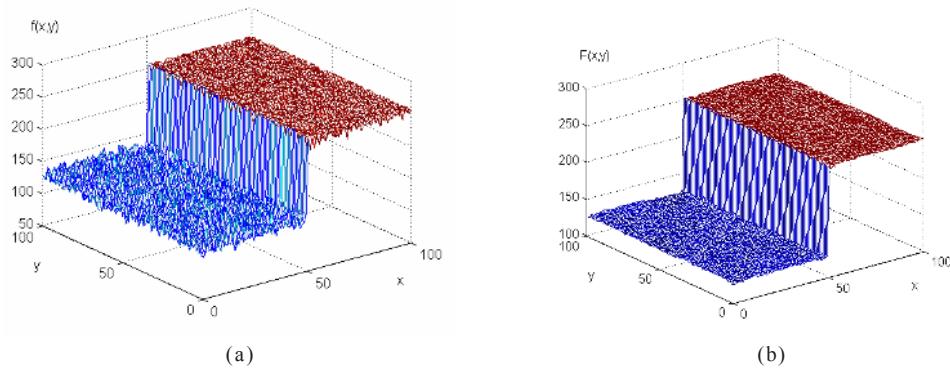


Figure 2. Nonlinear edge-preserving smoothing filtered images by using a joint spatial-intensity mean shift filter with Gaussian kernels. First column: the original images; second column: filtered images with Gaussian kernel parameters $\sigma_d=2$ and $\sigma_r=20$; 3rd column: filtered images with Gaussian kernel parameters $\sigma_d=2$, $\sigma_r=50$



Relations of Mean Shift Filters and Bilateral Filters: Similarity and Difference

There are considerable efforts for developing efficient edge-preserving nonlinear smoothing filters. Bilateral filters (Tomasi, 1998; Barash, 2002) are among one of the most frequently used methods.

Brief Review of Bilateral Filters

Similar to joint space-range mean shift filters, a bilateral filter combines a domain filter and a range filter, hence taking into account the geometrical closeness as well as the photometrical closeness property of the image.

Domain filters: The filtered image $\mathbf{F}_d(\mathbf{x})$ from a time-invariant domain filter is obtained as:

$$\mathbf{F}_d(\xi) = \frac{1}{c_d} \int \mathbf{f}(\mathbf{s}) h_d(\xi - \mathbf{s}) d\mathbf{s} \quad (17)$$

where c_d is a constant used for normalization so that the filter preserves the energy of the filtered signal and $\mathbf{f}(\mathbf{s})$ is the original image, $\mathbf{f}(\mathbf{s})=[R(\mathbf{s}), G(\mathbf{s}), B(\mathbf{s})]^\top$ for a color image, and $\mathbf{f}(\mathbf{s})=\mathbf{f}(\mathbf{s})$ for a gray-scale image. A domain filter only considers the geometrical closeness of data samples. A domain filter is the conventional type of filter that we most frequently use in digital signal processing applications.

Range filters: A range filter uses image intensity as the variable. The filtered image from a time-invariant range filter is obtained as:

$$\mathbf{F}_r(\xi) = \frac{1}{c_r} \int \mathbf{f}(\mathbf{s}) h_r(\mathbf{f}(\xi) - \mathbf{f}(\mathbf{s})) d\mathbf{s} \quad (18)$$

where c_r is a constant used for normalization. A range filter considers the range closeness of data samples, for example, photometric similarity of image pixels.

Bilateral filters: A bilateral filter combines the domain and range filters. The filtered image from a time-invariant bilateral is:

$$\mathbf{F}_b(\xi) = \frac{1}{c_b} \int \mathbf{f}(\mathbf{s}) h_d(\xi - \mathbf{s}) h_r(\mathbf{f}(\xi) - \mathbf{f}(\mathbf{s})) d\mathbf{s} \quad (19)$$

where c_b is a constant used for normalization. When applying a bilateral filter to an image, one can observe the following: If the differences of pixel intensities are small, a bilateral filter acts as a lowpass filter in a local image region. However, if the intensity differences of pixels are large (e.g., around edges), then the range filter is close to zero value, hence no filtering is actually applied to these pixels. When choosing a Gaussian kernel, then the bilateral filtered image becomes:

$$\mathbf{F}_b(\xi) = \frac{1}{c_b} \int \mathbf{f}(\mathbf{s}) \exp\left(-\frac{\|\xi - \mathbf{s}\|^2}{2\sigma_d^2}\right) \exp\left(-\frac{\|\mathbf{f}(\xi) - \mathbf{f}(\mathbf{s})\|^2}{2\sigma_r^2}\right) d\mathbf{s} \quad (20)$$

Similarity and Differences Between Joint Space-Range Mean Shift and Bilateral Filters

Comparing, for example, equation (20) with equation (7) (and notice the Gaussian kernel in equation (15)), one can see that a bilateral filter is mathematically similar to a joint space-range mean shift filter. However, a significant difference is that the kernel in the mean shift filter uses different data samples/feature sets during the mean shift iterations—the kernel moves both in spatial and in range domains. While in the bilateral filter, the filter kernel uses data samples/features from a fixed spatial area. Hence, a mean shift filter is expected to perform better than a bilateral filter.

Examples of 2D Image Segmentation

One of the main applications of mean shift filters is 2D image segmentation. It is also used as one of the building blocks in the subsequent video segmentation method. The

Table 3. Pseudo codes for mean shift-based segmentation of 2D image

Algorithm 3: Mean shift-based 2D segmentation.

Define the feature vector for mean shift filter as: $\mathbf{x} = [\mathbf{x}^d \quad \mathbf{f}(\mathbf{x}^d)]^T$

Given: original image $\mathbf{f}(\mathbf{x}^d)$, or equivalent feature set $S = \left\{ [\mathbf{x}_i^d \quad \mathbf{f}(\mathbf{x}_i^d)]^T, i=1,2,\dots,n \right\}$;

Step 1. Apply mean-shift filtering (Algorithm 2 in Table 2), and store both the feature set at the convergence points $\{\mathbf{z}_i = [\mathbf{y}_{c,i}^d \quad \mathbf{f}(\mathbf{y}_{c,i}^d)]^T, i=1,2,\dots,n\}$ and $\mathbf{F}(\mathbf{x}^d)$;

Step 2: Cluster feature vectors \mathbf{z}_i whose range difference is smaller than h_r and domain difference is smaller than h_d , and assign region labels R_j to each cluster; $j=1,2,\dots$;

Step 3: For each small region having less than M pixels, merging it to a neighboring region having the closest mode.

aim of the segmentation is to obtain an image partitioning, or regions where pixels in the same region share some common properties, for example, intensity homogeneity, texture similarity or mode similarity. Segmentation of a mean shift filtered image is based on the mode of image pixels, that is, all neighbouring pixels sharing a common mode are considered as belonging to the same region. Mean shift-based segmentation consists of joint space-range mean shift image filtering and some post-processing for segmentation. Table 3 describes the pseudo codes for the mean shift-based segmentation algorithm.

Segmentation Examples

As an example of 2D image segmentation, Figure 3 shows some results from the mean shift-based image segmentation using the algorithm in Table 3. Step 1 in Table 3 was applied four times before segmentation to obtain more stable regions. In the mean shift-based segmentation algorithm, the feature vector is defined as:

$$\mathbf{x} = [\mathbf{x}^d \quad \mathbf{f}(\mathbf{x}^d)]^T \quad (21)$$

where

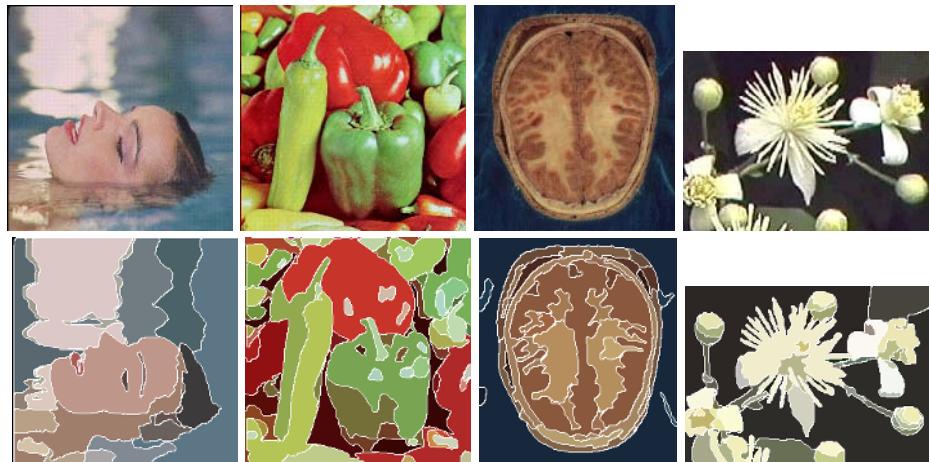
$$\mathbf{f}(\mathbf{x}^d) = [R(\mathbf{x}^d) \quad G(\mathbf{x}^d) \quad B(\mathbf{x}^d)]^T, \quad \mathbf{x}^d = [s_x \quad s_y]^T$$

and the kernel is defined as:

$$g(\mathbf{x}) = g_d(\mathbf{x}^d)g_r(\mathbf{f}(\mathbf{x}^d)) \quad (22)$$

Equation 15 gives the corresponding expression of Equation 22 when the kernels are Gaussian. Comparing the “brain” image with the remaining three images in Figure 3, a narrower kernel bandwidth was applied to the “brain” image as the intensity variations

Figure 3. Results from mean shift-based segmentation of 2D color images. First row: the original color images: swimming lady, pepper, human brain and flowers; second row: segmented images marked with boundaries. For the images “swimming lady,” “flower” and “pepper,” the bandwidths of Gaussian kernels were set to be $2\sigma_r^2 = 1024$ and $2\sigma_d^2 = 28$. In the post-processing, the threshold for merging small regions was set to be $M=50$ pixels. The Gaussian kernel was truncated to size 19×19 to obtain a finite support. For “brain” image, the kernel bandwidths were $2\sigma_r^2 = 576$ and $2\sigma_d^2 = 16$, the merging threshold was $M=100$ pixels and the kernel was truncated to size 11×11 .



between different semantic entities are small. Correspondently, the support of the kernel also becomes smaller. Furthermore, the threshold for region merging was adjusted according to the sizes of semantically meaningful regions in the image. Therefore, a large threshold was chosen for the “brain” image due to relatively large semantic regions. These four images represent a range of images with rather different properties. The segmented results in Figure 3 show that the space-range mean shift-based segmentation is effective and robust, and the segmented regions are indeed closely related to semantically meaningful entities.

VIDEO SEGMENTATION BY USING JOINT SPACE-TIME-RANGE ADAPTIVE MEAN SHIFT

In many object-based image coding applications, such as MPEG, multimedia and 3D virtual reality, foreground moving objects are encoded separately in order to obtain high image quality with low coding rate. Video segmentation is aimed at segmenting (moving) objects and background along the temporal direction. We assume that the motion areas in images are associated with the movement of some foreground/background objects/

regions (i.e., a part of an object), for example, foreground persons, background trees or roads (e.g., using a camera that is mounted on a moving car). Based on the ideas that each object/distinct part of object is associated with a pdf in certain mode(s), we can track the moving objects if we trace the motion of these particular modes that represent the object/region along the temporal direction. It is important to mention that these objects/regions are likely to be non-rigid, and experience some degree of deformation. If we take into account that the joint probability density function for a video volume is the product of two independent pdfs for spatial and temporal domains, then mode seeking by 3D mean shift can be simplified as seeking the modes for the component pdfs. We shall describe a novel joint space-time-range mean shift filter for video segmentation that is inspired by such an idea. The method is based on mode matching between consecutive image frames combined with dynamically generating new regions through the use of a joint space-time-range adaptive mean shift filtering procedure.

Inter-Frame Local Mode Matching by Mean Shift

The main purpose of inter-frame mode matching is to segment moving (or static) objects/regions between consecutive image frames, where particular interest is on the moving object-related segmentation. Video segmentation through *mode matching* is based on the idea that the mode of a pixel(s) in the current frame should be coincident with the mode of a subset of data (or region) representing the same object or region. To achieve this, a joint space-time-range mean shift filter is employed for matching the modes of pixels in two consecutive image frames.

For a given image sequence $\mathbf{f}(\mathbf{x}^d(t))$, where $\mathbf{x}^d(t) = [s_x(t) \ s_y(t)]^T$, and $\mathbf{f}(\mathbf{x}^d(t)) = [R(\mathbf{x}^d(t)) \ G(\mathbf{x}^d(t)) \ B(\mathbf{x}^d(t))]^T$, let the feature vector be $\mathbf{x}(t) = [\mathbf{x}^d(t) \ \mathbf{f}(\mathbf{x}^d(t))]^T$. Define a mean shift in the joint space-time-range domain as follows:

$$m(\mathbf{x}(t); \{\mathbf{x}_i(t-1)\}) = \frac{\sum_{i=1}^n \mathbf{x}_i(t-1) g_d \left(\left\| \frac{\mathbf{x}^d(t) - \mathbf{x}_i^d(t-1)}{h_d} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{f}(\mathbf{x}^d(t)) - \mathbf{f}(\mathbf{x}_i^d(t-1))}{h_r} \right\|^2 \right)}{\sum_{i=1}^n g_d \left(\left\| \frac{\mathbf{x}^d(t) - \mathbf{x}_i^d(t-1)}{h_d} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{f}(\mathbf{x}^d(t)) - \mathbf{f}(\mathbf{x}_i^d(t-1))}{h_r} \right\|^2 \right)} - \mathbf{x}(t) \quad (23)$$

where the kernel is defined by equations (22) and (15). It is important to notice that the feature set is taken from the previously segmented $(t-1)$ -th frame, $\{\mathbf{x}_i(t-1), i=1,2,\dots,N\}$, while the feature for the initial centre of the kernel $\mathbf{x}=\mathbf{x}(t)$ is taken from the t -th frame of the image that is being segmented. During the mean shift iterations, the feature set from the previous frame changes as the location of the kernel centre shifts. The mode matching process continues until $\mathbf{x}(t)$ converges to a mode that matches, or all feature subsets are exhausted for the $(t-1)$ -th frame.

A feature vector $\mathbf{x}(t)$ is defined to match an object/region in the previous frame if the convergence point $\mathbf{y}_c(t-1)$ has a small range distance to $\mathbf{x}(t)$, that is, $\|\mathbf{x}(t) - \mathbf{y}_c(t-1)\|^2 \leq h_r$. Otherwise, it is assumed that the pixel belongs to a new object/region that is introduced in the current frame. This can be the result of removing an occlusion or introducing a new object / region. In such a case, the segmentation procedure is switched to a so-called *intra-frame local mode estimation*.

To save the computation, instead of mode matching over all possible image areas one can limit the area such that the maximum expected object movement will not exceed it. It is straightforward to find the particular form of equation (23) if a Gaussian kernel is used. A Gaussian kernel may obtain a better segmented image after the convergence, at the expense of slower convergence speed as compared to the case of Epanechnikov kernels.

Intra-Frame Local Mode Estimation for Newly Appearing Objects / Regions

For those pixels that have no matched modes from the previous frame, the segmentation procedure is switched to intra-frame local mode estimation. In such a case, new objects / regions are estimated by exclusively using the 2D information from the unmatched pixels in the current frame, that is, by intra-frame segmentation.

Mean Shift-Based Segmentation

Those pixels that have found their matching modes in the previously segmented frame are naturally grouped into regions. If a region has a large displacement as compared with the region in the previous frame having a similar mode, then it is a moving object or a motion region. Otherwise, it is a static background / object region. This can also be used as a basis for moving object tracking and segmentation.

Those pixels that have not found correct mode matching and hence switched back to using space-range mean shift filtering in the current frame are then filtered and intra-frame segmented. However, whether these new regions are associated with static or motion objects / regions can only be estimated in the subsequent image frames. The final step in the current frame segmentation consists of region mode re-evaluation using data from the current frame. This allows tracking illumination changes and other variations of regional features. It is also beneficial in terms of segmentation stability, along with a temporal mean shift filter described in the next paragraph.

Algorithm

The video segmentation algorithm can be described as follows. For a given image sequence, we first apply a mean shift-based space-range filter to the first frame of image sequence for 2D object/region segmentation. Starting from the second frame, a joint space-time-range mean shift filter is applied as follows: For each pixel $\mathbf{x}^d(t) = [s_x(t) \ s_y(t)]^T$ in the current frame t , let the corresponding feature vector $\mathbf{x}(t) = [\mathbf{x}^d(t) \ \mathbf{f}(\mathbf{x}^d(t))]^T$ be in the kernel centre, and let the set of feature vectors $\{\mathbf{x}_i(t-1) = [\mathbf{x}_i^d(t-1) \ \mathbf{f}(\mathbf{x}_i^d(t-1))]^T\}$, $i = 1, 2, \dots, n\}$ be from the previous frame $(t-1)$. The joint space-time-range mean shift filter is then applied. Since a new centre of mean shift is generated from each iteration, a new set of data/features from frame $(t-1)$ is effectively used in the next iteration. Therefore, the iterative algorithm seeks the *mode matching* between a pixel in the current frame and a variable set of data in the previous frame. There are two possible situations for the mode matching. One is that the pixel in the current frame finds a similar mode to which it converges. In such a case, the resulting image pixel is replaced by the filtered pixel at the convergence point $\mathbf{F}(\mathbf{x}^d(t)) = \mathbf{f}(\mathbf{x}_c^d(t-1))$. Another

Table 4. Pseudo codes for joint space-time-range adaptive mean shift-based video segmentation

<p><i>Step 1:</i> Set the frame number $t=1$ (the 1st image frame): <i>define</i> $\mathbf{x} = [\mathbf{x}^d(t) \quad \mathbf{f}(\mathbf{x}^d(t))]^T$, apply the 2D space-range mean-shift segmentation algorithm in Table 3 to image $\mathbf{f}(\mathbf{x}^d(t=1))$, and yielding a filtered and segmented image $\tilde{\mathbf{F}}(\mathbf{x}^d(t=1))$.</p> <p><i>Step 2:</i> Set the frame number $t \leftarrow t + 1$,</p> <p><i>Step 3:</i> For each pixel $\mathbf{x}^d(t) = [s_x(t) \quad s_y(t)]^T$ in the t-th frame image, do:</p> <p><i>Step 4:</i> Pre-processing for regularizing intensity values along the temporal direction: <i>Set</i> $\mathbf{x} = [t \quad R(\mathbf{x}^d(t)) \quad G(\mathbf{x}^d(t)) \quad B(\mathbf{x}^d(t))]^T$ for a given fixed \mathbf{x}^d, apply a 1D range mean shift filter along the temporal direction, with a kernel bandwidth h_t.</p> <p><i>Step 5:</i> Joint space-time-range adaptive mean shift filtering and segmentation:</p> <p>5.1. Set the initial mean shift kernel centre at $\mathbf{x} = \mathbf{x}(t) = [\mathbf{x}^d(t) \quad \mathbf{f}(\mathbf{x}^d(t))]^T$ drawn from a pixel in the current frame t, and set $j=1$, $\mathbf{y}_1 = \mathbf{x}$;</p> <p>5.2. Assign the feature set $\left\{ \mathbf{x}_i = \mathbf{x}_i(t-1) = [\mathbf{x}_i^d(t-1) \quad \mathbf{f}(\mathbf{x}_i^d(t-1))]^T, i=1,2,\dots,n \right\}$ centered at $\mathbf{y}_1 = \mathbf{x}$ and within the bandwidth h_d from the frame $(t-1)$ to the filter;</p> <p>5.3. Mean shift iteration: $j \leftarrow j + 1$, <i>compute</i> \mathbf{y}_{j+1} by using Eq.(16), <i>shift</i> the data window centre to \mathbf{y}_{j+1}, <i>compute</i> the mean shift $m(\mathbf{x}) = \mathbf{y}_{j+1} - \mathbf{y}_j$, <i>and assign</i> a new set $\left\{ \mathbf{x}_i = \mathbf{x}_i(t-1), i=1,2,\dots,n \right\}$;</p> <p>5.4. Repeat iterations in Step 5.3 until convergence $\ m(\mathbf{x})\ = \ \mathbf{y}_{j+1} - \mathbf{y}_j\ < \epsilon$;</p> <p>5.5. Case 1. Inter-frame mode matching: <i>If</i> the range distance $\ \mathbf{x}^r(t) - \mathbf{y}_c\ ^2 \leq h_r$, where $\mathbf{y}_c = [\mathbf{x}_c^d(t-1) \quad \mathbf{f}(\mathbf{x}_c^d(t-1))]^T$, <i>then</i> a similar mode is found. <i>Assign</i> filtered pixel value $\mathbf{F}(\mathbf{x}^d(t)) \leftarrow \mathbf{f}(\mathbf{x}_c^d(t-1))$;</p> <p>5.6. Case 2. Intra-frame mode estimation: <i>If</i> no similar mode in the $(t-1)$ frame that matches $\mathbf{x}(t)$, <i>then assign</i> $\left\{ \mathbf{x}_i = \mathbf{x}_i(t) = [\mathbf{x}_i^d(t) \quad \mathbf{f}(\mathbf{x}_i^d(t))]^T, i=1,2,\dots,n \right\}$. <i>Apply</i> a space-range mean shift filter to $\mathbf{x}(t)$ until it converges. <i>Assign</i> $\mathbf{F}(\mathbf{x}^d(t)) \leftarrow \mathbf{f}(\mathbf{x}_c^d(t))$, where $\mathbf{y}_c = [\mathbf{x}_c^d(t) \quad \mathbf{f}(\mathbf{x}_c^d(t))]^T$.</p> <p>Step 6: Repeat Step 3 to Step 5 until all pixels in the t-th frame image are filtered;</p> <p>Step 7: Segment mean shift filtered image $\mathbf{F}(\mathbf{x}^d(t))$, and assign the results to $\tilde{\mathbf{F}}(\mathbf{x}^d(t))$;</p> <p>Step 8: Link the similar regions between frames $(t-1)$ and t.</p> <p>Step 9: Repeat Step 2 to Step 8 until all frames are processed.</p> <p>Step 10: Output segmented image sequence $\tilde{\mathbf{F}}(\mathbf{x}^d(t))$, $t=1,2,\dots$</p>
--

situation is that the pixel cannot find a similar mode from the previous frame. In such a case, a space-range mean shift filter is then applied to current image frame t to obtain the filtered pixels. This process repeats over all pixels in the current frame. After that, the resulting mean shift filtered image for the current frame $\mathbf{F}(\mathbf{x}^d(t))$ is segmented.

To enhance the data regularity along the temporal direction, a range mean shift filter is applied to $\mathbf{x}(t)=\mathbf{f}(\mathbf{x}^d(t))$ along the temporal direction before applying the joint space-time-range mean shift filter for video segmentation. This brings additional regularity to the segmentation that could be unstable, for example, small variations of the next frame pixel values may lead to different partitions. The temporal directional mean shift filter alleviates such a potential problem.

To decide whether a region is associated with a static or a moving object, one may examine the deviation of the mode location after the convergence, that is., using the so-called *mode of motion vector* $(\mathbf{x}_c^d - \mathbf{x}_0^d)$, where \mathbf{x}_0^d is the centre of the region in frame t , and \mathbf{x}_c^d is the centre of the region in frame $(t-1)$ at which $\mathbf{f}(\mathbf{x}_0^d)$ converges. If the magnitude of the motion vector is large, then it is recognized as a motion region. Table 4 summarizes the pseudo codes of video segmentation through iteratively using space-time-range mean shift filtering.

Examples and Results

The proposed joint space-time-range adaptive mean shift-based video segmentation algorithm in Table 4 has been applied to several image sequences. Gaussian kernels were applied for the mean shift. Figure 4 shows the results from video segmentation “Tennis” where RGB color images were used. The spatial bandwidth of the Gaussian kernel was set as $2\sigma_d^2 = 28$, and the range bandwidth was set as $2\sigma_r^2 = 1024$. For 1D temporal directional range mean shift filter, the kernel bandwidth was set to be $2\sigma_t^2 = 36$. To save the computation, the kernel in the spatial domain was truncated to size 19×19 , and the kernel in the temporal direction was truncated to size 5. For post-processing of merging small regions, the threshold was set as $M=25$ pixels. One can observe that the segmentation results are rather satisfactory. The moving objects and object parts (e.g., the ball and the arm) have been correctly matched.

Figure 5 shows the results from segmenting the “flower” sequence using RGB color images. The flower images contain many small regions (e.g., flowers with different colors, and rich in textures).

In the testing, the spatial and range bandwidths of Gaussian kernels were set as $2\sigma_d^2 = 28$ and $2\sigma_r^2 = 1024$, respectively. For the 1D temporal directional range mean shift filter, the kernel bandwidth was set to be $2\sigma_t^2 = 36$. To save the computation, the kernel in the spatial domain was truncated to size 11×11 , and the kernel in the temporal direction was truncated to size 5. Since there are many small regions containing flowers with different colors, a small region merging threshold $M=5$ pixels was applied in this case. The results in Figure 5 have shown good segmentation results in matching moving tree and flowers, and re-generate new regions (e.g., an uncovered window in the house).

Figure 6 shows the results from segmenting the “Mom and Daughter” sequence using RGB color images. Consider that the images contain very similar colors for different semantic entities (skin, hair, shirt, and even wall), and also eyes and lips contain a relatively small number of pixels; the threshold for small regions merging was chosen to

Figure 4. Results of video segmentation “Tennis” by using joint space-time-range mean shift filters and color images $\mathbf{f} = (R, G, B)$. From left to right, first row: frame number 35, 36, 37; second row: frame number: 38, 39, 40.

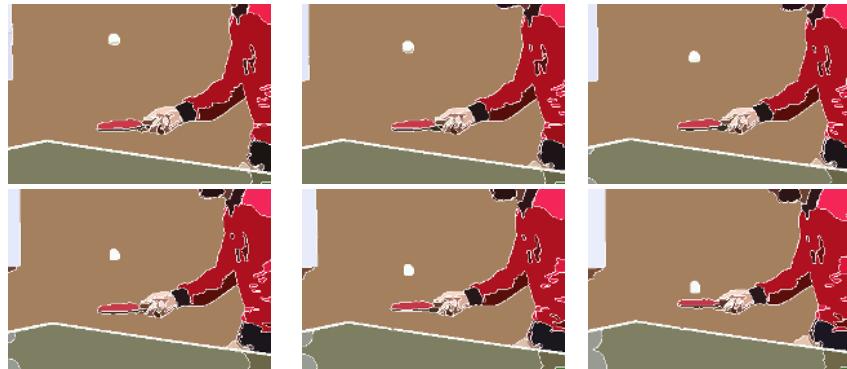


Figure 5. Results of video segmentation “flower” by using joint space-time-range mean shift filters and color image $\mathbf{f} = (R, G, B)$. From left to right, first row: frame number 32, 33, 34; second row: frame number: 35, 36, 37.



be $M=10$ pixels. Gaussian kernels were applied with the spatial and range bandwidths $2\sigma_d^2 = 28$ and $2\sigma_r^2 = 1024$, respectively. A relatively small range bandwidth was used since the differences between the range values in different semantically meaningful regions are small. For 1D temporal directional range mean shift filter, the kernel bandwidth was set to be $2\sigma_t^2 = 36$. To save the computation, the kernel in the spatial domain was truncated to size 11×11 , and the kernel in the temporal direction was truncated to size 5. The

Figure 6. Results of video segmentation “Mom and Daughter” by using joint space-time-range mean shift filters and color images $\mathbf{f}=(R,G,B)$. From left to right, first row: an example of the original image (frame 6), results for frame number 6,7; second row: results for frame number: 8, 9,10.

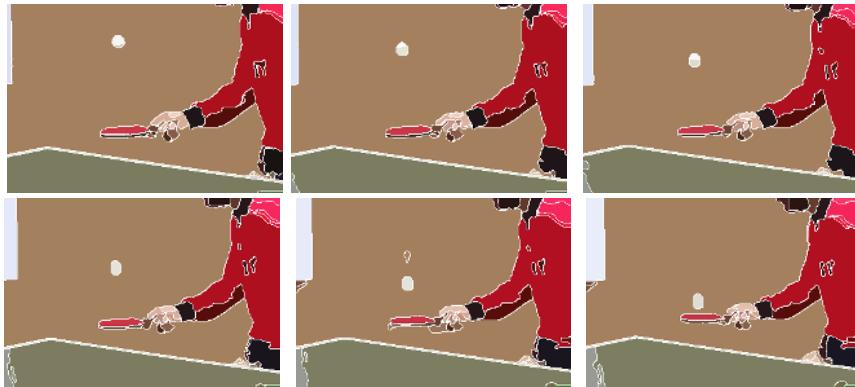


segmentation results in Figure 6 are reasonably good, considering that many different semantic entities are close in the range domain (see one original image shown in the upper left corner of Figure 6), and the dynamic range of spatial bandwidths for different objects (or object parts) is large.

Instead of using RGB color images, one may use some perceptually more relevant color spaces. Since image luminance and chroma are more independent features, one can often exploit the color luminance differences more efficiently for image segmentation. Figure 7 shows the video segmentation of the “Tennis” image sequence, where perceptually relevant $L^*u^*v^*$ color images were used. In the tests the range bandwidth was $2\sigma_r^2 = 512$, the spatial bandwidth was $2\sigma_d^2 = 16$, the temporal bandwidth was $2\sigma_t^2 = 36$, and the Gaussian kernel was truncated to finite size of 19×19 pixels in the spatial domain. The threshold for merging small regions was set to $M=25$ pixels. Comparing the segmentation results of Figure 7 with Figure 4, one can observe some improvement in using $L^*u^*v^*$ space segmentation. For example, the “pingpong” ball is better matched, although there is an extra region in frame 39 which is probably caused by the shadow of ball. One can also observe that there is less region-split in the body of the person. However, segmentation of hands seems to be worse than that in Figure 4.

We can also observe that the above examples contain image sequences with very different characteristics: from fast motion of “pingpong” ball and color rich images (“Tennis” sequence), to images that are rich in textures and small regions (“Flower” sequence) and those whose semantic entities have close colors and very different region sizes (“Mom and Daughter” sequence). From these examples, one can observe that the joint space-time-range adaptive mean shift-based segmentation algorithm in Table 4 is effective and robust for segmenting video with a range of properties.

Figure 7. Results of video segmentation of “Tennis” by using joint space-time-range mean shift filters and perceptually relevant color image $\mathbf{f}=(L^*, u^*, v^*)$. First row: image frame number 35, 36, 37; second row: 38, 39, 40.



DISCUSSION

Mean shift seeks local modes of the kernel density estimate and is related to the gradient of the kernel density estimate of image pdf at \mathbf{x} . Once a kernel density function is selected, it is essential to select some proper kernel bandwidths. A smaller bandwidth may result in more local modes (or peaks) than that from a large bandwidth. The performance of mean shift segmentation will be influenced by the choice of kernel bandwidths. One usually needs to adjust the bandwidths empirically so that the best performance can be achieved.

High dimensional features often concentrate in a small region of the space, and with different variances in different dimensions. Therefore, a transformation is often applied before the mean shift. Further, if the dimension of feature space is high (e.g., higher than 6), some reduction in the dimension of feature space is required both from the computational viewpoint and in order to find the proper bandwith settings.

Since the mean shift segmentation method only utilizes image/video features extracted from the pixel-level, there will be a gap between the segmented regions and the high-level image semantics, or human perception. However, this is expected for almost all segmentation methods that are based on low-level image information. The main advantages of mean shift segmentation include: producing edge-preserving regions and robust performance, requiring little *a priori* information of the image (e.g., free from pre-specifying the expected number of segments as being an unsupervised segmentation method) and producing final results largely insensitive to the initial process. Further, it is also computationally inexpensive as compared with other stochastic segmentation methods. The main disadvantages are the requirement to specify the shape and the bandwidths of the kernel in advance, where extra false modes (or peaks) may appear as

the consequence of bandwidth settings that are too small. Overall, the mean shift-based segmentation is shown to be a very robust method among those pixel-based image/video segmentation methods. To further enhance the performance of segmentation, one may add some extra semantic information to help the inference, e.g., by combining with either the high-level processing or a semi-automatic process from a user, as addressed in the beginning of this chapter.

CONCLUSIONS

Mean shift is shown to be a powerful tool for image and video segmentation. The basis for mean shift-based image segmentation is to seek geometrically close pixels with similar modes in the kernel-based probability density estimate.

We have discussed the selection of kernel, of which a probability density function estimate is formed. Two frequently used kernels, that is, Gaussian kernels and Epanechnikov kernels, with L_2 norm distance and radial symmetric bandwidths are discussed. When a radial symmetric kernel is applied, we can obtain an isotropic mean shift. For anisotropic mean shift, one needs to set different bandwidths to the multidimensional kernel. For a given kernel, that is, a Gaussian kernel, it is often straightforward to set non-symmetric kernel bandwidths. To limit the scope of this chapter, only radial symmetric kernels are discussed.

We have also discussed various definitions of kernel variables in mean shift. Depending on the definition of kernel variables, a mean shift filter can be related to a domain filter, a range filter or a bilateral filter. Furthermore, a mean shift filter has an extra strength due to its flexibility. An important difference of mean shift filtering is that the actual data set used by the kernel (related to probability density estimate) of mean shift changes during the iteration, as compared to using fixed data for a conventional bilateral filter kernel (or impulse response).

Basic application examples of mean shift-based filtering for edge-preserving image smoothing and 2D image segmentation are included. In addition, the processing algorithms are also among the fundamental building blocks in mean shift-based video segmentation.

For mean shift segmentation of video volume, we have considered the joint pdf of a video volume as the product of two independent spatial and temporal component pdfs. This can simplify the mode seeking process for segmentation of the video volume. We presented a novel joint adaptive space-time-range mean shift-based segmentation method. Two major processes, that is, inter-frame mode matching and intra-frame new mode estimation, are included. Inter-frame mode matching is used to find similar modes between the pixel in the current frame and the pixels in the previous frame. This can be used to track the corresponding motion/static regions through image frames. Intra-frame mode estimation is mainly designed to handle the new regions that are introduced in the current frame, that is, new foreground objects or uncovering previously occluded background. Experimental results have demonstrated the effectiveness of the method.

REFERENCES

- Abramson, I., (1982). On bandwidth variation in kernel estimates—a square root law. *The Annals of Statistics*, 10(4), 1217-1223.
- Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 16(6), 641-647.
- Allmen, M., & Dyer, C. R. (1993). Computing spatiotemporal relations for dynamic perceptual organization. *CVGIP: Image Understanding*, 58, 338-351.
- Barash, D. (2002). A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(6), 844-847.
- Collomosse, J. P., Rowntree, D., & Hall, P. M. (2005). *Video paintbox: The fine art of video painting, computers & graphics*. Special editon on digital arts. Elsevier.
- Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Recognition & Machine Intelligence*, 24(5), 603-619.
- Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE Conference of Computer Vision & Pattern Recognition* (Vol. 2, pp. 142-149).
- Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 17(8), 790-799.
- Cheng, H.-D., Jiang X.-H., Sun, Y., & Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12), 2259-2281.
- Dave, R. N., & Krishnapuram, R. (1997). Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, 5(2), 270-293.
- DeMenthon, D., & Megret, R. (2001). The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of the IEEE 8th Int. Conf. on Computer Vision*, Canada (pp. 438-445).
- DeMenthon, D., & Megret, R. (2002, July 1-2). Spatial-temporal segmentation of video by hierarchical mean shift analysis. In *Proceedings of the Statistical Methods in Video Processing Workshop*, (SMVP 02), Copenhagen, Denmark (pp. 800-810).
- Fowlkes, C., Belongie, S., & Malik, J., (2001). Efficient spatiotemporal grouping using the Neystrom method. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (Vol. 1, pp. 231-238).
- Fu, K. S., & Mui, J. K. (1981). A survey on image segmentation. *Pattern Recognition*, 13(1), 3-16.
- Fukunaga K., & Hostetler, L., (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21, 32-40.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6(6), 721-741.
- Greenspan, H., Goldberger, J., & Mayer, A. (2004). Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions Pattern Analysis & Machine Intelligence*, 26(3), 384-396.

- Huber, P. (1981). *Robust statistics*. New York: Wiley.
- Jain, A. K., & Dubes, R. C. (1998). *Algorithms for clustering data*. NJ: Prentice Hall.
- Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4), 321-331.
- Khan, S., & Shash, M. (2001). Object based segmentation of video using color, motion and spatial information. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition* (Vol. 2, pp. 746-751).
- Klinker, G. J., Shafer, S. A., & Kanade, T. (1988). Image segmentation and reflection analysis through color. In *Proceedings of the IUW'88*, Cambridge, MA (Vol. 2, pp. 838-853).
- Koprinska, I., & Carrato, S. (2001). Temporal video segmentation: A survey, signal processing. *Image Communication*, 16, 477-500.
- Li, L., Huang, W., Gu, I. Y. H., Leman, K., & Tian, Q. (2003, October 5-8). Principal color representation for tracking persons. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Washington, DC (pp. 1007-1012).
- Li, L., Huang, W., Gu, I. Y. H., & Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11), 1459-1472.
- Lundberg, M., Svensson, L., & Gu, I. Y. H. (2001). Infrared detection of buried land mines based on texture modeling. In *Detection & Remediation Technologies for Mines and Minelike Targets VI*, SPIE 4394 (Vol. 1, pp. 199-206).
- Megret, R., & Jolion, T. (2002). Representation of dynamic video content by tracking of grey level blobs. In *Proceedings of the RFIA Conference (13ème Congrès de Reconnaissance des Formes et Intelligence Artificielle)*, Angers, France.
- Meier, T., & Ngan, K. N. (1998). Automatic segmentation of moving objects for video object plane generation. *IEEE Transactions on Circuits & Systems for Video Technology*, 8, 525-538.
- Moscheni, F., Bhattacharjee, S., & Kunt, M. (1998). Spatiotemporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(9), 897-915.
- Pan, J., Li, S., & Zhang, Y. (2000, May 28-31). Automatic extraction of moving objects using multiple features and multiple frames. In *Proceedings of IEEE Symposium Circuits and Systems*, Geneva, Switzerland (Vol. 1, pp. 36-39).
- Pauwels, E.J., & Frederix, G. (1999). Finding salient regions in images-non-parametric clustering for image segmentation and grouping. *Computer Vision and Image Understanding*, 75, 73-85.
- Pollak, I., Willsky, A.S., & Krim, H. (2000). Image segmentation and edge enhancement with stabilized inverse diffusion equations. *IEEE Transactions on Image Processing*, 9(2), 256-266.
- Singh, M., & Ahuja, N. (2003). Regression based bandwidth selection for segmentation using parzen windows. In *Proceedings of IEEE International Conference on Computer Vision* (Vol. 1, pp. 2-9).
- Stauffer, C., & Grimson, W. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 22(8), 747-757.
- Sun, S., Haynor, D.R., & Kim, Y. (2003). Semiautomatic video object segmentation using Vsnakes. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1), 75-82.

- Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision*, Bombay, India (pp. 836-846).
- Torr, P. H. S., & Zisserman, A. (1998). Concerning Bayesian motion segmentation. In *Proceedings of European Conference of Computer Vision, I* (pp. 511-527).
- Van de Weijer, J., & Van den Boomgaad, R. (2001). Local mode filtering. In *Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition (CVPR 01)* (Vol. 2, pp. 428-433).
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(6), 583-591.
- Wang, J., Thiesson, B., Xu, Y., & Cohen, M. (2004, May 11-14). Image and video segmentation by anisotropic kernel mean shift. In *Proceedings of the European Conference of Computer Vision (ECCV'04)*, Prague, Czech Republic, LNCS 3022 (pp. 238-249).
- Weickert, J. (1998). *Anisotropic diffusion in image processing*. Stuttgart: Teubner-Verlag.
- Wren, C., Azarbaygiani, A., Darrell, T., & Pentland, A. (1997). Pfinder: Real-time tracking of human body. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 19(7), 780-785.
- Xu, C., & Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3), 359-369.
- Zhang, Y.J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335-1346.
- Zhang, Y.J., Gao, Y.Y., & Luo, Y. (2004). Object-based techniques for image retrieval. In S. Deb (Ed.). *Multimedia systems & content-based image retrieval* (pp. 156-181). Hershey, PA: Idea Group Inc.
- Zhu, S.C., Wu, Y.N., & Mumford, D. (1998). Filters, random fields and maximum entropy: Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2), 107-126.
- Zhu, S.C., & Yuille, A. (1996). Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 18(9), 884-900.

Chapter VII

Fast Automatic Video Object Segmentation for Content-Based Applications

Ee Ping Ong, Institute for Infocomm Research, Singapore

Weisi Lin, Institute for Infocomm Research, Singapore

Bee June Tye, Dell Global BV, Singapore

Minoru Etoh, NTT DoCoMo, Japan

ABSTRACT

An algorithm has been devised for fast, fully automatic and reliable object segmentation from live video for scenarios with static camera. The contributions in this chapter include methods for: (a) adaptive determination of the threshold for change detection; (b) robust stationary background reference frame generation, which when used in change detection can reduce segmentation fault rate and solve the problems of disoccluded objects appearing as part of segmented moving objects; (c) adaptive reference frame selection to improve segmentation results; and (d) spatial refinement of modified change detection mask by incorporating information from edges, gradients and motion to improve accuracy of segmentation contours. The algorithm is capable of segmenting multiple objects at a speed of 12 QCIF frames per second with a Pentium-4 2.8GHz personal computer in C coding without resorting to any code optimization. The result shows advantages over related work in terms of both fault rate and processing speed.

INTRODUCTION

Video object segmentation has attracted vast research interest and substantial effort for the past decade because it is the prerequisite for object-based compression and coding (e.g., MPEG-4 codecs), visual content retrieval (e.g., MPEG-7 related schemes), object recognition and many other applications. It is a non-standardized but indispensable component for an MPEG4/7 scheme to be successful in a complete solution. In fact, in order to utilize MPEG-4 object-based video coding, video object segmentation must first be carried out to extract the required video object masks.

Temporal and spatial information and their appropriate combination have been extensively exploited (Aach & Kaup, 1993; Bors & Pitas, 1998; Castagno, Ebrahimi, & Kunt, 1998; Chen, Chen, & Liao, 2000; Chen & Swain, 1999; Chien, Ma, & Chen, 2002; Cucchiara et al., 1999; Kim et al., 1999; Kim et al., 2001; Koller et al., 1994; Li et al., 2001; Li et al., 2002; Li et al., 2004; Liu et al., 1998; Liu, Chang, & Chang, 1998; Mech & Wollborn, 1997; Meier & Ngan, 1999; Mester & Aach, 1997; Neri et al., 1998; Odobez & Bouthemy, 1998; Ong & Spann, 1999; Shao, Lin, & Ko, 1998; Shao, Lin, & Ko, 1998b; Toklu, Tekalp, & Erdem, 2000) for segmentation of objects from video sequences. However, a general solution is still elusive in practice due to its well-acknowledged inherent difficulties: (1) the difficulty in estimating objects' motion and other distinguishing features accurately in order to facilitate the separation of foreground and background moving objects; (2) the inevitable presence of noise/clutter that affects accuracy in segmentation; (3) the effects of occlusions and disocclusions; (4) the lack of generosity of an algorithm, especially in the situations in which no human intervention or fine-tuning of the segmentation parameters is possible; and (5) the high computational complexity involved which makes it difficult to design an algorithm that is robust enough to run in real-time for real-life applications.

Automatic extraction of semantically meaningful objects in video is extremely useful in many practical applications but faces problems like *ad hoc* approaches, limited domain of application, over-complex algorithms and need of excessive parameter/threshold setting and fine-tuning. With current levels of algorithm development, only supervised segmentation approaches, for example, Castagno, Ebrahimi, and Kunt (1998) and Toklu, Tekalp, Erdem (2000) are capable of detecting semantic objects more accurately from video in general. Supervised approaches find applications such as studio editing and content retrieval.

Nevertheless, automatic segmentation is essential and possible for some specific but important scenarios like video-conferencing, on-line/mobile security systems and traffic monitoring. For visual signal storage and transmission, object-based coding improves the compression and coding efficiency. In very low bit-rate applications, object segmentation enables transmitting objects of interest in higher quality and objects of less importance in lower quality (e.g., allocating smaller number of bits for them or only refreshing them occasionally, as used in sprite-based video coding), while a conventional frame-based video coding scheme would have to suffer an across-the-board reduction in quality to fit into the bandwidth available.

The requirements for a practical system to be useful in the above-mentioned scenarios are:

1. Fully automatic operations;
2. Efficient processing; and
3. Robustness with noisy data.

Automatic manipulation is necessary because user intervention is not feasible for online applications. Appropriate domain knowledge (e.g., head and shoulder scenes are mostly expected as foreground in video-conferencing) can be utilized in the process, apart from spatial and temporal information. Real-time realization is usually needed for such applications and therefore low computational complexity is the key. Fast processing itself also facilitates the success of segmentation since frame difference due to the motion becomes too big for good segmentation if a segmentation process fails to catch up with the speed of live video camera input (unlike the case with pre-stored video sequences). Moreover, the segmentation module in an integrated system may need to share the available computing power with the video compression and coding module, for example, Lin et al. (1999), which is also computationally expensive. In addition, algorithms developed should be reasonably resilient to noise for reliable segmentation with live video. Possibilities can be exploited to incorporate appropriate stationary regions (e.g., a human trunk that does not move all the time) with moving regions (like head and hands in the example) to form a semantic object.

Liu et al. (1998) describes an object boundary extraction algorithm based on optical flow that is real-time implemented on a HyperSparc workstation. For more restricted applications, real-time vehicle detection systems have been developed with dedicated hardware design (Cucchiara et al., 1999; Koller et al., 1994) and a PC-based platform (Siyal & Fathy, 1995). Such systems are based upon inter-frame difference analysis. Methods using optical flow calculation and other motion parameter estimators are generally too computationally expensive for economical real-time PC/DSP implementation.

There are various techniques for video object segmentation, but the faster video object segmentation techniques (Aach & Kaup, 1993; Cheung & Ramath, 2004; Kim et al., 1999; Liu, Chang, & Chang, 1998; Mester & Aach, 1997; Neri et al., 1998; Shao, Lin, & Ko, 1998b) are based on change detection approach (with and without pre-processing to cater for global motion) followed by further post-processing. However, the different video object segmentation methods differ in how the change detection and the post-processing are being performed. These have a significant influence on the actual speed, accuracy and reliability of the final segmentation results.

In this chapter, an automatic and robust segmentation algorithm will be proposed for typical static-camera scene applications with prospective real-time software/firmware implementation in personal computer (PC) or affordable DSP/embedded systems. The algorithm adopts the spatio-temporal paradigm. The reasoning for the algorithm and the related work are presented. The module with robust background generation, single-pixel change detection and adaptive threshold determination is introduced to obtain the change detection mask (CDM) that is the basis for the preliminary object mask (POM). Spatial refinement for the POM using edges, moving edgels and gradients' information is then described to yield segmentation of foreground objects as the final object mask

(FOM). Experimental results for the proposed algorithm and the comparison with related work in terms of accuracy and computational costs will be shown.

Related Work and Reasoning for the Proposed Algorithm

Spatio-temporal segmentation is more likely to obtain good segmentation for video when compared to methods using only motion information because spatial features usually provide more information on precise object boundaries. In this proposed work, a spatio-temporal approach using change detection (Aach & Kaup, 1993; Cheung & Kamath, 2004; Kim et al., 1999; Liu, Chang, & Chang, 1998; Mester & Aach, 1997; Neri et al., 1998; Shao, Lin, & Ko, 1998b) instead of motion model/parameter estimation (Bors & Pitas, 1998; Li et al., 2001; Liu et al., 1998; Meier & Ngan, 1999; Odobezi & Bouthemy, 1998; Shao, Lin, & Ko, 1998) is proposed since the scheme aims at real-time processing, and different motion in a foreground object does not need to be distinguished in the targeted applications (that is, a scene is separated into only two classes of objects: foreground and background). Temporal information is extracted via change detection as straightforward hints of foreground objects and then spatial features are utilized for more accurate extraction of object contours. On the balance of incorporating multiple features for good segmentation (Castagno, Ebrahimi, & Kunt, 1998) and fulfilling the real-time requirement, the minimum set of the most effective temporal and spatial features (change due to motion, edge and gradients) is used for this development.

Change detection is a fast way to extract moving foreground in video sequences with stationary cameras. However, the change detection method has the following drawbacks:

1. the correct threshold used in detecting inter-frame difference is not easy to obtain;
2. uncovered areas affect the accuracy of the CDM;
3. there are usually many *holes* inside the resultant CDM due to the less textured areas within a moving object; and
4. object contours are not accurate.

The thresholds used in change detection are often set empirically (Kim et al., 1999; Mester & Aach, 1997; Shao, Lin, & Ko, 1998b) or determined off-line using prior knowledge of noise variance of the camera (Aach & Kaup, 1993). These approaches result in poor portability of the algorithm and can also be regarded as a kind of user intervention (Castagno, Ebrahimi, & Kunt 1998). In this work, a method of adaptively determining the threshold is proposed for fully automatic change detection (based upon the similar idea in Neri et al. (1998)) that is a much more efficient criterion for online real-time manipulations. With the proposed method, proper domain knowledge is utilized to choose the *most probably stationary pixels (MPSPs)* for the said estimation, with at least 50% for the outlier tolerance in the candidate set of pixels.

Thresholding here is performed at each pixel difference instead of a neighborhood of pixel differences (in contrast to the change detection processes in Kim et al., 1999; Liu, Chang, & Chang, 1998; Mester & Aach, 1997; Neri et al., 1998; Shao, Lin, & Ko, 1998b) in order to eliminate the *blurring* effect and add computational advantage. The CDM is eroded to alleviate over-segmentation caused by uncovered areas, and then modified

with appropriate morphological operators (Vincent, 1993) to reduce the noise and holes inside the mask before forming a POM to be passed for spatial refinement. The reference frame for change detection can be the immediate previous frame or the stationary background image generated in the initialization phase. The latter approach is able to solve the problems of dis-occluded regions appearing as part of the segmented moving objects.

For the case in change detection where the reference frame is a stationary background image that has been generated in an initialization phase using a simple averaging, it is of paramount importance that there is no motion in the images being used to generate the stationary background. However, in practice, it is very difficult or almost impossible to ensure that there is no motion during the initialization phase in real-life applications (in many situations where you do not have full control over the environments—for example, in traffic monitoring and video surveillance applications). As such, we hereby proposed a method for generating the stationary background image to solve the above-mentioned problem. The proposed method has a breakdown point of 50% (which means that it is able to reject up to 50% of outliers in the data set) and is also more efficient in the presence of Gaussian noise (Rousseeuw & Leroy, 1987) compared to the use of a simple median filter (Cheung & Kamath, 2004; Piccardi, 2004). Cheung and Kamath (2004) studied the methods for generating a background model, such as frame differencing, median filter, linear predictive filter, non-parametric model, kalman filter and Mixture of Gaussians model. It has been concluded via experiments that median filtering offers a simple alternative with competitive performance. Piccardi (2004) reviewed the various background subtraction methods, such as average, median, running average, Mixture of Gaussians, kernel density estimators, mean-shift, sequential kernel density approximation and Eigen backgrounds. The conclusion drawn is that it is difficult to assess which method has the best accuracy as there is no unbiased comparison with a significant benchmark. However, simple methods such as standard average, running average and median can provide fast speed and also acceptable accuracy in specific applications. In the proposed method, the median-filter based shortest half approach is more efficient in the presence of Gaussian noise, especially when a small number of samples are being used, as compared to the simple median method.

The proposed algorithm is on-the-fly and switches the reference frame between the generated background image and the immediately previous frame, to allow maximum segmentation quality according to scenes being viewed, and also to cater for significant changes in lighting and even accidental movement of the otherwise stationary camera etc. When the generated background image is used as the reference frame for change detection, the proposed algorithm is able to solve the problems of dis-occluded regions appearing as part of the segmented moving object when large object motion occurs in the scene and thus obtain much more accurate segmentation results.

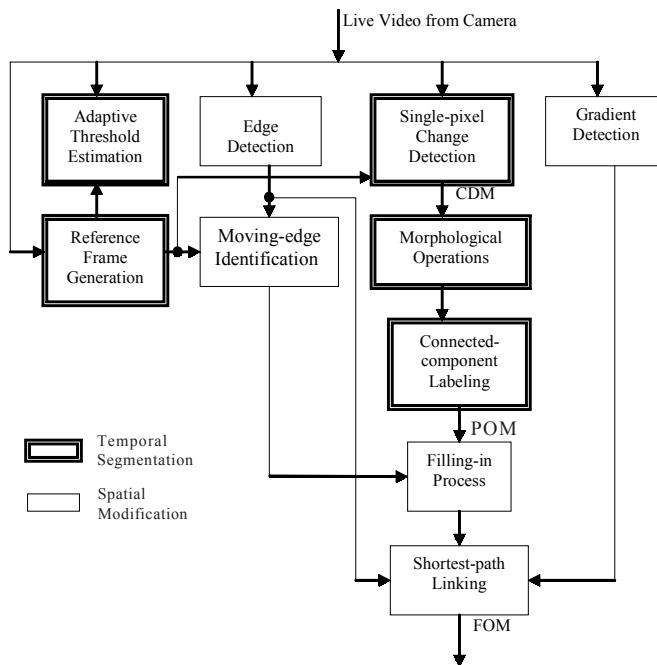
At the spatial refinement stage, the POM is consolidated by a filling-in process (similar to Meier & Ngan, 1999) but with consideration of moving edgels. Each object contour is then adjusted with the shortest path scheme (Dijkstra, 1959) to derive the FOM for more accurate object contours, and this is realized with an improved strategy over that used by Meier and Ngan (1999) because gradients of non-edge pixels are also taken into account as a way to increase robustness. The algorithm is also capable of detecting multiple unconnected objects.

Our proposed algorithm does not require any training/learning processing for foreground object segmentation. In addition, no continuous background maintenance method is used, as such method is bound to produce significant errors in unexpected situations due to unexpected and significant video scene changes which cannot be sufficiently taken into account (such as the more complicated learning-based foreground object detection and also the use of background maintenance method reported in Harville, 2002; Li et al., 2002; Li et al., 2004). The overall flow chart of the proposed algorithm is illustrated in Figure 1.

In summary, this paper proposes a fast, automatic and reliable video object segmentation algorithm suitable for content-based applications. The main contributions in the proposed approach consist of:

1. robust and adaptive change detection threshold estimation which facilitates single-pixel thresholding;
2. robust background reference frame generation;
3. adaptive reference frame selection; and
4. spatial refinement to change detection mask by incorporating moving edgels, edges, and gradients' information.

Figure 1. Overall flow chart for the proposed algorithm



PROPOSED ALGORITHM

Adaptive Change Detection

Change detection process examines the grey value difference between two neighboring image frames being considered (or the current frame and the generated background frame), and labels each pixel as either a *changed* or a *stationary* pixel when a function of its grey value difference is greater than or smaller than a certain decision threshold, respectively. In this section, strategies will be discussed to address the problems highlighted in the previous section for change detection.

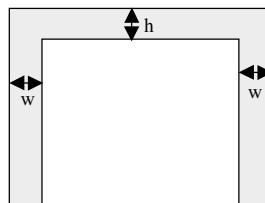
Adaptive Determination of Threshold

Correct determination of the decision threshold will result in lower error probability and thus greater robustness against noise. However, the threshold is often determined empirically (Kim et al., 1999; Mester & Aach, 1997; Shao, Lin, & Ko, 1998b) or requires prior knowledge of the noise variance of the camera for off-line estimation (Aach & Kaup, 1993).

The decision threshold is basically a scaled factor of the Gaussian-assumed standard deviation of camera noise. Although the decision threshold can be estimated off-line for the particular camera being used, this results in the need for system calibration for every camera being used and restricts the portability of the algorithm for different cameras. As a better approach, the decision threshold can be estimated online from unchanged regions with live video input. However, this is obviously impossible in general because the unchanged regions cannot be found before performing change detection (in fact, separation of unchanged regions is the very aim of change detection).

In order to deal with this chicken-and-egg problem, we propose to utilize appropriate domain knowledge for choosing a statistically stationary area (SSA)—the area with less statistical motion for the category of video under consideration. For most video scenarios captured with a stationary camera, strips bordering the left, right and top boundaries of the image (similar to the selection in Neri et al., 1998) can be chosen as the SSA (see Figure 2) since the stationary camera will be installed with the possible object of interest or target object usually occupying the central portion of the screen. Data in this SSA are less likely to contain part of the moving objects, and occurrence of moving objects in these areas can be filtered out using our following simple yet effective procedure.

Figure 2. Region for choosing data for computing threshold



Assume that d_i is the frame difference for a pixel in the *SSA*, where the *SSA* is hereby denoted as U , that is, $d_i \in U$, $i = 1, 2, \dots, n$ where n is the number of pixels in U .

The array of corresponding absolute-value data, $\{|d_i|\}$, is then sorted in an ascending order to give $\{|d_1|, |d_2|, \dots, |d_{n-1}|, |d_n|\}$. The median value of $\{|d_i|\}$ can be found:

$$m = |d_p| \quad (1)$$

$$\text{where } p = \begin{cases} n/2 & \text{if } n \text{ is even} \\ (n+1)/2 & \text{otherwise} \end{cases}$$

A binary attribute for d_i is defined as:

$$a_i = \begin{cases} 1 & \text{if } |d_i| \leq m \\ 0 & \text{if } |d_i| > m \end{cases} \quad (2)$$

The p elements with non-zero a_i in U constitute the sub-array of data U_p representing the most probably stationary pixels (*MSPSs*), and then the mean and the variance for U_p are calculated as follows:

$$\hat{\theta} = \sum_{i=1,n} a_i * d_i / p \quad (3)$$

$$\hat{\sigma}^2 = \sum_{i=1,n} a_i * r(i, \hat{\theta})^2 / p \quad (4)$$

where $r(i, \hat{\theta}) = d_i - \hat{\theta}$. The threshold, T , for change detection can then be determined by the standard deviation $\hat{\sigma}$:

$$T = 2.5\hat{\sigma} \quad (5)$$

Here, a scaling factor of 2.5 is used, based on the assumption that the data follow a Gaussian distribution and a factor of 2.5 encompasses a 99% confidence interval. This value has been found to be quite good empirically as a higher value results in many holes in the change detection mask while a lower value causes some noisy areas in the video frame to be labeled as moving pixels.

The above procedure tolerates 50% of pixels in the *SSA* to be part of moving objects (i.e., it can tolerate up to a maximum of 50% of outliers in the *SSA*). In fact, the actual tolerance is even higher since a small number of outliers in *MSPSs* do not significantly affect the resultant T , and the degradation of the estimate is graceful. Since the above procedure is efficient in implementation, it is possible to adaptively determine the threshold for each frame to cater to environmental (e.g., lighting and camera noise) changes with time.

Single-Pixel Thresholding

In the literature, a CDM is usually computed by thresholding the mean of a small neighborhood of the pixel difference in grey levels of the current frame and the chosen reference frame (Kim et al., 1999; Liu, Chang, & Chang, 1998; Mester & Aach, 1997; Neri et al., 1998; Shao, Lin, & Ko, 1998b). However, this result in a *blurring effect* since pixels in the neighborhood may not all be stationary or moving, yielding mistakes in those areas. In the proposed algorithm, a CDM is obtained by directly thresholding the grey-level, inter-frame difference of a single pixel.

The grey-level inter-frame difference for the i^{th} frame is:

$$D^i = I^i - I^{i-1} \quad (6)$$

where I^i represents the grey-level image at the frame. Understandably, highlights both moving objects and uncovered areas. An element in the resultant binary CDM is determined as follows:

$$m^i(x, y) = \begin{cases} 1 & \text{if } |d^i(x, y)| > T \\ 0 & \text{if } |d^i(x, y)| \leq T \end{cases} \quad (7)$$

where (x, y) represents the position of a pixel, and $d^i(x, y)$ is the corresponding element of D^i .

Single-pixel thresholding not only eliminates the *blurring effect* but also reduces the amount of computation required. The disadvantage is that this method produces more small *holes* in the resultant CDM. However, as mentioned earlier, having holes inside the mask is a problem in change detection anyway, and the problem can be solved by a post-processing with morphological operations that are to be described later.

Background Frame Generation

In any change detection process, the change detection mask can be obtained by performing the process on the current frame and a reference frame. The reference frame may be the immediately previous frame (as in the case of the video object segmentation algorithm in Aach & Kaup, 1993; Kim et al., 2001; Mester & Aach, 1997; Neri et al., 1998; Shao, Lin, & Ko, 1998b) or a fixed stationary background image as used in Cheung and Kamath (2004), Chien, Ma, and Chen (2002), Li et al. (2004), and Piccardi (2004). The latter approach is able to solve the problems of dis-occluded regions appearing as part of the segmented moving objects (for the case of a fixed stationary camera). In this second scenario, a simple approach is to obtain the fixed stationary background reference frame by an initialization phase where several background frames are averaged to obtain the reference. However, in doing so, it is inherently assumed that there cannot be any moving object in the numerous background frames captured in the initialization phase. This is not good if the camera spans field of view where it is difficult or even impossible to ensure that there are absolutely no moving objects in the scene during the camera setup and initialization phase (e.g., in public places, common corridors or crowded areas where people/vehicles are moving across the scene now and then). In order to remove such restrictions and solve the above-mentioned problem, a robust method for reference frame generation has been proposed.

This reference frame generation method uses the shortest-half robust statistical method (akin to least-median-squares (LMS) for 1-D data (Rousseeuw & Leroy, 1987)). The proposed method has a breakdown point of 50%, which means that it is able to reject up to 50% of outliers in the data set and is also more efficient in the presence of Gaussian noise (Rousseeuw & Leroy, 1987) compared to the use of a simple median filter (Cheung & Kamath, 2004; Piccardi, 2004). The LMS robust statistical method has also previously been shown to be capable of solving difficult optical flow estimation problems (Ong & Spann, 1999).

The proposed algorithm using the shortest-half statistical method is described as follows: Given a set of data $G = \{g_1, g_2, \dots, g_{2N}\}$ (which in our case are the various pixel values at the same pixel location from the $2N$ frames to be used for generating the reference stationary background frame), ϕ , which is the value to be computed and assigned to each pixel in the resultant reference frame, is defined in the model as:

$$g_x = \phi + e_x \quad (8)$$

where e_x is a noise term, and also writing g_x interchangeably with $g(x)$ for the convenience of notation. If e_x is an independently distributed normal random variable, it is well known that the arithmetic mean is the minimum least-squares (LS) estimator of ϕ . Unfortunately, least-squares estimation breaks down in the presence of outliers. The median-based estimator, however, is capable of rejecting up to 50% of the outliers and is the basis of the estimation procedure used in this work. In this case, the estimated ϕ is therefore given by:

$$\hat{\phi} = \arg \min_{\phi} \text{median}_{x \in G} r(x, \phi)^2 \quad (9)$$

where $r(x, \phi)$ is the residual of the x -th sample for a given value of ϕ , and $r(x, \phi) = g(x) - \phi$.

The robust standard deviation is then defined as:

$$\hat{\rho}_0 = 1.4826 \min_{\phi} \sqrt{\text{median}_{x \in G} r(x, \phi)^2} \quad (10)$$

where the factor 1.4826 is introduced to make $\hat{\rho}_0$ a consistent estimator of ρ when e_x is distributed as $N(0, \rho^2)$ random variables (as recommended in Rousseeuw, 1987).

Computation of $\hat{\phi}$ involves simply sorting the data in G in order to produce the sorted list: $G' = \{g'_1 < g'_2 < \dots < g'_{2N}\}$.

The LMS location estimator is then the mid-point of the shortest half sample:

$$\hat{\phi}_l = (g'_{v} + g'_{v+N})/2 \quad (11)$$

where:

$$v = \arg \min_{h=1..N} (g'_{h+N} - g'_h) \quad (12)$$

This is essentially a mode detection algorithm applied to the set of data.

The initial scale estimate is then used to determine a set of initial weights W , where an initial weight w_x is attached to each data point and these can be used to determine the outliers within the set G :

$$W = \{w_1, w_2, \dots, w_{2N}\}$$

$$w_x = \begin{cases} 0 & \text{if } |r(x, \hat{\phi}_1)| > 2.5\hat{\rho}_0 \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

The second step of the procedure is to use these weights to update the estimate:

$$\hat{\theta}_2 = \sum_{x=1}^{2N} w_x g_x / \sum_{x=1}^{2N} w_x \quad (14)$$

This location estimate $\hat{\theta}_2$ is then assigned to each pixel in the reference background frame (which will be used in the change detection process in the subsequent stages).

Since G is formed for a pixel position, the background construction can be meaningful as long as a moving foreground object does not always occupy that position for N frames or more. Similar to the *MPSP* determination for threshold estimation described earlier in this chapter, this method can tolerate at least 50% of outliers in G . However, even if there are slightly more than 50% of outliers in G , equation (14) is still able to give a graceful estimate.

Adaptive Reference Frame Selection

The proposed algorithm automatically switches the reference frame between the generated background image and the immediately previous frame, to allow maximum segmentation quality according to scenes being viewed and also to cater to significant changes in lighting and even accidental movement of the otherwise stationary camera, etc. When the generated background image is used as the reference frame for change detection, the proposed algorithm is able to solve the problems of dis-occluded regions appearing as part of the segmented moving object when large object motion occurs in the scene and thus obtain much more accurate segmentation results.

During the algorithm start-up, a background image is generated with the method previously presented in this chapter, and is used as the reference frame for change detection. The size of the changed areas is then constantly monitored. If the size of the changed areas is greater than 75% of the image size, the algorithm automatically switches to using the immediately previous frame as reference frame for change detection and then constantly monitors the size of the changed areas again. When it detects relatively small changed areas (less than 25% of image size) for a number of consecutive frames, it starts to generate a background image and then switches to using the generated background image as the reference frame again.

By using the stationary background image as the reference frame for change detection, the proposed algorithm is able to solve the problems of dis-occluded regions appearing as part of the segmented moving object when large object motion occurs in the scene and thus obtain much more accurate segmentation results.

Spatial Refinements of Change Detection Mask

For spatial refinements of the CDM, the following steps are performed: (1) preliminary refinement of the change detection mask using morphological filters, (2) extraction of disconnected objects, (3) mask modification and refinement using spatial features.

Preliminary Modification of Change Detection Mask

Selected binary morphological operations (Vincent, 1993) using 5x5 structuring elements are applied on the CDM to reduce the artifacts due to noise and less textured regions in moving objects, and solidify it to become a compact motion mask. More specifically, an opening by reconstruction is first performed to filter out those isolated small regions that are mainly due to background noise (outside the mask). Afterward, a closing is applied in order to fill those small holes inside the mask and also the narrow gaps among parts of the mask.

Then, by performing an erosion operation, a motion mask is obtained. The erosion combined with spatial mask modification tackles the over-segmentation problem caused by uncovered regions.

Extraction of Disconnected Objects

A motion mask may consist of several disconnected moving objects. Different connected regions are extracted from the mask using a connected-component labeling process (Gonzalez & Wintz, 1987). These individually connected components are regarded as different extracted objects in the resultant POM and will be treated separately in the spatial processing that follows.

Mask Modification and Refinement with Spatial Features

Temporal manipulation presented previously provides the rough mask (POM) for the foreground objects in the scene. Spatial features in the image carry more information about precise object contour and therefore are used for mask modification and refinement in this section. Moving edgels, edges and gradients are adopted in two steps to reconfirm and modify the POM.

Filling-in Process for Each Connected Component

There may still be holes inside the mask even after the morphological operations and some part of the object boundary is not detected correctly in the POM. Because of its coherence with foreground objects, moving edge over the reference frame is incorporated to authenticate and consolidate the mask. This also creates a chance for the mask to be corrected for the inaccuracy with erosion in the previous phase.

Edgels in both the current frame and the reference frame are obtained via the Canny's edge detector (Canny, 1986). Each edgel in the current frame is regarded as a moving edgel if no corresponding edgel can be found in the same location of the reference frame. If the reference frame is the generated background image, all edgels belonging to foreground objects may be detected as moving edgels.

Then, a region-filling method (similar to the one used by Meier & Ngan (1999)) is employed to fill in the region encompassed by moving edgels that are authenticated to be the correct object boundary edgels. More precisely, those moving edgels that are found to be in the vicinity of the boundary of the POM are first labeled as the correct edgels. The filling-in process is employed to every row for the pixels between the first and the last correct edgels with each connected-component in the mask. The same procedure is then repeated for each column and once more for each row. The technique is based upon the assumption that foreground objects are mostly non-concave to alleviate the effect of missing edge in the neighborhood of object boundary. Any protruding branches or thin lines that fail to form a closed contour are then identified and removed.

Contour Refinement Using Shortest Path Technique

The above-mentioned filling-in process results in blockiness in certain parts of the object mask where edge could not be found in the vicinity. In order to remedy these false outlines, a method that incorporates graph-based edge-linking techniques has been utilized.

First, those mask boundary segments that do not correspond to the edgels are removed because they must have been created by the filling-in process, while all the other parts of the mask boundary will remain unchanged. After each wrong boundary segment is removed, a gap is created in the otherwise closed contour. The correct boundary between the two endpoints of this gap will then be determined using a shortest path technique.

In a shortest path algorithm, each pixel can be regarded as a node to be linked to a neighboring pixel by an arc after the associated weight (or distance) is evaluated. Pixels that already belong to the correct object boundary are excluded from the search. The weight for edgels is set to a constant q_0 , and each non-edge pixel is assigned a weight, $q(x,y)$ (being always greater than q_0), which is determined according to the magnitude of its gradient $z(x,y)$:

$$q(x,y) = q_s + (q_l - q_s)(z_l - z(x,y))/(z_l - z_s) \quad (15)$$

where z_s and z_l are the smallest and the largest magnitudes of the gradients in the image respectively, and q_s and q_l are the predetermined smallest and largest weights to be assigned. As can be seen from equation (15), the larger the $z(x,y)$, the smaller the $q(x,y)$ becomes (therefore the corresponding pixel has more resemblance to an edge pixel).

In the proposed work, the best (shortest) path between the two endpoints with respect to the defined weights is found using the Dijkstra's shortest-path method (Dijkstra, 1959) to obtain the final object mask (FOM).

The proposed edge-linking approach is an improved method over the one used by Meier and Ngan (1999), the latter employs only two fixed weight values for edge and non-edge pixels and consequently has the undesirable property of producing straight links between two correct edgels whenever there are no or very few other edgels within the region where the two correct edgels are to be joined. The advantage of the proposed approach is that the edge-linking process takes into account the possibility of the presence of weak edges that cannot be confidently detected by the edge detector. Thus,

the proposed approach will be able to handle cases of broken and missing edges that are commonly encountered in real-life video.

EXPERIMENTS AND PERFORMANCE

The video object segmentation algorithm has been implemented in C codes running on a personal computer (PC). The software accepts digitized live YUV images from a video camera. The segmented result together with the original input image is displayed on the screen of the PC. In all the experiments, the following parameter values have been set for formulae (15): $q_0 = 1$, $q_s = 5$, and $q_i = 20$.

Robust Background Frame Generation

To illustrate the effectiveness of the proposed algorithm in generating a background frame that is robust against the presence of moving objects, a sequence of 20 frames taken with the aforementioned experimental setting, of which nine frames contain moving objects, is used to generate the background frame. Figure 3a shows one of the 11 frames (where only the luminance component of the image is shown) that only contains stationary background used in generating the background frame, while Figure 3b shows one of the nine frames that contains moving objects. The background frames generated using simple averaging and the algorithm presented earlier are both shown in Figure 4a and 4b, respectively.

Figure 3. Images used for background frame generation: (a) one of the frames containing only stationary background; and (b) one of the frames containing moving objects

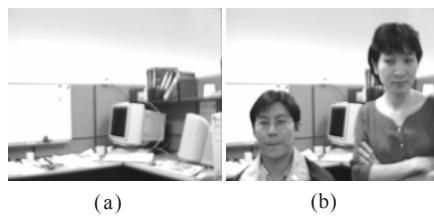
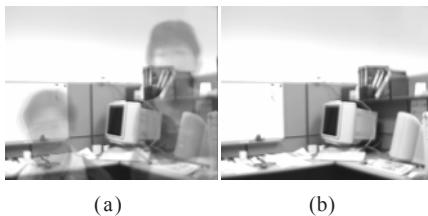


Figure 4. Background frame generated using: (a) simple averaging; and (b) proposed robust algorithm



It can be seen that in Figure 4a, since simple averaging across all the frames is used to generate the background frame, the background frame contains traces of the moving objects. Figure 4b shows that the proposed algorithm is able to eliminate the moving objects when the background frame is being generated. In this example, the proposed algorithm has shown the robustness of background frame generation in the presence of 45% of outliers in the data.

Segmentation of Foreground Objects

Figure 5a and 5b show a snapshot in the middle of a live video clip (capturing two persons in the laboratory) and its segmentation results. It can be seen that two semantic foreground objects (persons) have been segmented nicely in this snapshot.

Figure 6 compares two examples with the immediately previous frame and with the stationary background image as a reference frame, respectively, when relatively large motion occurs between two consecutive frames. The central and right columns of images

Figure 5. Segmentation for two foreground objects in a live video session: (a) a snapshot in the middle of a live video session, (b) segmentation results



Figure 6. Comparison of segmentation results with two options for reference frame when large motion occurs. Left column: original images; central column: segmented results with immediately previous frame as reference frame; right column: segmented results with the stationary background image as reference frame



Figure 7. Examples of video object segmentation results: Original image (on the left), and segmented results of proposed algorithm (on the right)



in Figure 6 show the segmented results with the immediately previous frame and the stationary background image as the reference frame, respectively. The latter obviously gives more accurate segmentation because a more precise CDM is detected.

A few samples of the segmented results from a long stream of video sequence are shown in Figure 7.

Figure 7 shows that the proposed algorithm that builds a robust background reference frame is able to solve the problem of dis-occluded regions appearing as part of the segmented moving objects and the segmented results are relatively consistent through a long video sequence. Similar results have been observed on many different video sequences and real-time video capture and segmentation.

Comparison on Accuracy and Processing Speed

For the purpose of comparison, the FUB (Fondazione Ugo Bordoni) technique (Neri et al., 1998), one of the three MPEG-4 informative segmentation tools (MPEG-4 Video Ver-2 WD, 1999) mainly based upon change detection and implemented in C codes, has been tested on the same Pentium-4 2.8GHz PC. We have compared our method with the FUB method, as it has been verified and included in the informative part of MPEG-4 International Standard.

In order to calculate the fault rate without manual segmentation, two simple video sequences (70 frames each) are synthesized from two pictures taken in the laboratory by making copies of arbitrary rectangular blocks in each picture and moving the block(s) across the picture, as shown in Figure 8 (the straight lines with arrows indicating the motion direction of the blocks are also highlighted with rectangular frames in white). Sequence 1 has one foreground object while Sequence 2 has two foreground objects. All objects complete the movement along their trajectories in about 65 frames.

The average segmentation fault rate is defined as:

$$\bar{F} = \sum_{i=1,f} (M_{ow}^i + M_{bw}^i) / (M * f) \quad (16)$$

where M_{ow}^i and M_{bw}^i are the number of pixels that belong to the foreground objects but are wrongly classified as background and the number of pixels that belong to the background but are wrongly classified as foreground at the i^{th} frame, respectively. M is the number of pixels in a frame while f is the number of frames under measurement.

Figure 8. Two synthetic video sequences for testing (the straight line with arrow indicates the direction of motion for the block and the rectangular frame for each block is for illustration only): (a) Sequence 1 with one moving object; (b) Sequence 2 with two moving objects

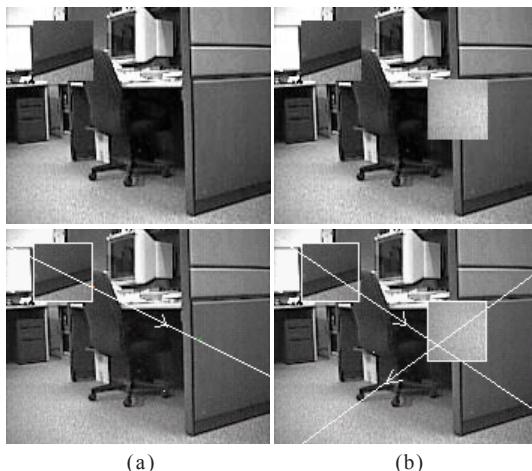


Table 1. Comparison of the FUB technique and the proposed algorithm in terms of average fault rate

	\bar{F} (%)	
	FUB Technique	The Proposed Algorithm
Sequence 1	1.34	0.44
Sequence 2	2.84	1.88

Table 1 compares the FUB technique and the proposed algorithm in terms of average fault rate. As can be seen, the proposed algorithm results in significantly smaller average segmentation error than the FUB technique with both one foreground object and two foreground objects.

In addition, the proposed algorithm takes merely about 0.083 seconds to segment video objects from a QCIF frame, while the FUB technique needs around 33 seconds to process the same frame on the same machine. It should be noted that the C codes in both implementations have not been optimized in any aspects. The processing speed of about 0.083 sec/frame for the proposed algorithm has been confirmed by experiments for other test sequences.

CONCLUSIONS AND FUTURE WORK

Targeted at the type of scenario with static camera, an algorithm has been presented for automatic and robust real-time multi-object segmentation with reasonably low fault rate.

With the prospect of real-time realization towards online camera inputs, the change detection approach is adopted as temporal segmentation due to its efficiency, and direct single-pixel thresholding is used for further speedup and for avoidance of blurring effect. The algorithm is fully automatic since there is no need for user interference at any stage and the threshold for change detection is adaptively decided without empirical selection or prior calibration for the camera used. The reference frame for change detection is also auto-selected from the immediately previous frame or the robustly-generated background frame to allow maximum segmentation quality according to different circumstances. With the construction of the background reference frame, the algorithm is able to solve the problems of dis-occluded regions appearing as part of the segmented moving objects.

For the algorithm to be useful with live video input, robustness is another important consideration. The threshold determination scheme for change detection compensates camera and environmental noise dynamically, and ensures correct calculation when at least 50% of the pixels in the statistically stationary area (SSA) are not in motion (if there are more than 50% of pixels in motion, the degradation of the estimate is graceful). The same tolerance rate and similar gracefulness are exhibited in the proposed background frame generation. Use of multiple features (motion, edge and gradients) in different steps creates chances for inconsistent object boundary to be adjusted. Inaccuracy of object

contours due to broken and missing edges is alleviated by utilizing gradients in spatial refinement. Only edges and gradients are utilized in the spatial processing stage as a compromise of segmentation quality and efficiency.

Experiments confirm the effectiveness of the algorithm in segmenting multiple objects in the targeted scenario via live camera input. The algorithm requires much less computation than methods depending on motion parameter estimation and other change-detection-based approaches. The C-implementation of the algorithm runs at about 12 frames per second on a Pentium-4 2.8GHz PC before any code optimization. Significant speed-up (x5 to x10, based on the previous experience (Lin et al., 1999)) is expected when coding and process are optimized. Processing speed can be improved further if foreground objects once segmented are tracked (instead of being segmented at every frame) for a fixed number of frames or until scene change is detected.

The algorithm can be applied to applications such as real-time MPEG-4 content-based video coding with video object planes (VOPs) being automatically extracted online. Other possible applications include home/office/warehouse security where monitoring and recording of intruders/foreign objects, alarming the personnel concerned or/and transmitting the segmented foreground objects via a bandwidth-hungry channel during the appearance of intruders are of particular interest.

The algorithm may be extended for other scenarios (e.g., traffic/vehicle monitoring) if domain knowledge is incorporated (e.g., for determining SSA). It may also be extended for the cases with cameras in motion because dominant motion can be detected (Kim et al., 1999; Mech & Wollborn, 1997; Shao, Lin, & Ko, 1998) and change detection can then be performed with the current frame and the motion-predicted frame. If different classes of foreground objects need to be discriminated with motion velocity, each resultant connected component in the object mask can be divided into multiple regions based upon spatial homogeneity (Li et al., 2001). Motion velocity for every region can be estimated and all regions would be grouped according to a motion similarity measure (e.g., the one suggested by Li et al. (2001)). This attempt is expected to be less computationally expensive than those using motion estimation for the whole image, especially when foreground objects occupy a relatively small portion of it.

REFERENCES

- Aach, T., & Kaup, A. (1993). Statistical model-based change detection in moving video. *Signal Processing*, 31, 165-180.
- Bors, A. G., & Pitas, I. (1998). Optical flow estimation and moving object segmentation based on median radial basis function network. *IEEE Transactions on Image Processing*, 7(5), 693-702.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8 (pp. 679-698).
- Castagno R., Ebrahimi, T., & Kunt, M. (1998). Video segmentation based on multiple features for interactive multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5), 562-571.
- Chen, L. H., Chen, J. R., & Liao, H. Y. (2000). Object segmentation for video coding. In *Proceedings of the 15th International Conference on Pattern Recognition* (pp. 383-386).

- Chen, T., & Swain, C. (1999). *Method & apparatus for segmenting images prior to coding* (U.S. Patent 5960111).
- Cheung, S. C. S., & Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. *SPIE Electronic Imaging: Video Communications and Image Processing*, San Jose, CA (pp. 881-892).
- 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*, 12(7), 577-585.
- Cucchiara, R., Onfiani, P., Prati, A., & Scarabottolo, N. (1999). Segmentation of moving objects at frame rate: A dedicated hardware solution. In *Proceedings of the 7th International Conference on Image Processing and Its Applications* (pp. 138-142).
- Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271.
- Gonzalez, R. C., & Wintz, P. (1987). *Digital image processing* (2nd ed.). Reading, MA: Addison Wesley.
- Harville, M. (2002). A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *Proceedings of European Conference on Computer Vision* (pp. 543-560).
- Kim, M., Choi, J. G., Kim, D., Lee, H., Lee, M. H., Ahn, C., & Ho, Y. S. (1999). A VOP generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), 1216-1226.
- Kim, M., Jeon, J. G., Kwak, J. S., Lee, M. H., & Ahn, C. (2001). Moving object segmentation in video sequences by user interaction and automatic object tracking. *Image and Vision Computing*, 19, 245-260.
- Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., et al. (1994). Towards robust automatic traffic scene analysis in real-time. *Proceedings of International Conference Pattern Recognition*, Jerusalem, Israel (pp. 126-131).
- Li, H., Tye, B. J., Ong, E. P., Lin, W., & Ko, C. C. (2001). Multiple motion object segmentation based on homogenous region merging. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, Sydney, Australia (pp. 175-178).
- Li, L., Gu, Y. H., Leung, M. K., & Tian, Q. (2002). Knowledge-based fuzzy reasoning for maintenance of moderate-to-fast background changes in video surveillance. In *Proceedings of the 4th IASTED International Conference on Signal and Image Processing*, HI (pp. 436-440).
- Li, L., Huang, W., Gu, Y. H., & Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11), 1459-1472.
- Lin, W., Tye, B. J., Ong, E. P., Xiong, C., Miki, T., & Hotani, S. (1999). Systematic analysis and methodology of real-time DSP implementation for hybrid video coding. In *Proceedings of IEEE International Conference on Image Processing*, Kobe, Japan (pp. 847-851).
- Liu, H. C., Hong, T. H., Herman, M. & Chellappa, R. (1998). Motion-model-based boundary extraction and a real-time implementation. *Computer Vision and Image Understanding*, 70(1), 87-100.

- Liu, S. C., Chang, W. F., & Chang, S. (1998). Statistical change detection with moments under time-varying illumination. *IEEE Transactions on Image Processing*, 7(9), 1258-1268.
- Mech, R., & Wollborn, M. (1997). A noise robust method for 2D shape estimation of moving objects in video sequence considering a moving camera. *Signal Processing*, 66, 203-217.
- Meier, T., & Ngan, K. N. (1999). Segmentation and tracking of moving objects for content-based video coding. *IEEE Proceedings on Vision, Image and Signal Processing*, 146(3), 144-150.
- Mester, R., & Aach, T. (1997). *Method for change detection in moving images* (U.S. Patent 5654772).
- MPEG-4 Video Ver-2 WD. (1999). *MPEG-4 doc. ISO/IEC JTC1/SC29/WG11/M4583 (FDIS 14496-2)*, Annex F (Informative), Seoul.
- Neri, A., Colonnese, S., Russo, G., Talone, P. (1998). Automatic moving object and background segmentation. *Signal Processing*, 66(2), 219-232.
- Odobez, J. M., & Bouthemy, P. (1998). Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2), 143-155.
- Ong, E. P., & Spann, M. (1999). Robust optical flow computation based on least-median-of-squares regression. *International Journal of Computer Vision*, 31(1), 51-82.
- Piccardi, M. (2004). Background subtraction techniques: A review. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, The Hague, The Netherlands (pp. 3099-3104).
- Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. New York: John Wiley & Sons.
- Shao, L., Lin, W., & Ko, C. C. (1998). Dominant motion segmentation of video objects towards object-oriented video coding. In *Proceedings of the 6th IEEE International workshop on Intelligent Signal Processing and Communication Systems*, Melbourne, Australia (pp. 258-261).
- Shao, L., Lin, W., & Ko, C. C. (1998b). Video object segmentation based on change detection and region growing. In *Proceedings of IASTED International Conference on Signal and Image Processing*, Las Vegas, NV (pp. 387-391).
- Siyal, M. Y., & Fathy, M. (1995). Real-time measurement of traffic queue parameters by using image processing techniques. In *Proceedings of the 5th International Conference on Image Processing and its Applications*, Edinburgh, UK (pp. 450-454).
- Toklu, C., Tekalp, M., & Erdem, A. T. (2000). Semi-automatic video object segmentation in the presence of occlusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(4), 624-629.
- Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2), 176-201.

Chapter VIII

A Fully Automated Active Shape Model for the Segmentation and Tracking of Unknown Objects in a Cluttered Environment

Besma Rouai-Abidi, University of Tennessee, USA

Sangkyu Kang, LG Electronics Inc., Korea

Mongi Abidi, University of Tennessee, USA

ABSTRACT

The segmentation of shapes is automated using a new objective function to deform and move a contour toward the actual shape. New profile modeling and optimization criterion to automatically find corresponding points are also applied for segmentation and tracking of people in cluttered backgrounds. The proposed framework utilizes a Pan-Tilt-Zoom (PTZ) camera and automatically performs the initial target acquisition through motion and color-based segmentation. Successful results are presented for within and between frame segmentation and tracking. This algorithm presents a major extension to the state of the art and the original active shape model (ASM) which was designed for known objects in smooth non-changing backgrounds and where the landmark points need to be manually picked off-line. This is a fully automated, real time ASM that deals with changing backgrounds and does not require prior knowledge of the object to be segmented and tracked.

INTRODUCTION

Let us assume that an object of interest is a clear glass bottle containing a liquid of some sort. If this content is distinct from its surroundings, the color of the content will be the best means to locate the glass bottle. On the other hand, if there is nothing in the glass bottle, color is no longer an apparent feature. Other features including silhouette (shape), texture, edges and shadows are required as a substitute for color to characterize the bottle for subsequent location and tracking. Texture and edges on the bottle are good features if the bottle only shows translational movement, parallel to the imaging sensor. If the bottle incurs rotational motion, texture and edges will become self-occluding. The amount of degradation caused by rotation in silhouettes is usually smaller than that of edges.

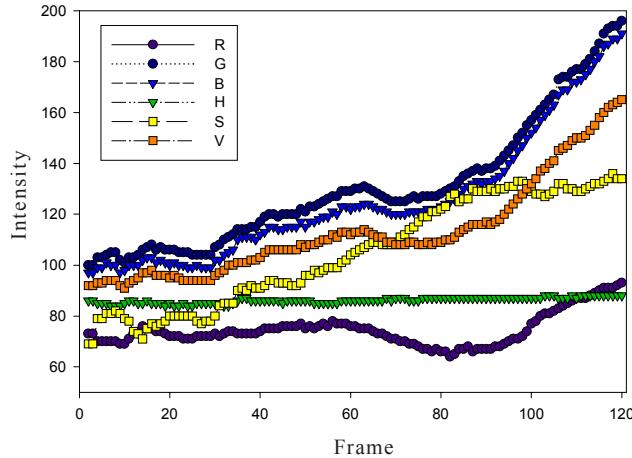
A priori information on texture and/or silhouette can help deal with occlusion problems by restricting the search to variations of this prior information. With texture being more vulnerable to camera motion than silhouette, color and shape remain as optimum choices for the description of an object for tracking.

Color is an efficient feature for tracking due to its computational simplicity. Color, however, has disadvantages under abrupt illumination changes, especially when using a pan/tilt/zoom (PTZ) camera in indoor environments, where many small light sources, windows and glass walls may be present. Multiple static cameras can be used instead of one PTZ camera, but this setup has its own shortcomings, such as object handover, color constancy between cameras, camera placement and cost.

Figure 1. An example of color variations: (a) The first frame, (b) the 199th frame, (c) extracted shirt from (a), and (d) extracted shirt from (b)



Figure 2. Color variations on 120 consecutive frames from the PTZ camera



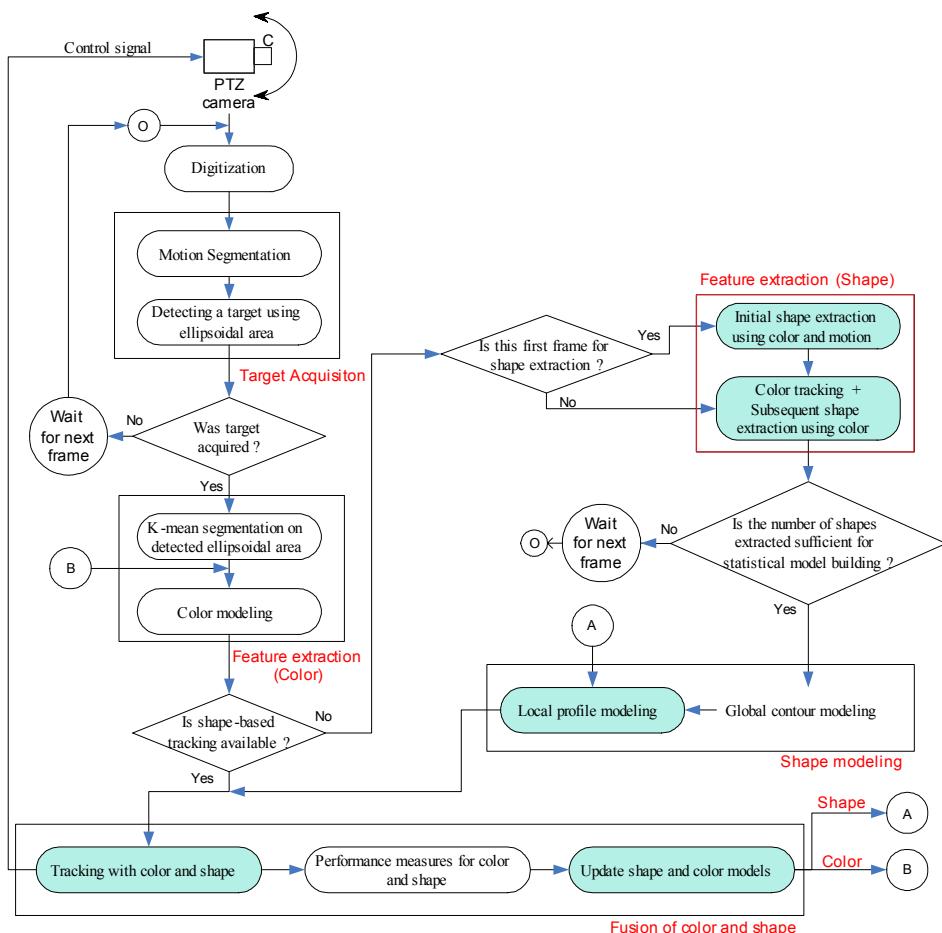
The examples in Figures 1 and 2 show the limitations of using of a PTZ camera inside a local airport. An intruder was tracked based on the color of her shirt which was recorded for 120 frames. The first and 119th frames are shown in Figures 1a and 1b, and extracted shirt areas in Figures 1c and 1d, respectively. The time gap between the first and the 119th frame is about 10 seconds. The variations in the mean values of each color channel are illustrated in Figure 2, which shows rapid changes in all channels except Hue values.

Shape can be used to assist color in the presence of illumination changes, occlusion and similar colors in the background. Shape-based tracking is usually insensitive to illumination changes since it typically uses gradients of pixels. Deformable shape tracking can be divided into snake and statistically-based methods. The snake method, first introduced by Kass, is akin to an elastic band pulling a shape towards the edges (Kass, 1987). The active shape model (ASM), introduced by Cootes, is a statistical method that utilizes prior information on shapes to perform tracking (Cootes, 1995). Snake methods impose limited or no constraints on shape, which often results in erroneous results. On the other hand, the ASM is less sensitive to complex backgrounds and generates more plausible shapes by using training sets. Since surveillance systems are required to track the object of interest under various conditions, including partial and whole occlusion, the ASM was deemed more suitable for this task. In this work, the training of shapes required for the ASM, including selection of landmark points, is automated and done online in real time. A new profile modeling approach suitable for cluttered and changing backgrounds is also introduced. If using the original ASM profile model, pixels on both sides of each landmark point should be considered. This, however, will introduce errors when the background is not constant. The original ASM assumed a Gaussian distribution of the profile and used the Mahalanobis distance to find the nearest point to each landmark point. But with a changing background, the profile can no longer be modeled as Gaussian. In this work, only pixels inside the contour are considered and a new matching function is used to establish correspondences.

In this tracking, color is used in a limited way and on consecutive frames, where variations can be ignored when initiating the training shapes but assist in moving a shape to the real boundary of the target. Selecting a distinct color after motion-based segmentation eliminates manual initialization for a fully automated tracking. The overall system is shown in Figure 3.

The following section reviews related work in the areas of tracking, with emphasis on the latest developments in shape-based tracking. Section 3 describes the core contributions of this chapter in terms of automating initial shape detection, with the ASM model building both in terms of global contour as well as local landmark points and their profiles. New algorithms in profile representations are introduced and an objective function for optimum modeling is formulated. Shape tracking using the statistical model

Figure 3. The proposed system for real-time video tracking using color-assisted shape. The green boxes show the functions detailed in this chapter.



thus built is also described and experimental results presented. Section 4 summarizes the contributions of this chapter and gives suggestions for future work.

BACKGROUND AND RELATED WORK

Most popular tracking algorithms use background subtraction due to its simplicity and fastness to detect movement. *W⁴* (Haritaoglu, 2000) and VSAM (Collins, 2000) are examples of systems utilizing background subtraction for target acquisition. Static cameras are usually used with background subtraction algorithms. Extensions to non-static cameras (Medioni, 2001; Ren, 2003) and PTZ cameras (Dellaert, 1999; Kang, 2003) only work in non-crowded scenes with no major changes in the background.

Comaniciu proposed a kernel-based color algorithm to track a target after a user initializes the area to be tracked and showed its good performance on crowded scenes and under rapid movements (Comaniciu, 2003). But color-based algorithms will always suffer from changes in illumination. This can be compensated for by using shape, either in the form of contours attracted to edges, that is, snakes (Kass, 1987) or shapes generated from a statistical model, that is, ASM (Cootes, 1995). The ASM provides statistical shape modeling that generally shows better performance than the snake when intensities on either side of the boundaries are not very distinct. ASM requires a training phase and is not restricted to the tracking of a single object (Zhao, 2004).

Snakes, or deformable active contours, were first introduced by Kass (Kass, 1987) as elastic bands that are attracted to intensity gradients. They can be used to extract outlines if a target object is distinct from the background. But snakes exhibit limitations when the initialization is not well done or the object presents concavities. In this case, the algorithm oftentimes converges to the wrong shape. Other research uses shape parameters such as smoothness and continuity to deal with topological changes. These algorithms fail when more than one object splits from the initial shape, such as a group of people who split and walk in different directions. A geodesic active contour was developed by Caselles (Caselles, 1995) and a level set approach proposed by Osher (Osher, 1988) and used for curve evolving, which makes the contour merge and split depending on the image. These algorithms require additional steps, including the computation of difference images, which are not evident for non-stationary views from a PTZ camera, and the suppression of edges from the background.

The majority of active contour-based algorithms introduced to detect moving objects use features from the area enclosed by a contour, rather than the statistical characteristics of deformations on the contour. In other words, in case any feature used by these algorithms incurs errors, the algorithm may converge to the wrong shape altogether. Examples of errors include wrong motion compensation to segment foreground objects and errors in edge detection.

Another variation of shape-based tracking uses prior information on the target. For example, if the mean shape of the target is known beforehand, this shape information can be integrated into an objective function that computes the distance between the current shape and that mean shape (Amit, 2002). While this algorithm uses one prior mean shape with variations for each point, there are different approaches to utilize a group of shapes in the training stage. Cremers considers shape variations as a Gaussian distribution and computes similarity between the current shape and the training set using a distance

similar to the Mahalanobis distance (Cremers, 2002, 2003). This approach is different from the ASM in the sense that no projection onto the eigenvectors is performed. However, the ASM is more suitable when tracking a complex object since the ASM uses a statistical model for each landmark point to locate the corresponding point. Shen proposed an attribute vector for each point, which is a triangle that connects a current point to nearby points and describes both local and global shape characteristics (Shen, 2000). While this method is based on a snake's energy minimization function, one aspect is different from the general snake's capture of the prior shape. This method actually updates a prior shape during energy minimization, whereas an Euclidean distance is used for the general snake. A deformation of the prior shape is achieved by an affine transform on the attribute vector to adaptively find a shape in an input image. Shen also proposed a PCA-based statistical shape model (Shen, 2000). This is the so called an adaptive-focus statistical model with different weights used to represent small variations ignored by previous ASMs due to their small eigenvalues. Even though Shen's method used prior shapes for segmentation, ASM incorporates a more general statistical shape model for shape variations, which is more suitable for complex images. ASM variations used different profile and statistical modeling and a population of training sets to get more variations from the statistical model. Ginneken and Bruijne introduced a new way to characterize each landmark point for ASM-based tracking. They use an N by N window around each landmark point and kNN to find matching points. Experimental results include the segmentation of the cerebellum and corpus callosum from MRI brain images (Ginneken, 2002; Bruijne, 2003). Wang utilized a different approach to populate a given shape to represent various deformations. It is achieved by moving several points from a training set and building shape models after applying a smoothness constraint (Wang, 2000). This is useful when the number of training shapes is limited, which results in restricted shape variations. While Wang populated shapes in the spatial domain, Davatzikos proposed another method to populate training sets using a hierarchical ASM based on wavelet decomposition of extracted shapes (Davatzikos, 2003). Wavelet decomposition divides shapes into a number of bands, and PCA is performed on each band. As a result, the number of eigenvectors is the same as the number of bands. Therefore, these eigenvectors encompass a relatively large subspace, which removes the over-constraining problems of the original ASM. Edges are used to attract landmark points and the algorithm was applied to track corpus callosum and hand contours.

Statistical shape models, or prior shapes, are useful for applications where a reasonably well defined shape needs to be extracted. Medical imaging is a good example, since the shapes of internal organs vary according to a patient's age, height, weight, etc., but the overall structure remains similar in all of these. If the internal organ is of simple shape, snake-based algorithms can be used. For example, Xu used a snake algorithm for segmenting the left ventricle of a human heart (Xu, 1998) and Debreuve applied space-time segmentation using level set active contour to myocardial gated single photon emission computed tomography (Debreuve, 2001). Moreover, Han used minimal path active contour models for segmenting the human carotid artery without prior shape information (Han, 2001). These can be achieved because of the simple structure or good separation of organs of interest from surrounding tissues.

Although many applications of medical imaging show good performance in segmenting internal organs, only a few ASM-based algorithms have appeared for tracking

objects other than internal organs, such as pedestrians and hands. The main difference between medical images and other applications is in the complexity of the background. Medical images are usually represented in black and white, depending on the density of tissues, and organs of interest always have different intensity levels than their surrounding tissue. The original ASM was applied to medical imaging and uses profile information for each landmark point including pixel values from inside and outside the boundary. This, however, is no longer an option when the background changes, that is, profile information for each pixel does not have the same background component as in medical imaging. Although Baumberg used ASM for tracking a pedestrian after automatic shape extraction through background subtraction (Baumberg, 1995), profiles containing background pixels for each landmark point can confuse the algorithm in locating a correct point if the background is complex and changing. For this reason, a new way of characterizing landmark points will be presented in this chapter.

The closest work to solving the problem of PTZ-based tracking in a cluttered and complex background under variable illumination can be found in Nguyen (2002). Nguyen proposed contour tracking of deformable objects acquired by a moving camera. Edge map and motion compensation are used to remove any edges from the background and other moving objects. This method is relatively sensitive to illumination changes but is not sensitive to occlusions. If a partial occlusion occurs for few frames, the extracted shape would no longer be similar to the contour given by the user and the object being tracked would be lost.

PTZ TRACKING WITH ACTIVE SHAPE MODELS

As noted before, when illumination conditions are not constant, such as when glass walls and windows let direct sunlight get through, video signals from consumer-level cameras often saturate because of their low dynamic range. It is also likely that objects similar in color to the target appear in the scene. In both cases color-based tracking often fails. This led to investigating an additional feature for a more robust tracking. One of the restrictions of this work was to use consumer level equipment without including additional expensive infrared or 3D sensors.

Shape was selected as a second feature, defined as the boundary of a target, and will obviously vary when the target moves. Since shape is independent of illumination changes, it can be used for tracking when the color detector shows low performances. Prior shape models will be used, but since all moving objects have different shapes and patterns of variations, a single shape model can not be used to represent all possible moving objects. Online training is required for the efficient acquisition and modeling of shape variations. This online modeling represents the first extension to the original ASM algorithm, which was restricted to offline modeling and manual landmark point selection.

It is assumed that the moving shape is the contour of the area extracted during initial color-based tracking and that the color between two consecutive frames remains relatively constant (see Figure 2). The overall procedure for tracking when using shape consists of three major processes: shape extraction, shape modeling and tracking. In the first stage, once the target is acquired from the PTZ camera using motion-based

segmentation, both motion and color are used to extract the first shape, and only color is used to extract shapes afterwards. The reason for this is that motion information is no longer available once color-based tracking starts because the PTZ camera will vary its angles to follow the target. Once a number of shapes are extracted while the system initially performs color-based tracking, a principal component analysis is used to build a statistical shape model to be used later for the tracking of the shape. Landmark points are also selected and modeled for a more robust shape tracking.

Shape Segmentation for Statistical Model Building

An initial shape extraction using motion and color will be presented first, followed by subsequent shape extraction using color only.

Segmentation of Initial Shape Using Color and Motion

An initial shape is extracted using a threshold value determined by the mean shift filter. Since no prior information is available, a rectangle is used as search window and placed at the center of an ellipsoid region detected using top-down motion on the input image $I(x, y)$, which represents the similarity between the current frame and the color model acquired during the target acquisition step (Kang, 2004). The threshold value is determined by:

$$th = \beta \cdot \frac{\sum_{x \in W_0, y \in W_0} I(x, y)}{K_w \times K_h} \quad (1)$$

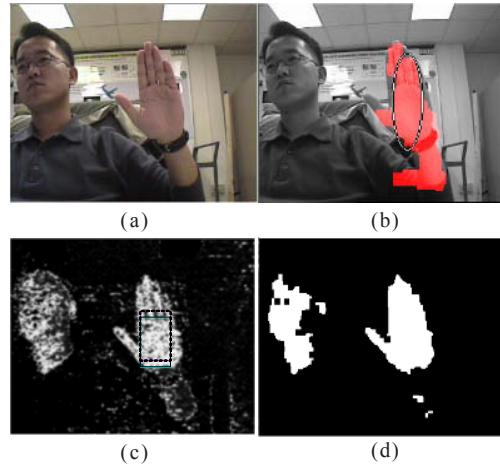
where K_w and K_h are the width and height of the search window W_0 and β is a coefficient empirically determined to be 0.7. Thresholding is performed on $I(x, y)$ using the value th . Examples of this process are shown in Figure 4.

If an object with similar color to the target is present in the image, more than one blob will appear in the binarized image as shown in Figure 4d. Since the initial shape was acquired using the detected ellipsoidal region, the desired blob B_i will be determined as being closest to the ellipsoidal region.

In the traditional ASM (Cootes, 1995), landmark points are manually placed at the same locations of forward facing objects, which is not possible with randomly moving objects seen by a PTZ camera. Landmark points in this case need to be established differently. The blob, B_i , is used to extract the major and minor axes of the object and the minor axis used as a reference angle for extracting the landmark points on the contour. The first landmark point for the shape is placed along the minor axis on the blob's boundary and a search is performed from the centerpoint to the boundary in the direction of the following vector:

$$\mathbf{u}_k = \begin{pmatrix} \cos \frac{2\pi k}{N} \\ \sin \frac{2\pi k}{N} \end{pmatrix} \quad (2)$$

Figure 4. An example of determining the threshold value using the mean shift filter for initial shape extraction: (a) original image; (b) result of motion-based segmentation. Region inside the ellipse is used to determine the distinct color of the target; (c) color similarity image P_c computed by histogram backprojection and result after applying the mean shift filter. The initial location of the mean shift kernel is shown with the dotted rectangle and the final location is shown with the solid rectangle; (d) the binarized and denoised version B_{P_c} of (c).



where \mathbf{u}_k represents the k -th unit vector for $k = 0, 1, \dots, N - 1$ and N represents the number of desired landmark points on the boundary of B_i . The procedure to find the initial shape s_0 from a single frame is shown in Figure 5.

An example of an initial shape extraction is shown in Figure 6. The detected ellipsoidal area segmented using motion analysis is shown in Figure 6b. Blue is detected as the object's distinct color and the extracted shape s_0 is shown in Figure 6c with its minor axis.

Segmentation of Subsequent Shapes Using Color

The initial shape as extracted in the previous section may not, in some cases, be fully representative of the actual shape since it is always subject to the limitations of motion-based segmentation. An example is shown in Figure 7, where the red area, resulting from motion-based segmentation, is smaller than the target and the initial shape does not fully characterize the actual shape of the target.

The initial shape, however, has a very important role in this whole framework because of two factors: (1) it is highly unlikely that this initial shape contains pixels from the background, and (2) the color values of pixels inside the initial shape will be very similar to their values in the next frame. This color model is constantly updated every time a new shape is extracted by substituting the previous color model with colors from the

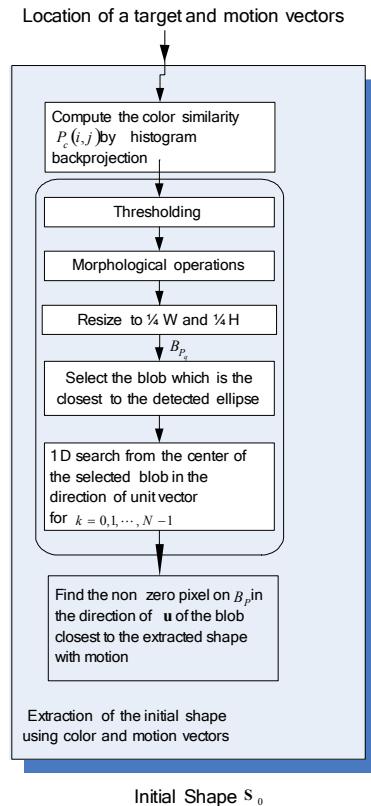
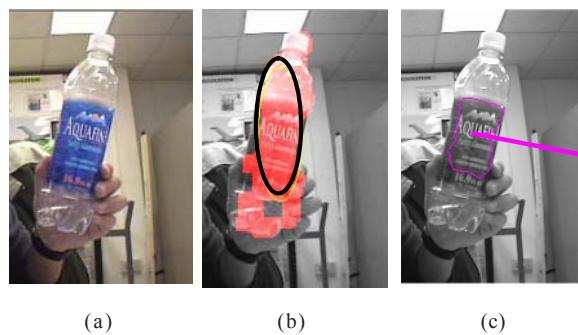
Figure 5. The overall procedure to extract an initial shape of the target s_0 *Figure 6. Example of detecting a moving object and ellipsoidal area for initial color and shape extraction: (a) original image, (b) detected ellipsoidal area, and (c) extracted initial shape and minor axis*

Figure 7. Example of initial shape extraction (Shaded area represents the result of motion-based segmentation and contour in green are of the extracted initial shape S_0)



enclosed area of the detected shape. Let's define the area of the previously extracted shape as B_p . The probability of a color pixel $\mathbf{I}_{i,j}$ in the present frame belonging to B_p can be expressed as:

$$P(\mathbf{I}_{x,y} | B_p) = C \cdot \sum_{i,j \in B_p} \delta[b(\mathbf{I}_{x,y}) - b(\mathbf{I}_{i,j})] \quad (3)$$

where C is a normalization factor equal to the size of B_p , $b(\mathbf{x})$ assigns an index number to a color vector \mathbf{x} , and δ is the Kronecker delta function. The function $b(\mathbf{x})$ is represented by:

$$b(\mathbf{x}) = \sum_{i=1}^{\dim(\mathbf{x})} \left\lfloor \frac{x_i}{2^{M-N_i}} \right\rfloor \cdot 2^{\sum_{j=i+1}^{\dim(\mathbf{x})} N_j} \quad (4)$$

where 2^N is the number of indexes for the i -th element of \mathbf{x} , which is a color component, M represents the number of bits for each color component, and $\lfloor x \rfloor$ is the floor function that gives the largest integer less than or equal to x . Once the probability $P(\mathbf{I}_{x,y} | B_p)$ is computed, the whole image is searched using the following procedure:

1. Start with the previously extracted shape;
2. Inflate the shape as long as the area enclosed by the current shape has a probability larger than a predefined threshold, or shrink the shape if the energy defined by equation (5) increases;
3. Maintain equally distributed angles from the centroid to each landmark point; and
4. Apply the smoothness constraint to neighboring points.

Step 2 can be achieved by minimizing the following energy function:

$$E = \alpha \cdot \sum_{x,y \in \Omega} (P(\mathbf{I}_{x,y} | B_p) - t_p) \quad (5)$$

where α is a negative constant, t_p is a threshold value determined empirically and Ω is the area enclosed by the new shape. The third step optimizes the smoothness measure given by:

$$E_{curv} = \sum_{k=1}^N e_{curv}^k = \sum_{k=1}^N \frac{\|\mathbf{P}_{k-1} - 2\mathbf{P}_k + \mathbf{P}_{k+1}\|^2}{\|\mathbf{P}_{k-1} - \mathbf{P}_{k+1}\|^2} \quad (6)$$

where e_{curv}^k is the curvature energy of the k -th point \mathbf{P}_k , which depicts the normalized smoothness at \mathbf{P}_k with $\mathbf{P}_0 = \mathbf{P}_N$ for $k = 1$. Finally, in addition to maintaining the condition of step 3, the energy function to be minimized is:

$$E_T = \alpha \cdot \sum_{x,y \in \Omega} (P(\mathbf{I}_{x,y} | B_p) - t_p) + \gamma \sum_{k=1}^N e_{curv}^k. \quad (7)$$

The constants α and γ are selected empirically with α being negative and equal to -0.25 and γ positive with absolute value smaller than α and equal to 0.001 in our experiments. To minimize equation (7), the following procedure is performed for each landmark point:

1. Compute the centroid of the present shape;
2. Compute the angle of the minor axis and select the first landmark point ;
3. For the k -th landmark point, compute \mathbf{u}_k and compute the total energy for the shape at point k and after moving $\pm \Delta$ pixels (usually set to $\Delta = 1$) along the line from the centroid in the direction of \mathbf{u}_k ;
4. Select the point that gives the minimum energy E_T among the three locations;
5. Repeat steps 3 and 4 for every landmark point; and
6. Stop if no change occurs, or the number of points changed is less than a threshold. Otherwise, go to step 1.

Experiments showed that by minimizing equation (7), the process acts like an accurate mean shift filter that will pull a shape closer to the actual object. Unlike the mean shift filter, the amount of displacement after each iteration is not governed by high color similarity values inside the kernel.

Once the desired number of shapes is extracted from the few first frames, the shapes are aligned with respect to the initial shape by compensating different angles and scale. An iterative algorithm for shape alignment can be achieved using the following procedure (Cootes, 1995):

1. Translate the center of gravity of each shape to the origin.
2. Select one shape \mathbf{S} as an initial estimate of the mean shape, $\bar{\mathbf{S}}_0$, and scale so that $|\bar{\mathbf{S}}_0| = 1$.
3. Record the first estimate as the current estimate of the mean shape $\bar{\mathbf{S}}_c$ to define a default reference orientation.
4. Align all the shapes with the current estimate of the mean shape $\bar{\mathbf{S}}_c$.
5. Re-estimate the mean shape $\bar{\mathbf{S}}_i$ from the aligned shapes.
6. Align the mean shape $\bar{\mathbf{S}}_i$ to $\bar{\mathbf{S}}_c$, normalize $|\bar{\mathbf{S}}_i| = 1$, and record $\bar{\mathbf{S}}_i$ as the current estimate of the mean shape $\bar{\mathbf{S}}_c$.

7. If the sum of distances between the current and previous $\bar{\mathbf{S}}_c$ changes significantly after each iteration, return to 4.

The mean shape is computed after each iteration to determine a well suited reference orientation. If the first selected estimate is not similar to the rest of shapes, the reference orientation from this shape may not be adequate to align other shapes to it. Step 4 can be achieved by minimizing the following energy:

$$E = |T(\mathbf{S}) - \bar{\mathbf{S}}_c| \quad (8)$$

where \mathbf{S} is a 2D shape from the training set, $\bar{\mathbf{S}}_c$ the current estimate of the mean shape and T is the two dimensional similarity transformation given by:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (9)$$

where (t_x, t_y) is a translation, equal to zero in our case since the centers of gravity are already aligned. The solution to equation (9) is given by:

$$a = \frac{\mathbf{S} \cdot \bar{\mathbf{S}}_c}{|\mathbf{S}|^2}, \quad b = \frac{\sum_{i=1}^N (x_i y'_i - y_i x'_i)}{|\mathbf{S}|^2} \quad (10)$$

An example of shape extraction is shown in Figure 8. The subject is detected through motion-based segmentation and the shapes of her shirt are extracted as shown. Fifty landmark points were used and every fifth landmark point is represented by a rectangle on the extracted shape. These extracted shapes are also shown in Figure 9 with examples of shape alignment. The shapes were aligned after two iterations.

Global Contour and Local Profile Modeling

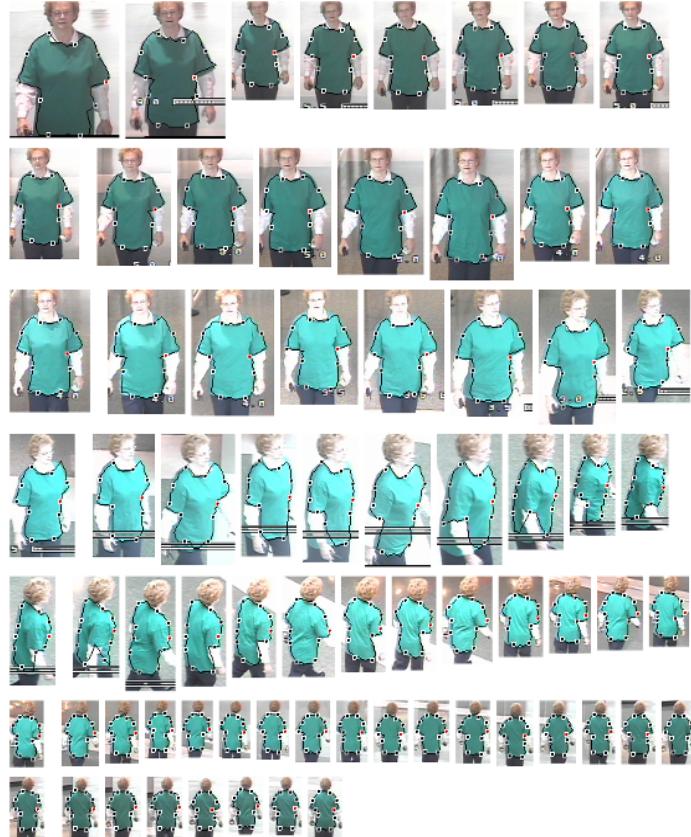
The modeling of the global contour of the shapes is performed using principal component analysis (PCA). Profile modeling extracts and stores the characteristics of neighbor pixels for each landmark point in the direction of normal vectors. Model profiles are then used to find a corresponding pixel during tracking. The overall procedure of shape modeling is shown in Figure 10.

Statistical Global Contour Modeling

While it is impossible to acquire every single shape variation of a moving target, it is possible to generate a plausible shape from a statistically built model with enough variability. PCA is used to build this statistical model from extracted contours as follows:

1. Compute the mean global shape from all extracted shapes;
2. Compute the covariance matrix of these shapes;

Figure 8. An example of online shape extraction while zooming and rotating the camera. These shapes are used for real time statistical model building.

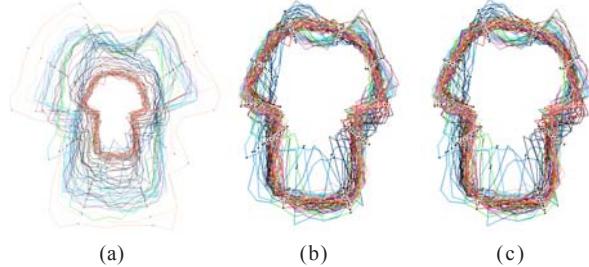


3. Compute the eigenvectors and corresponding eigenvalues of \mathbf{S} and sort eigenvectors in descending order of the corresponding eigenvalues; and
4. Choose the number of modes by selecting the smallest t that satisfies:

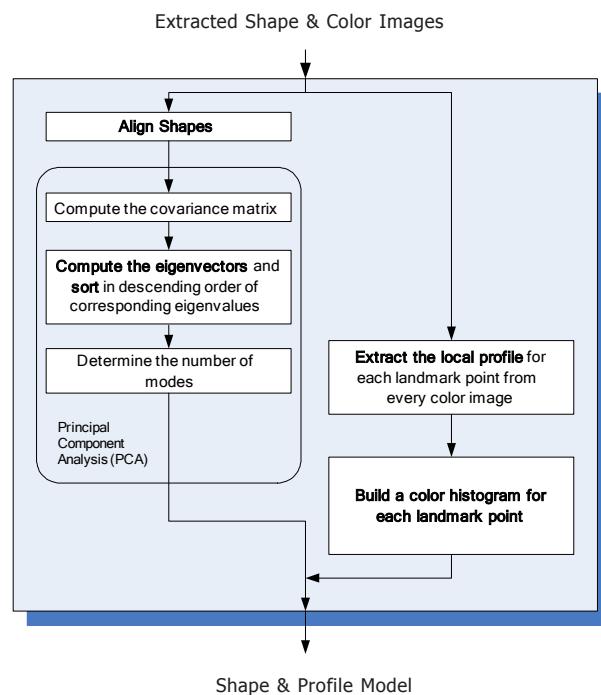
$$\sum_{i=1}^t \lambda_i \geq f_v V_T \quad (11)$$

where f_v represents the percentage of the total variation, and V_T is the sum of all eigenvalues (Cootes, 1995). For example, $f_v = 0.98$ means that this model can represent 98% of the data in the original distribution.

After the mode, or number of eigenvectors, is selected, a shape \mathbf{S} can be approximated by:

Figure 9. An example of shape alignment

(a) Extracted shapes after translating the center of each shape to the origin, (b) aligned shapes after first compensation of rotation and scaling, and (c) final shapes after second iteration. Shapes from one sequence, composed of about 480 frames, are used. Outliers in (c) represent the side views of the target as shown in Figure 8.

Figure 10. The overall procedure of global contour and local profile modeling

$$\mathbf{S} \approx \bar{\mathbf{S}} + \Phi \mathbf{b} \quad (12)$$

where $\Phi = (\phi_1 | \phi_2 | \dots | \phi_t)$ and \mathbf{b} is a t dimensional vector given by:

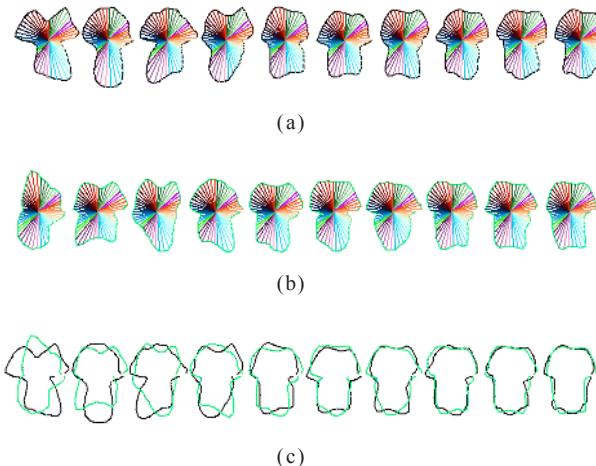
$$\mathbf{b} = \Phi^T (\mathbf{S} - \bar{\mathbf{S}}) \quad (13)$$

Equation 12 not only minimizes the dimension of the data set, but also allows the projection of any new shape onto the eigenvectors to find the nearest shape from the statistical model. In this manner, noisy shapes representing the same object from the training set can be projected onto the eigenvectors to remove the noise. In this case, the i -th parameter is usually restricted by $\pm 3\sqrt{\lambda_i}$ (Cootes, 1995). An example of shape variations is shown in Figure 11. Only one parameter of \mathbf{b} was used and the other parameters set to zero to determine the effect of each eigenvector. This is equivalent to $\mathbf{S}_i = \bar{\mathbf{S}} + \phi_i \cdot b_i$ for the i -th shape. Figure 11a shows 10 shapes using $b_i = -3\sqrt{\lambda_i}$ and Figure 11b using $b_i = 3\sqrt{\lambda_i}$. The i -th shapes for each (a) and (b) represent the variation with the eigenvector corresponding to the i -th eigenvalue. The overlapped shapes of Figures 11a and b are shown in Figure 11c. The amount of variation decreases from the left to the right since the larger eigenvalues correspond to the eigenvectors that cause the larger variations. In actual shape tracking, a combination of these shape variations is used to reproduce a shape likely to be found in the training set.

Adjustment of Number and Location of Landmark Points

In the original ASM, the number of landmark points is determined by the user. In this work, this number is selected automatically for the tracking of any unknown object.

Figure 11. Examples of shape variations with $\mathbf{S}_i = \bar{\mathbf{S}} \pm \phi_i \cdot 3\sqrt{\lambda_i}$, for the i -th shape: (a) shape variations for the i -th shape by $\mathbf{S}_i = \bar{\mathbf{S}} - \phi_i \cdot 3\sqrt{\lambda_i}$ for ; (b) shape variations for the i -th shape by $\mathbf{S}_i = \bar{\mathbf{S}} + \phi_i \cdot 3\sqrt{\lambda_i}$ for $i = 1, \dots, 10$; (c) overlapped shapes from (a) and (b)



Landmark points were placed on T-junctions and corners, points in-between are placed at equally distributed lengths and training images were selected to include many variations for a good representative statistical model. Automating these tasks requires a measure to identify T-junctions and corners from segmented shapes and a similarity measure to select different images that have different deformations. T-junctions and corners are found by computing angles between tangents to initial landmark points. Since angles are equally distributed, the angle at P_k can be computed by:

$$\alpha_k = \cos^{-1} \left(\frac{\overline{P'_{k-1} P_k} \cdot \overline{OP_k}}{\|\overline{P'_{k-1} P_k}\| \|\overline{OP_k}\|} \right) + \cos^{-1} \left(\frac{\overline{P'_{k+1} P_k} \cdot \overline{OP_k}}{\|\overline{P'_{k+1} P_k}\| \|\overline{OP_k}\|} \right) \quad (14)$$

where $\overline{P'_{k-1}}$ and $\overline{P'_{k+1}}$ represent mean values of neighbor points. For example, $\overline{P'_{k-1}}$ is the mean of three points P_{k-1} , P_{k-2} , and P_{k-3} . This averaging is necessary to reduce noise during shape extraction. An example of computed angles between landmark points is shown in Figure 12. Shapes from Figure 8 are used and the angles from each shape are shown with a different color line. The mean of these angles is shown in Figure 12 with a thick black line located around 150° . The line on the bottom, below 50° in the Y axis, represents selected landmark points. After the mean angles have been computed -black line-, minimum and maximum points are found where the sign of the first derivative changes and discarded if minimum or maximum values are around 180° , which represents flat lines. A value of 50 is assigned to selected minimum and maximum points and 25 assigned to neighbor points on the line on the bottom. Since these selected points have bigger or smaller angles than their neighboring points, it is highly probable that these points represent corners or T-junctions where landmark points need to be placed. By selecting points that have non-zero values in the line below 50, the number of landmark points is reduced. An example of extracted shapes after the number of landmark points is reduced is shown in Figure 13. Red lines are used to connect selected landmark points on corners. Out of the original 50 landmark points, 23 are kept as shown in Figure 13.

Figure 12. An example of angle estimation of shapes from Figure 8. The darkest line is the mean of angles from the training set, and the line on the bottom, below 50° in the Y axis, represents selected points of non-zero value for each landmark point.

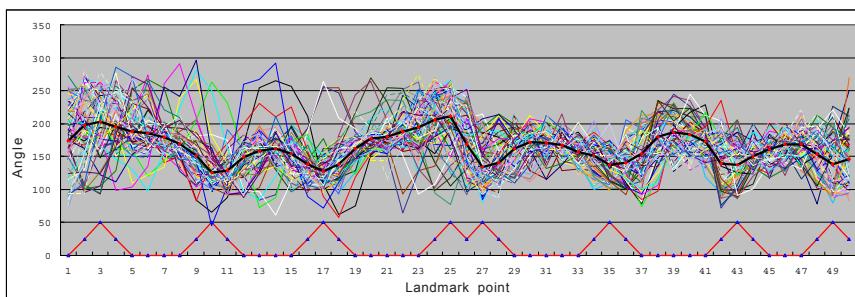


Figure 13. An example of shape extraction after reduction of the number of landmark points. Lines are used to connect selected landmark points. Points on straight edges are discarded.



Profile Modeling

Landmark points belong to the target boundary which separates the object from the background. Although boundaries are usually strong edges, other strong edges in the background or inside the object may interfere with the localization of the actual shape's boundary. Therefore, the modeling of each landmark point is also required to discriminate between edges. First, the profile modeling used in the original ASM (Cootes, 1995) is summarized and then a more suitable profile modeling approach to PTZ tracking in cluttered and varying environment is described.

The profile \mathbf{g}_i of the i -th landmark point is defined as a group of pixel values on the normal vector to the contour at the i -th landmark point with 1 pixel spacing. The original ASM uses a normalized derivative profile \mathbf{g}'_i for the i -th landmark point and the j -th element of \mathbf{g}'_i ; g'_{ij} , is computed as follows:

$$g'_{ij} = \frac{1}{\sum_j |g_{ij}|} (g_{ij} - g_{j+1}) \quad (15)$$

where g_{ij} represents the j -th point along the i -th profile. The direction of each profile can be computed either by using neighboring points or the unit vector used to extract the points. These profiles are then used to compute the similarity between a profile at a current search point and profiles obtained from this modeling process, using the Mahalanobis distance:

$$f(\mathbf{g}'_s) = (\mathbf{g}'_s - \bar{\mathbf{g}}'_i)^T \mathbf{S}_g^{-1} (\mathbf{g}'_s - \bar{\mathbf{g}}'_i) \quad (16)$$

where $\bar{\mathbf{g}}'_i$ is the mean of derivative profiles for the i -th landmark point, \mathbf{S}_g the covariance matrix of i -th profiles and \mathbf{g}'_s the derivative of a sample profile at a current search point. Illumination changes often affect pixel values by either adding or subtracting an offset value when it becomes bright or dark. Using the derivative minimizes this effect. This modeling usually works well for medical imaging since these images usually involve clear differences in intensity values between objects of interest and surrounding pixels for both the training set and test images. This, however, produces a problem in visual tracking, especially tracking of moving objects using a PTZ camera. Unless the surround-

ing has a single color, which is very unlikely, the background around an object being tracked will always be changing. The original profile modeling in ASM would include pixels from a varying background, which would produce errors. An alternative to building profile models is presented with two major differences to the original ASM profile model: (1) The new profile includes only pixels inside the target, and (2) a new measure to find a corresponding point during tracking is used. Instead of computing derivatives of profiles, a color histogram of each profile is built to produce a color similarity image. For example, with 50 landmark points on each image and 40 training images, there will be 50 profiles with each profile model including the profile data from the 40 images. The similarity between the current input $\mathbf{I}_{x,y}$ and the i -th profile from n training images is computed as:

$$P(\mathbf{I}_{x,y} | (\mathbf{q}_i^1, \mathbf{q}_i^2, \dots, \mathbf{q}_i^n)) = C \cdot \sum_{z=1}^n \sum_j \delta[b(\mathbf{I}_{x,y}) - b(q_{ij}^z)] \quad (17)$$

where z represents the z -th training image, and C is a normalization factor, $C = nxN_p$, where N_p is the length of a profile, \mathbf{q}_i^z is the i -th extracted profile, a set of color vectors q_{ij}^z from point \mathbf{P}_k to the center of gravity \mathbf{O} , from the z -th training image and $b(x)$ is as described in Equation 4. An example of \mathbf{q}_i^z is shown in Figure 14 as black lines inside the shirt.

In summary, global shape modeling using PCA to describe shape variations is first performed; profile modeling is then conducted to characterize each landmark point on the shape contour. During tracking, each shape parameter is updated to generate the shape closest to the current frame, and landmark search is performed until no major changes occur on the shape parameter \mathbf{b} . The details of shape-based tracking will be presented in the following section.

Figure 14. Example of profile extraction. Black lines represent the line where pixels are extracted to build a color histogram of the profile.



Tracking with Shapes

The shape model derived in the previous subsection contains local structures for each landmark point and global shape parameters to generate any variations on the contour. The global shape parameters are used to find a plausible shape from the training set and are updated at each iteration after a best match is found for each landmark point. The mean shape from the training set is usually used as the initial shape placed on the target.

Once an initial shape is placed on the target, a search is conducted to find the best match for each profile at every landmark point by minimizing a new objective function as described by Equation 18. A contour composed of the best matching points is then used to find the pose parameters of the shape, since the statistical shape model did not account for pose variations, such as rotation, translation, and scaling. After pose variations are corrected, the contour is projected onto the eigenvectors found during the global modeling phase to extract its parameter. This procedure ensures that the final shape is always a component from the statistical shape model with constraints for \mathbf{b} in the limits of $\pm 3\sqrt{\lambda_i}$.

Profile Search

For each point of the initial shape, the search area is restricted to a 1D line along the normal vector. Since each profile only has pixels inside the object and the Gaussian assumption does not hold anymore, the Mahalanobis distance used with the original ASM is no longer valid. The new objective function should find the boundary by moving a current search point in the direction of each unit vector \mathbf{u}_k , from the center of gravity to the k -th landmark point \mathbf{P}_k . The objective function to be minimized for each landmark point is given by:

$$E_k = \frac{e_i \cdot e_o}{|e_i \cdot e_o| + 1} + \frac{e_i}{|e_i| + 1} |\mathbf{OP}_k| \quad (18)$$

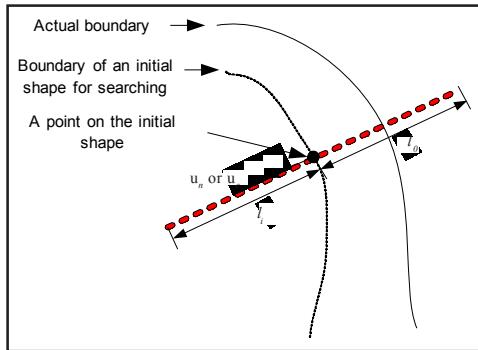
where $|\mathbf{OP}_k|$ is the distance between the current center of gravity and the current point \mathbf{P}_k and e_i is formulated as:

$$e_i = \sum_{x,y \in l_i} (t_i - P(\mathbf{I}_{x,y} | (\mathbf{q}_i^1, \mathbf{q}_i^2, \dots, \mathbf{q}_i^n))) \quad (19)$$

where l_i is a set of k pixels from \mathbf{P}_k to the center of gravity \mathbf{O} with a spacing of 1 and t_i is a threshold value; l_o represents pixels outside of the boundary in the direction of the i -th unit vector. An example of l_i and l_o is shown in Figure 15.

The first term of equation (18) will be minimized when e_i and e_o have different signs. If the similarity P computed on l_i in equation (19) is larger than the threshold value t_i , then e_i will be negative. Assume that l_i and l_o appear inside of the object, then the first term of equation (18) will be positive and e_i is negative. The second term is then minimized if a point moves to the boundary while increasing $|\mathbf{OP}_k|$. If l_i and l_o appear outside of the boundary, the sign of e_i will be positive and the energy will be minimized if $|\mathbf{OP}_k|$ decreases,

Figure 15. An example of profile search for landmark point positioning



which pulls a point toward the center of gravity. If l_i and l_o are placed on opposite sides of the actual boundary, the first term in equation (18) becomes negative, and the second term in equation (18) will push or pull a current point \mathbf{P}_k to the boundary based on the sign of e_i . This may cause oscillating of a current point \mathbf{P}_k around the actual boundary. This, however, will not affect the performance of the algorithm since each new shape will be projected onto the eigenvectors after finding the corresponding points. Small fluctuations will be considered as noise and will only change the shape parameters by very small amounts. In summary, the following procedure is used to find a location that minimizes equation (18):

1. Compute the center of gravity \mathbf{O} of the current shape
2. Compute the k -th unit vector $\mathbf{u}_k = \frac{\overrightarrow{\mathbf{OP}_k}}{|\overrightarrow{\mathbf{OP}}|}$ for a given point
3. For each point \mathbf{P}_k , three E_k values are computed.
 - a. Compute E_k at \mathbf{P}_k .
 - b. Move \mathbf{P}_k 1 pixel toward the center of gravity along \mathbf{u}_k and compute E_k .
 - c. Move \mathbf{P}_k 1 pixel in the opposite direction and compute E_k .
 - d. Find the location where the smallest E_k appears.
4. Update the shape parameters.

With the center of a current shape placed anywhere inside the object, this objective function will still move each landmark point toward the boundary.

Updating Global Contour Parameters

In the previous section, we described how to find matching points for each landmark point by minimizing the objective function of equation (18). Let's define the contour that consists of all best matching points as \mathbf{Y} . When the statistical shape model was built, the shapes were first aligned before applying the PCA. This procedure removes any shape

variation caused by a difference in pose, such as rotation and scaling. This aligning step is necessary because shapes not only show variations caused by object deformations, but also variations caused by camera motion, rotation and scaling, even if the contour does not deform. For this reason, pose variations on \mathbf{Y} are first removed by minimizing the following equation (Cootes, 1995):

$$|\mathbf{Y} - T_{X_t, Y_t, s, \theta}(\bar{\mathbf{S}} + \Phi \mathbf{b})| \quad (20)$$

where $T_{X_t, Y_t, s, \theta}$ represents translation by (X_t, Y_t) , scaling by s , and rotation by θ . A simple iterative approach to minimize Equation 20 is as follows:

1. Initialize the shape parameters, \mathbf{b} , to zero.
2. Generate the model instance $\mathbf{S} = \bar{\mathbf{S}} + \Phi \mathbf{b}$.
3. Find the pose parameters (X_t, Y_t, s, q) which best map \mathbf{Y} to \mathbf{S} by minimizing $E = |T(\mathbf{Y}) - \mathbf{S}|$.
4. Apply the pose parameters to \mathbf{Y} in the model coordinate frame: $\mathbf{y} = T_{X_t, Y_t, s, q}(\mathbf{Y})$.
5. Update the model parameter by $\mathbf{b} = \Phi^T(\mathbf{y} - \bar{\mathbf{S}})$.
6. Limit the variation of \mathbf{b} by $\pm 3\sqrt{\lambda_i}$.
7. If the Euclidean distance between the current and the previous \mathbf{b} is larger than the threshold, return to step 2.

This procedure is iterated until no major changes appear on \mathbf{b} . The reconstructed shape can be computed as:

$$\mathbf{Y}_{new} = T^{-1}_{X_t, Y_t, s, \theta}(\bar{\mathbf{S}} + \Phi \mathbf{b}) \quad (21)$$

This shape is then used as the initial shape to search the profiles for the next iteration. The overall procedure for shape-based tracking is summarized in Figure 16.

Experimental Results Using the Shape Tracker

Five hundred training frames were acquired from a single PTZ tracking sequence. Eleven of these frames were front, side, and back views of the target and were used as input images for shape-based tracking while the remaining 488 frames were used to build the statistical shape model. The mean shape is used as an initial shape for each experiment. Figure 17 shows the first within frame experimental results on a front image. A total of 100 iterations are performed when the initial shape is placed to cover only half of the expected shape. This result confirms that if an initial shape is not well placed it can still be used with our new objective function while delivering good results. The second experimental results with a side view of the target are shown in Figure 18. This result also provides as good performance as the first experiment. Tracking on consecutive frames was also conducted based on shape, where the result from one frame was used as an initial shape to the following frame. The results of the between frame tracking are shown in Figure 19.

Figure 16. The overall procedure of shape tracking

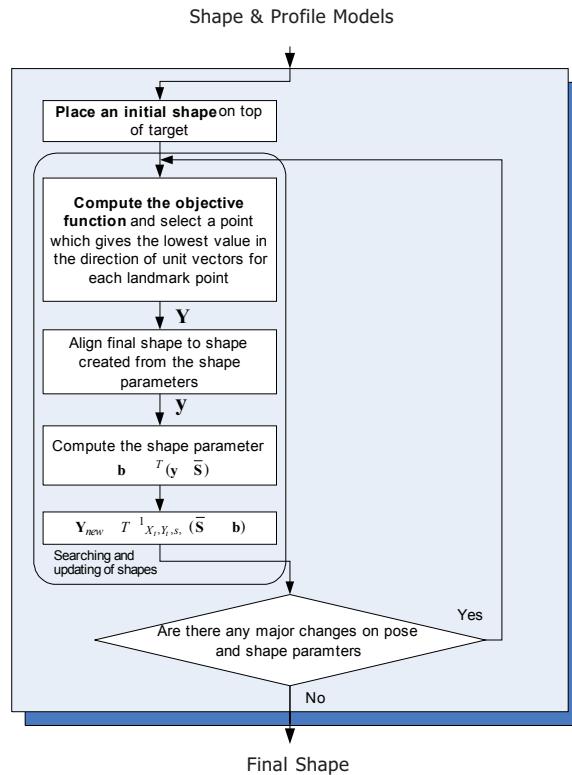


Figure 17. Within-frame shape fitting on frontal view (a total of 100 iterations are performed and every 10th result is displayed)

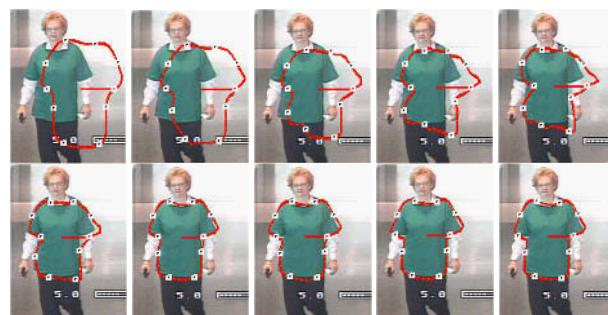


Figure 18. Within-frame shape fitting on side view (a total of 100 iterations are performed and every 10th result is displayed)

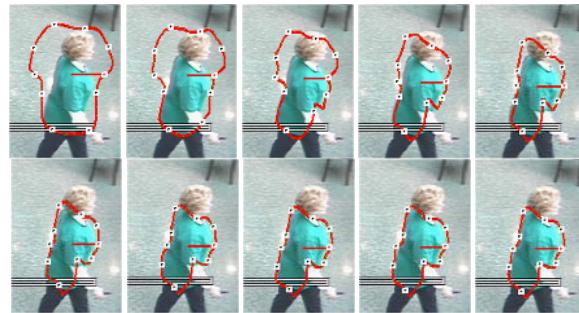


Figure 19. Between-frames shape-based tracking from the 280th to 288th frame



FUTURE TRENDS AND CONCLUSIONS

This chapter described an automatic tracking system utilizing color and shape for surveillance purposes. For initial target acquisition, either a static camera or a non-moving PTZ camera can be used. Once the target is detected, the PTZ camera extracts color features and the camera is controlled based on the location of the target. While the PTZ camera is initially tracking the object, the proposed algorithm extracts shapes to learn deformations of the target. Once the number of shapes is sufficient for a statistical model building, a shape model is computed by performing a global contour modeling and a local profile modeling. Based on a newly introduced cost function, a shape-based tracking is then used to track the target with minimal assistance from the color tracker.

Color and shape-based trackers in general can each work adequately under certain conditions. The color tracker performs well when illumination is relatively constant and

no similar colors appear around the target. The shape tracker yields good results when the object shows predictable shape variations with manageable occlusion. But using color and shape will always increase the performance, especially in hard to handle situations. Any successful tracking system must rely on more than one type of features such as an adaptive color and shape tracking scheme based on performance measures. It should be noted that color alone can not be easily updated since the extent of the color area can not usually be detected with color and mean shift filter-based methods alone. By combining color and shape, the entire color area can be computed based on the shape-based tracking allowing for the updating of the color model. Color also assists in locating the initial shape, which was an unrealistic assumption of the original ASM that requires the initial shape to be manually placed around the target. Future sample methodology for fusing color and shape is summarized in the following subsections.

Tracking with Color and Shape

Different situations usually require different strategies for selecting features. For example, in the presence of illumination changes, applying the mean shift filter for few frames before the shape-based tracking is initiated is not a good idea since the probability image based on the color of the extracted shapes may not represent the actual object. It is also possible that local profile models suffer from illumination changes. Local profile models, however, are less affected by illumination changes than a single color model representing the whole object. A single global color model may ignore small areas with slightly different colors from the rest of the object, but a local profile model of that small area will consider the slightly different color as a representative color. Since illumination changes do not always affect the whole object at the same time, parts of local profile models may still have valid colors to represent each local profile. If similarity and validity of both color and shape are able to detect illumination changes, the threshold value in the objective function to locate corresponding points can be changed to deal with illumination changes. It is also possible to change the local profile models using a different formulation, such as extracting the hue component from the local profile models, which is robust to illumination changes, or prepare a second local profile model during the training stage. One possible local profile model is the histogram of derivative on profiles. Occlusion and unexpected deformations can be detected by computing the validity of the shape recovered and color-based tracker will have a higher weight for the final decision in this case.

Updating Shape and Color Models

Similarity measures for both color and shape need to be designed and combined to assist each other for computing an overall confidence measure. Color similarity between two consecutive frames, shape similarity to the training, and shape similarity between two consecutive frames need all to be combined in an overall confidence measure to assess the quality of tracking and be used to update and adjust the color and shape models.

Some deformations of the target may not be available in the training set, but if the similarity between a previous and a current frame, with unexpected deformation, is very high, a new shape can be extracted using color from the previous frame and used to update the shape model. Color can be updated in the same way after high similarity of shapes from consecutive frames is found. This will adaptively adjust the single color model and

the local profile models. We may assume that at least one of the two methods has a high performance for each frame, and validate this assumption in the next frame. If this assumption is valid, we can update the other feature. Additional developments of this approach will consist of:

- Stopping criterion for the new objective function
- Study on threshold values for extracting and tracking a shape
- Color and shape similarity between consecutive frames
- Color and shape similarity to the training set
- Overall performance measure
- Decision rule for fusing shape and color

In summary, future work in the area of segmentation of moving and deformable objects in changing backgrounds should rely on more than one type of features. As noted earlier, the use of color alone is not sufficient as this mode fails miserably under changing illumination conditions. Active Shape Models are a good means of tracking deformable shapes in an off-line mode with most of the literature in this area relying on manual and lengthy landmark point selection phases. Extending this work to specifically incorporate color to assist in the automatic initialization of shapes and to possibly add texture features represents in our opinion the future of any automatic real time tracking concept.

REFERENCES

- Amit, Y. (2002). *2D object detection and recognition*. Cambridge, ME: MIT Press.
- Baumberg, A. M. (1995) *Learning deformable models for tracking human motion*. Doctoral dissertation, The University of Leeds, UK.
- Bruijne, M. D., Ginneken, B. V., Viergever, M. A., & Niessen, W. J. (2003, July). Adapting active shape models for 3D segmentation of tubular structures in medical images. In *Proceedings of the 18th International Conference on Information Processing in Medical Imaging*, Ambleside, UK (Vol. 2732, pp. 136-147).
- Caselles, V., Kimmel, R. & Sapiro, G. (1995, June). Geodesic Active Contours. In *Proceedings of the International Conference on Computer Vision*, Boston (pp. 694-699).
- Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., et al. (2000). *A system for video surveillance and monitoring* (Tech. Rep. No. CMU-RI-TR-00-12). Robotics Institute, Carnegie Mellon University, USA.
- Comaniciu, D., Ramesh, V., & Meer, P. (2003, May). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 564-577.
- Cootes, T. F., Cooper, D., Taylor, C. J., & Graham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1), 38-59.
- Cremers, D., Kohlberger, T., & Schnörr, C. (2003). Shape statistics in kernel space for variational images segmentation. *Pattern Recognition*, 36, 1929-1943.
- Cremers, D., Tischhäuser, F., Weickert, J., & Schnörr, C. (2002). Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50, 295-313.

- Davatzikos, C., Tao, X., & Shen, D. (2003, March). Hierarchical active shape models using the wavelet transform. *IEEE Transactions on Medical Imaging*, 22(3), 414-422.
- Debreuve, É., Barlaud, M., Aubert, G., Laurette, I., & Darcourt, J. (2001). Space-time segmentation using level set active contours applied to myocardial gated SPECT. *IEEE Transactions on Medical Imaging*, 20(7), 643-659.
- Dellaert, F., & Collins, R. (1999, September). Fast image-based tracking by selective pixel integration. In *Proceedings of the ICCV 99 Workshop on Frame-Rate Vision*, Kerkrya, Greece.
- Ginneken, B. V., Frangi, A. F., Staal, J. J., Romeny, B. M. T. H., & Viergever, M. A. (2002, August). Active shape model segmentation with optical features. *IEEE Transactions on Medical Imaging*, 21(7), 924-933.
- Han, C., Hatsukami, T. S., Hwang, J., & Yuan, C. (2001). A fast minimal path active contour model. *IEEE Transactions on Image Processing*, 10(6), 865-873.
- Haritaoglu, I., Harwood, D., & Davis, L. S. (2000, August). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 809-830.
- Kang, S., Paik, J., Koschan, A., Abidi, B., & Abidi, M. A. (2003, May). Real-time video tracking using PTZ cameras. In *Proceedings of the SPIE 6th International Conference on Quality Control by Artificial Vision*, Gatlinburg, TN (Vol. 5132, pp. 103-111).
- Kang, S., Koschan, A., Abidi, B., & Abidi, M. (2004, March). Video Surveillance of High Security Facilities. In *Proceedings of the 10th International Conference on Robotics & Remote Systems for Hazardous Environments*, Gainesville, FL (pp. 530-536).
- Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snake: Active contour model. *International Journal on Computer Vision*, 1, 321-331.
- Medioni, G., Cohen, I., Brémond, F., Hongeng, S., & Nevatia, R. (2001, August). Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), 873-889.
- Nguyen, H. T., Worring, M., Boomgaard, R. V. D., and Smeulders, A. W. M. (2002, September). Tracking nonparameterized object contours in video. *IEEE Transactions on Image Processing*, 11(9), 1081-1091.
- Osher, S., & Sethian, J. A. (1988). Front propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79, 12-49.
- Ren, Y., Chua, C., & Ho, Y. (2003). Motion detection with nonstationary background. *Machine Vision and Applications*, 13, 332-343.
- Shen, D., & Davatzikos, C. (2002, January). An adaptive-focus deformable model using statistical and geometric information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 906-913.
- Wang, Y., & Staib, L. H. (2000). Boundary finding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), 738-743.
- Xu, C., & Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3), 359-369.
- Zhao, T., & Nevatia, R. (2004, July). Tracking multiple humans in crowded environment. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 406-413).

Chapter IX

Video Shot Boundary Detection and Scene Segmentation

Hong Lu, Fudan University, China

Zhenyan Li, Nanyang Technological University, Singapore

Yap-Peng Tan, Nanyang Technological University, Singapore

Xiangyang Xue, Fudan University, China

ABSTRACT

This chapter presents a new and efficient method for shot boundary detection (SBD) and scene segmentation. Commonly the first step in content-based video analysis, SBD partitions video data into fundamental units of shots. Over the past decade, SBD has attracted a considerable amount of research attention. However, the detection accuracy achieved leaves much to be desired. In this chapter, a new SBD method based on sequential change detection is proposed to achieve improved detection accuracy. The method is then extended to segment videos into scenes. Compared with existing scene segmentation methods, the proposed method can also obtain more accurate results over a large set of test videos.

INTRODUCTION

The advances in computing technologies, broadband communication networks, mass storage devices and consumer electronics have resulted in large amounts of video data being generated and made accessible in digital form throughout the world. As a consequence, a large variety of video-based applications, such as video on demand,

interactive TV, digital library, online distance learning, remote video surveillance, video conferencing, and so forth, have attracted much interest and attention. However, as the contents of most videos are either lacking in structure (e.g., it is hard to predict the visual content of subsequent video scenes before seeing it) or without detailed annotation, one needs to sequentially browse through a video in order to grasp its content. Even though there are ways, such as fast-forward and fast-backward functions, that allow users to speed up the browsing process, it is still rather time-consuming and requires close examination from the users. Therefore, efficient ways to analyze, annotate, browse, manipulate and retrieve videos of interest based on their contents are becoming increasingly important and have attracted substantial research attention over the past decade. This is evident by the large amount of research activities reported in the literatures and the emergence of content description standards, such as MPEG-7, a related standardization effort of the Moving Picture Experts Group (MPEG) (Pereira, 1999; Nack, 1999a, 1999b).

In order to search and retrieve video data in an automated fashion, video contents need to be annotated and indexed in formats usable by a search engine. Today, most of these data do not carry much descriptive information with regard to their semantic contents. In particular, annotations for audiovisual data are mainly restricted to textual descriptions, such as title (e.g., "Gone with the Wind"), genre (e.g., drama, civil war), summary or abstract (e.g., movie abstract or review) and names of the subjects (e.g., Clark Gable & Vivien Leigh), etc. Conceivably, these textual descriptions can hardly be exhaustive or comprehensive in depicting the audiovisual data. Moreover, as generally agreed-upon vocabularies and syntax are not yet in place for the description of video data, these textual descriptions could be rather subjective and may not generally agree with each other. More objective descriptions and representations of video data, especially those that can be automatically and unambiguously extracted from audiovisual content, are therefore desirable and can complement the capability of textual descriptions.

In general, there are three main approaches to content-based video description or indexing: text-based, feature-based and semantic-based. In a text-based approach, keywords or phrases are used to describe video contents in various possible ways. However, these textual descriptions are incapable of detailing the rich information embedded in audiovisual data. In the feature-based approach, low-level audiovisual features such as color, texture, shape, motion and audio attributes are extracted from the data and used as indexing keys. The underlying rationale for this approach is that video data that are similar in terms of their audiovisual features are likely to be similar in content (Smoliar, 1994; Wang, 2000).

In a semantic-based approach, video data are annotated with their high-level semantic meanings. Substantial recent research efforts have focused on extracting these high-level semantics with automated or computer-assisted methods. Given the vast varieties of video contents and the difficulty of the problem itself, this approach is currently the most challenging and often requires some degree of human intervention. However, by restricting the analysis to a specific class of video, useful high-level semantics can be derived based on some primitive audiovisual features and domain-specific knowledge. Examples are news items in news programs (Ide, 2000), play events of sports games (Miyamori, 2000; Xu, 2001; Zhong & Chang, 2001) and rating of motion pictures (Nam, 1998).

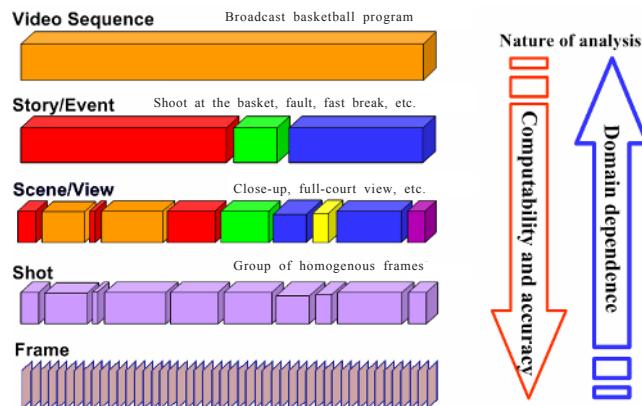
The main objective of our current research work is to develop new video analysis methods to narrow the gap between low level visual features and high level video semantics. Specifically, we are interested in efficient and automated techniques that can parse and group video shots into semantic units, providing a more compact and accurate abstraction of video contents useful for video analysis and retrieval.

BACKGROUND AND RELATED WORK

In content-based video analysis, video can be analyzed in five structured levels as shown in Figure 1. For the nature of the analysis involved in these five structured levels, the domain dependency tends to increase and the computability/analysis accuracy tends to decrease toward the higher structured levels. Specifically, in frame level analysis, low-level features such as color, texture, shape, motion and audio are generally used and the analysis requires no or minimum domain knowledge. At this level, many shot boundary detection (SBD) methods have been proposed to segment video into shots, each of which can then be represented by one or a few key frames from the shot.

As human users are more interested in the semantic levels concerning the underlying scenes, stories, events or plots of a video program, higher levels of analysis are generally required to analyze video shots for more compact or meaningful representations. The analysis includes, for example, scene clustering (clustering visually similar shots into scenes) and scene segmentation (grouping related shots into scenes, each featuring a dramatic event). Based on the detected shots and the clustered or segmented scenes, one or more key frames can be extracted. Afterwards, image features such as color, shape and texture are used to index these key frames. In addition, high-level representations such as regions, objects, and motion can also be used to infer semantic events of interest and help summarize the content of the whole video sequence. Finally, videos can be browsed and retrieved based on the similarity between the features or events of the query video sequence and the video sequences in the database.

Figure 1. Five structured video levels for content-based video analysis



In this chapter, we review some state-of-the-art research works for SBD and scene segmentation, and propose new techniques to address these important problems in analyzing videos based on their contents.

Shot Boundary Detection

Shot boundary detection (SBD) is commonly the first step in the process of indexing, characterizing and retrieving of video. As shown in Figure 2, a shot is a temporal sequence of frames generated and recorded continuously by a single camera act, which usually depicts a continuous action without significant content changes.

To form a video sequence, shots are joined together during video sorting and post editing with either abrupt cuts or gradual visual effects according to the nature of the scene changes or story sequences. As shown in Figure 3, there are generally two types of shot transitions: abrupt shot boundary (ASB), also known as shot cut or hard cut, where the change of video content occurs over a single frame; and gradual shot boundary (GSB), such as fade in, fade out, dissolve and wipe, where the change takes place gradually over a short sequence of frames.

Figure 2. Shot boundary detection and shot representation

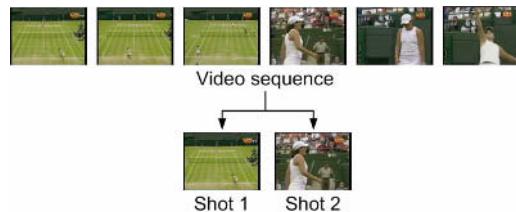
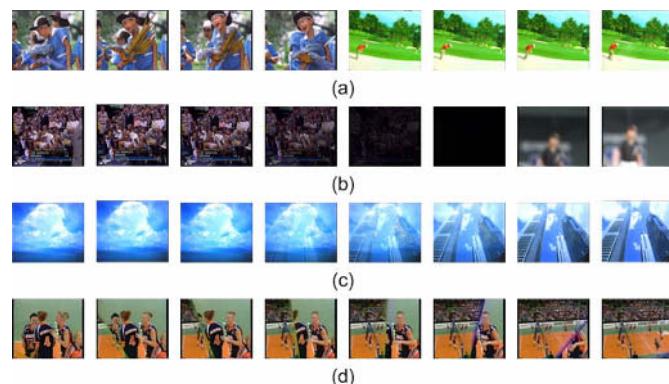


Figure 3. Different types of shot boundaries: (a) a shot cut, (b) a fade out followed by a fade in, (c) a dissolve, and (d) a wipe



The main technique of SBD is to detect the change between each consecutive frame pair. If the change exceeds a predetermined threshold, a shot boundary is assumed. In the literature, the change is usually measured by using such low-level video features as pixel intensities, color histograms, edges and compressed video attributes (Koprinska, 2001; Gargi, 2000; Brown, 2000; Zhong, 2001; Wu, 2003; Quenot, 2003; Zhai, 2003; Furht, 1995). For example, in Furht (1995), Zhang et al. propose a pair-wise comparison between each pixel in one frame with the corresponding pixel in the next frame. This method has been known to be sensitive to camera and object motions, and may cause excessive false detections. Color histogram is another type of feature used in SBD which is known to be less sensitive to small changes over consecutive frames (Gargi, 2000; Brown, 2000; Furht, 1995). Edge differences have also been used to counter problems caused by fades, dissolves, etc. (Brown, 2000). In compressed domains, much SBD work has been done by using Discrete Cosine Transform coefficients, motion vectors and macroblock types of MPEG video (Koprinska, 2001; Gargi, 2000; Brown, 2000).

Despite many visual features that have been proposed for SBD, color histogram remains the most popular because it is easy to compute, insensitive to small local object motion or partial occlusion and capable of achieving good detection performance (Gargi, 2000; Brown, 2000). In the color histogram-based SBD method, the two main issues of concern are color space selection and quantization. Much work has been done on the comparison of using different color spaces or quantization schemes in difference metrics for SBD (Gargi, 2000). Experimental results show that more perceptually uniform color spaces, such as Munsell, CIE L*a*b*, L*u*v* and HSV generally perform better than RGB when the same number of quantization bins are used (Gargi, 2000).

Another problem of concern in SBD is related to the selection of an appropriate threshold for identifying whether or not a change is large enough to signify a shot boundary. In O'Toole (1999), a global threshold is used to detect shot boundaries in different types of videos. The experimental results show that the detection performance can vary by up to 20% even for the same type of video content. To overcome this large performance variation, some later work proposes to use an adaptive threshold that is set according to the video content (Gargi, 2000; Furht, 1995; O'Toole, 1999; Yu, 2001). For example, Zhang et al., select the threshold as $T_b = \mu + \alpha\sigma$, where μ and σ are the mean and standard deviation of histogram differences between consecutive frames and α is suggested to be between 5 and 6 according to their empirical study (Furht, 1995). In Yu (2001), an entropic thresholding method is proposed to determine a threshold of inter-frame histogram difference (IFHD) for detecting ASBs.

For GSBs, detection of changes becomes more challenging, as the IFHDs within a gradual transition could be small and hence difficult to detect. To overcome this difficulty, a twin-comparison method is proposed in Furht (1995) to evaluate the histogram difference within a candidate gradual transition by using two thresholds—a high threshold T_h and a low threshold T_l . It has been suggested that T_h be set as threshold T_b discussed above for detecting ASBs (Furht, 1995), and T_l be set equal to $b \times T_h$, where b is selected from the range 0.1–0.5 (Zhang, 1997).

In Lelescu (2003), a shot boundary is modeled as an additive change in a vector parameter θ of the probability density function (pdf) associated with a video feature vector sequences. The feature vectors are assumed to form an independent and identically distributed (i.i.d.). By assuming that the d -dimensional feature vectors X_k follow Gaussian distribution, the parameter θ is set to $\theta = \mu$, the mean of the pdf:

$$p_{\mu, \Sigma}(X_k) = \frac{1}{\sqrt{(2\pi)^d (\det \Sigma)}} e^{-\frac{1}{2}(X_k - \mu)^T \Sigma^{-1} (X_k - \mu)}$$

Furthermore, $\theta = \theta_1$ before a shot boundary, and $\theta = \theta_2$ after the shot boundary. As the shot boundaries are detected sequentially, in general minimal or no information is available about parameter $\theta = \theta_2$ after a shot boundary. Consequently, the work in Lelescu (2003) formulates SBD as testing between two hypotheses H_1 and H_2 :

$$\begin{aligned} H_1 &= \{\theta : \|\theta - \theta_1\|_\Sigma^2 \leq a^2, k < t_0\} \\ H_2 &= \{\theta : \|\theta - \theta_1\|_\Sigma^2 \geq b^2, k \geq t_0\} \end{aligned} \quad (1)$$

where $\|\theta - \theta_1\|_\Sigma^2 = (\theta - \theta_1)^T \Sigma^{-1} (\theta - \theta_1)$, t_0 is the true change time, a and b are two thresholds with $a < b$. The formulation assumes that there is a known upper bound for θ_1 and a known lower bound for θ_2 . Consequently, the hypothesis testing is solved by using log-likelihood ratio test.

Of interest is the case where θ_1 is assumed to be known and θ_2 completely unknown. This appears to be a limiting case for the solution to the hypothesis testing above. The generalized likelihood ratio (GLR) algorithm (Lelescu, 2003) provides a possible solution by replacing the unknown parameters (i.e., θ_2) with their maximum-likelihood (ML) estimates.

Scene Segmentation

After segmenting a video into shots, related shots can be further grouped into scenes by incorporating some temporal constraint or correlation.

In Yeung (1996), time constrained clustering of video shots is performed, that is, two shots are considered similar if their content similarity is high enough and the difference of their occurrences in time is small. After clustering, shots belonging to the same cluster can be represented by the same label, such as cluster label A , B or C , and so forth. To detect scene boundaries, a scene transition graph based on the scene clustering results is proposed in Yeung (1996), where the scene segmentation problem is considered as segmentation of cluster label patterns. For example, if the cluster labels of a sequence of shots are $ABABCDCD$, then the corresponding graph for the temporal transition is:

$$A-B \Rightarrow C-D$$

However, this method has the limitation that it does not consider the shot length and depends upon some scene clustering threshold parameters which need to be manually tuned.

In Rui (1999), a more general time-adaptive grouping approach based on visual similarity and temporal locality of shots is proposed to cluster them into groups. Semantically related groups are then merged into scenes. The method relies on pre-defined thresholds for shot clustering and scene merging. To overcome these limitations, Kender and Yeo propose a continuous video coherence measure that gauges the extent of the current shot, reminding the viewer of a previous shot if the two shots are similar

(Kender, 1998). In Sundaram (2002a, 2002b), Sundaram and Chang propose a first-in-first-out (FIFO) memory model to detect audio and video scenes separately. An audio scene (a-scene) exhibits a long term consistency with respect to ambient sound, while a video scene (v-scene) is defined as a continuous segment of visual data that shows long-term consistency with respect to two properties: (1) chromaticity, and (2) lighting conditions. The segmentation results of audio scenes and video scenes are combined to obtain semantic scenes (termed as computable scenes, or c-scenes). Kender and Yeo's method and Sundaram and Chang's method are both memory-based. The difference is that Sundaram and Chang's method is based on a finite memory size, with an objective to speed up the processing time while maintaining segmentation performance similar to Kender and Yeo's method.

SEQUENTIAL METHODS FOR SHOT BOUNDARY DETECTION

Proposed Method

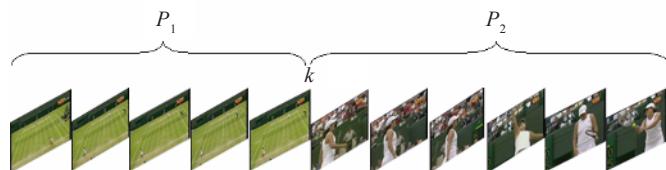
One way to look at the problem of SBD is by considering a sequence of frame features $X = \{X_1, X_2, \dots, X_M\}$ which are generated according to two different, identifiable distributions. A shot boundary can be detected by searching the change from one feature distribution P_1 to another P_2 before and after the shot boundary.

The problem can be formulated as follows:

Given a sequence of M features $X = \{X_1, X_2, \dots, X_M\}$ extracted from the corresponding frame sequence, there is an unknown change in the underlying feature distribution at frame k , where $1 < k \leq M$, such that the features are generated according to distribution P_1 before frame k and according to distribution P_2 after (and including) frame k . The goal is to detect this change (if existant) while minimizing the false detection rate.

Figure 4 further illustrates the formulation. There are two questions requiring answers in this formulation: (1) whether there is a change of underlying feature distribution within the frame sequence, and (2) where the change occurs if it exists.

Figure 4. Formulation of SBD as a sequential change detection problem



The problem can be formulated as a hypothesis testing between:

$$H_1: X_n \sim P_1, \text{ for } 1 \leq n \leq M$$

$$H_2: X_n \sim P_1, \text{ for } 1 \leq n < k$$

$$X_n \sim P_2, \text{ for } k \leq n \leq M$$

where H_1 is a simple hypothesis with no change in the underlying feature distribution, and $H_2 = \bigcup_{1 < k \leq M} H$ is a composite hypothesis with a change occurring at frame k .

In what follows, we first examine an optimal, off-line solution that requires all feature observations, and then present a customized algorithm that is suitable for online processing.

Off-Line Algorithm

Assuming that the feature observations from each scene follow i.i.d., the evidence in favor of hypothesis H_2 over hypothesis H_1 is given by the log-likelihood ratio:

$$\begin{aligned} J(k) &= \log \frac{\prod_{n=1}^{k-1} p_1(X_n) \prod_{n=k}^M p_2(X_n)}{\prod_{n=1}^M p_1(X_n)} \\ &= \sum_{n=k}^M \log \frac{p_2(X_n)}{p_1(X_n)} \end{aligned}$$

where k is the unknown abrupt change time. The maximum-likelihood (ML) estimate of this change time can be obtained as:

$$\hat{k} = \arg \max_{1 \leq k \leq M} J(k)$$

and the existence of a shot change can be determined by:

$$d = \begin{cases} H_1 & \text{if } \hat{J}(k) < THD \\ H_2 & \text{if } \hat{J}(k) \geq THD \end{cases}$$

where d is the decision choice of which hypothesis is true, THD is a threshold that depends on the priors of the hypotheses and the associated decision costs. Note that this algorithm can only be used for off-line detection because all feature observations are required to make the decision.

Online Algorithm

To develop an online algorithm for detecting the changes in the underlying feature distribution as soon as possible, we first consider the following property for the log-likelihood ratio of the feature observations:

$$E_{P_1}[\log \frac{p_2(X_n)}{p_1(X_n)}] \leq 0 \leq E_{P_2}[\log \frac{p_2(X_n)}{p_1(X_n)}]$$

where E_{P_1} and E_{P_2} denote the expectations of the log-likelihood ratio of the feature observations under distributions P_1 and P_2 , respectively. This inequality indicates that the sum of the log-likelihood ratios has a negative drift when the underlying distribution is P_1 , and it has a positive drift when the underlying distribution is P_2 . Similarly, the log-likelihood ratio $\log \frac{p_1(X_n)}{p_2(X_n)}$ has the following property:

$$E_{P_2}[\log \frac{p_1(X_n)}{p_2(X_n)}] \leq 0 \leq E_{P_1}[\log \frac{p_1(X_n)}{p_2(X_n)}]$$

To detect the change of log-likelihood ratio in SBD, we examine the following two *divergence measures*, which are symmetric extension and estimation of the Kullback information measure (Basseville, 1993).

In our proposed SBD method, we evaluate the divergence measures within a sliding window of observations. Suppose that frame k is the center frame of the window and $X = \{X_{k-N}, \dots, X_{k-1}, X_k, X_{k+1}, \dots, X_{k+N}\}$ are the frame color histograms (FCHs) within the window, where N is set to 8 in our implementation. We define the average and median-based divergence measures as follows:

$$J(k) = \frac{1}{N} \sum_{n=k-N}^{k-1} \log \frac{p_1(X_n)}{p_2(X_n)} + \frac{1}{N} \sum_{n=k+1}^{k+N} \log \frac{p_2(X_n)}{p_1(X_n)} \quad (2)$$

and

$$J(k) = \text{median}_{n=k-N}^{k-1} \log \frac{p_1(X_n)}{p_2(X_n)} + \text{median}_{n=k+1}^{k+N} \log \frac{p_2(X_n)}{p_1(X_n)} \quad (3)$$

where k is the postulated frame number at which a change in the underlying feature distribution occurs.

Since the two distributions before and after the boundary are unknown in SBD, and the number of observations available for estimating the distributions is limited, we use an k_n -nearest neighbor method (Duda, 2000) to estimate the probability of a frame being generated from each distribution. Furthermore, according to the general property of video, the frames having similar video content to one frame are very likely to appear near that frame. Hence, an k_n -nearest neighbor estimation, where k_n is set to 1, for approximating the probability of a frame being generated from each possible distribution is defined as follows:

$$p_1(X_n) = 1 - \min_j \left(\frac{\|X_n - X_j\|_d}{2} \right), \quad j = k-N, \dots, k-1$$

$$p_2(X_n) = 1 - \min_j \left(\frac{\|X_n - X_j\|_q}{2} \right), \quad j = k+1, \dots, k+N$$

where $\|X_n - X_j\|_q$ is the q -norm distance between FCHs X_n and X_j . Although other norm distances could be incorporated in this formulation and the proposed SBD method, we shall focus mainly on the 1-norm distance (i.e., $q=1$) for its good performance and ease of computation.

Figure 5 shows the measured changes $J(k)$ with different values of the parameter N for a test tennis video using the average-based and median-based divergence measures as defined in equation (2) and equation (3). In the equation, whether the postulated location k is a shot boundary or not depends on the ground truths, respectively. First, the figure shows that the proposed average-based and median-based divergence measures are effective in detecting video content changes. The measures computed for a shot boundary frame are much larger than that for a frame within a shot. Second, when the parameter N is larger than 7, the two measures computed have small variations. Obviously, too small a window size (i.e., $N=2$) cannot provide sufficient observations to estimate the probabilities p_1 and p_2 with good enough accuracy. Therefore, in our experiments, we set $N=8$.

Figure 6 shows the frames at the shot boundaries of eight shots from a test tennis video, in which a dissolve occurs between Shot 4 and Shot 5. Figure 7 shows the average-based and median-based divergence measures and the inter-frame histogram difference (IFHD) computed from the test tennis video. It can be observed from Figure 7 that the dissolve can be easily detected by using the average-based and median-based divergence measures. However, this is not the case for IFHD due to small changes between successive frames of a dissolve.

Figure 5. Divergence measures computed around a postulated change location k , where (a) k is at a shot boundary, and (b) k is not at a shot boundary of a test tennis video

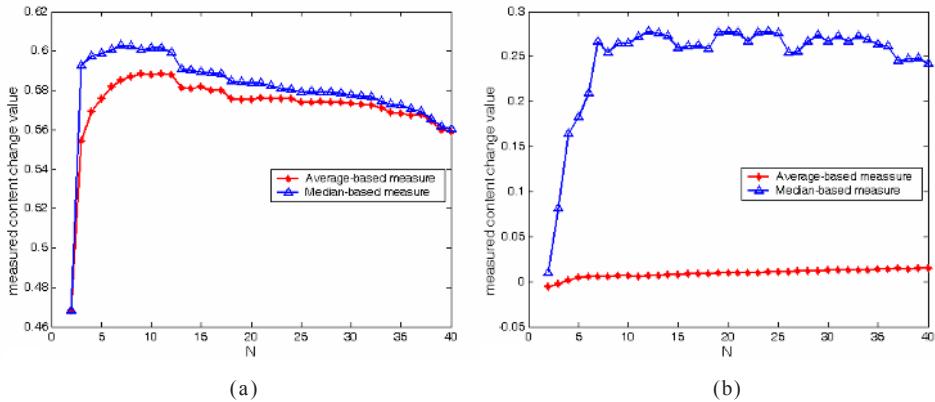
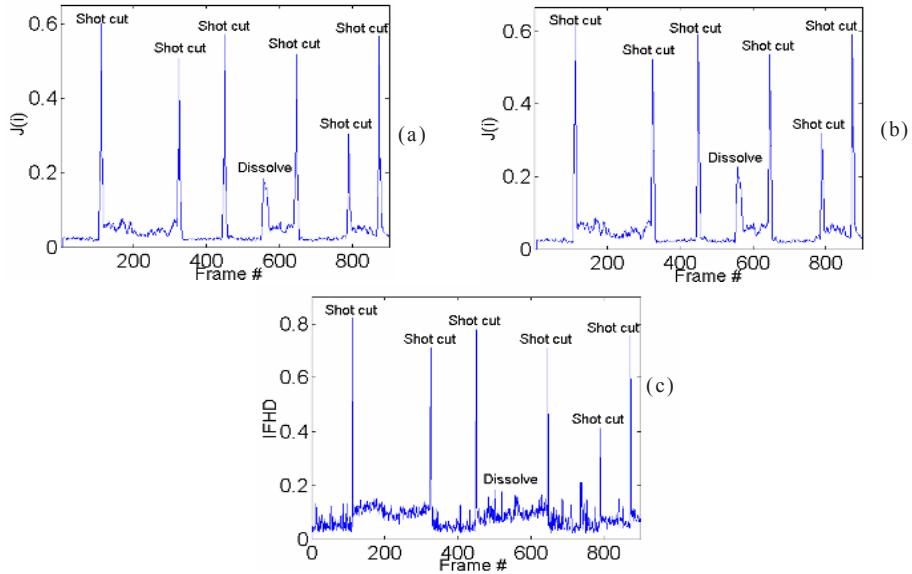


Figure 6. The first and the last frames of eight video shots from a test tennis video



Figure 7. (a) Average-based and (b) median-based divergence measures and (c) inter-frame histogram difference (IFHD) for a test tennis video



To detect a shot boundary, we identify the peak values by evaluating the following criterion within a sliding window:

$$\max_{k-W \leq i \leq k+W} J(i) \geq \max(T, \gamma z_m)$$

where $z_m = \text{median}_{k-W \leq i \leq k+W} J(i)$, $2W+1$ the width of the sliding window, T a threshold ensuring the change is large enough and γ an appropriate constant warranting a peak value. In other words, a peak value is detected at frame i^* when $J(i^*)$ assumes the maximum value is larger than a certain threshold value and γ times larger than the median value of all the $J(i)$ values within the sliding window. When using the average and median-

based divergence measures, we set $2W+1$ to 17. T and γ are determined using a peer group filtering (PGF) scheme (Kenney, 2001) in our experiments.

The main function of PGF is to separate a set of observations into two groups by maximizing the ratio of between-group distance to within-group distance (Kenney, 2001). Suppose $\{x_n, n=1, \dots, T\}$ is an observation sequence. For a specific observation x_i , the PGF generates a distance sequence $\{d_n, n=1, \dots, T\}$ of all observations from x_i and ranks the sequence in ascending order. The distance sequence is then separated into two groups at the location l , which is estimated by $\hat{l} = \arg \max_i \{(a_{i,1} - a_{i,2})^2 / (s_{i,1} - s_{i,2})\}$, where:

$$a_{i,1} = \frac{1}{i} \sum_{j=1}^i d_j, \quad a_{i,2} = \frac{1}{T-i} \sum_{j=i+1}^T d_j;$$

$$s_{i,1} = \sum_{j=1}^i |d_j - a_{i,1}|^2, \quad s_{i,2} = \sum_{j=i+1}^T |d_j - a_{i,2}|^2.$$

We extend the PGF scheme to estimate T and γ based on a simple observation that the log-likelihood ratios of a video feature sequence can be coarsely classified into two groups: shot boundary frames and frames within shots. Consider a subsequence of log-likelihood ratios computed from the input video FCH sequence,

$L(X_n) = \log(\frac{p_2(X_n)}{p_1(X_n)})$, $n=1, \dots, q$, where q is the training size and is set to half size of the test video sequence in our experiments, that is, $q=M/2$. Theoretically, the log-likelihood ratio $L(X_n)$ would be equal to zero for a frame X_n within a shot, and equal to nonzero when X_n is a shot change frame. For a specific observation $L(X_n)=0$, we form the distance sequence $\{SL(X_n)\}$, and obtain the location \hat{l} , which separates $\{SL(X_n)\}$ into two groups based on the PGF. We define the threshold T by: $T = SL(X_{\hat{l}})$.

Similarly, we investigate a sequence $\{R(X_n), n=p+1, \dots, q\}$, each of which is the log-likelihood ratio $L(X_n)$ to the median value of its prior p log-likelihood ratios ($p=5$ in our experiments), for determining γ . For an observation X_n from a frame within a shot, $R(X_n)=1$. We form the distance sequence for $R(X_n)=1$, and separate the sequence into two groups. γ is defined as two times of the distance value at the separating location.

Performance Evaluation for Shot Boundary Detection

The following two measures are commonly used to evaluate the performance of a SBD method or the detection results (Gargi, 2000):

$$\text{recall} = \frac{N_c}{N_c + N_m} \quad (4)$$

$$\text{precision} = \frac{N_c}{N_c + N_f} \quad (5)$$

where N_c is the number of correct detections, N_m the number of missed detections (or false negatives), N_f the number of false detections (or false positives), $N_c + N_m$ the total number

of actual shot boundaries (N_g) and $N_c + N_f$, the total number of detections by a SBD method (N_d). Generally, the recall gives the probability of correct detections, while the precision indicates how precise the detections are. For a good SBD method, both recall and precision are desired to be high.

We further employ two additional measures to evaluate the performance of SBD, over-segmentation and under-segmentation, which are used in image segmentation (Hoover, 1996). We employ the two measures to determine the overlapping between the ground-truth shots and the detected shots.

Suppose N_{s_n} is the number of frames of a shot s_n in the ground truths, N_{s_m} is the number of frames of a shot s_m detected by a SBD method and $O_{mn} = N_{s_m} \cap N_{s_n}$ is the number of frames overlapped between shots s_m and s_n . A shot s_n in the ground truths and a set of detected shots s_{m1}, \dots, s_{mx} , where $2 \leq x \leq N_d$, are classified as an over-segmentation if $\forall i \in x, O_{mi,n} \geq T \times N_{s_m}$ and $\sum_{i=1}^x O_{mi,n} \geq T \times N_{s_n}$. A set of shots s_{n1}, \dots, s_{nx} in the ground truths, where $2 \leq x \leq N_g$ and a detected shot s_m , are classified as an under-segmentation if $\sum_{i=1}^x O_{mi,n} \geq T \times N_{s_m}$ and $\forall i \in x, O_{mi,n} \geq T \times N_{s_n}$. In our implementation, T is set to 0.8 as recommended in Hoover (1996).

The over-segmentations produce smaller detected shot segments than the ground truths, while the under-segmentations produce larger detected shot segments than the ground truths. For a SBD method, both over-segmentation and under-segmentation are desired to be low, while high recall and precision are to be maintained.

Experimental Results for Shot Boundary Detection

We have tested the proposed SBD method using a variety of test videos as listed in Table 1. Among these test videos, the news is from the MPEG-7 test set (ISO/IEC JTC1/SC29/WG11/N2467), the movie is from the feature film “You’ve Got Mail” and the rest are from a variety of broadcast TV programs including sports videos, sitcoms and commercials. All the test videos are digitized at 25 frames/second, with a total length of 109,761 frames, i.e., more than 70 minutes. The total number of shots is 770, including 716 shot cuts and 54 gradual transitions. In these test videos, the news and movie have more gradual transitions, the tennis and soccer have more motion and the commercial has many short shots. As such, a large number of shot boundaries in these test videos pose difficult challenges to many existing SBD methods.

To evaluate the performance of the existing and our proposed SBD methods, the ground-truth shot boundaries of the test videos are manually identified and labeled as shot cuts or gradual transitions. Three adaptive thresholding methods, (1) adaptive

Table 1. Number of frames, shots, shot cuts and gradual transitions of the test videos

Video	Frame #	Shot #	ASB #	GSB #
Tennis	8711	46	38	8
Soccer	12563	48	41	7
News	28492	187	165	22
Commercials	1382	43	40	3
Sitcom	16305	133	130	3
Movie	42308	313	302	11
Total	109761	770	716	54

Table 2. Performance of SBD by using Zhang et al.'s adaptive thresholding method (1), twin-comparison method (2), entropic thresholding method (3), and our proposed method using the average-based (Average) and median-based (Median) divergence measures, respectively

Test video	Method	Shot cuts			Gradual transitions			All shot boundaries						
		N_g	N_d	N_o	N_g	N_d	N_o	N_g	N_o	N_d	N_o	N_s	Recall	Precision
Tennis	Method 1	38	67	34	8	0	0	46	67	34	26	8	0.74	0.51
	Method 2	38	81	35	8	2	2	46	83	37	28	5	0.80	0.45
	Method 3	38	190	37	8	5	5	46	155	42	39	2	0.91	0.27
	Average	38	38	37	8	7	7	46	45	44	1	1	0.96	0.98
	Median	38	41	38	8	7	7	46	48	45	3	1	0.98	0.94
Soccer	Method 1	41	45	41	7	0	0	48	45	41	4	5	0.85	0.91
	Method 2	41	82	41	7	2	2	48	84	48	22	0	0.90	0.51
	Method 3	41	78	41	7	1	1	48	79	42	21	4	0.88	0.53
	Average	41	41	41	7	5	5	48	46	46	0	2	0.96	1.00
	Median	41	42	41	7	5	5	48	48	46	1	2	0.96	0.96
News	Method 1	165	263	122	22	15	15	187	278	137	89	32	0.73	0.49
	Method 2	165	305	125	22	19	19	187	324	141	91	23	0.77	0.44
	Method 3	165	501	164	22	19	19	187	520	183	179	2	0.98	0.35
	Average	165	166	164	22	19	19	187	185	183	1	3	0.98	0.99
	Median	165	173	164	22	21	21	187	194	185	6	2	0.99	0.95
Commercials	Method 1	40	6	6	3	1	1	43	7	7	0	7	0.16	1.00
	Method 2	40	7	7	3	2	2	43	9	9	0	7	0.21	1.00
	Method 3	40	69	38	3	3	3	43	72	41	20	2	0.95	0.57
	Average	40	37	36	3	3	3	43	40	39	1	4	0.91	0.98
	Median	40	38	36	3	3	3	43	41	38	2	4	0.91	0.95
Sitcom	Method 1	130	96	95	3	0	0	133	96	95	1	14	0.71	0.99
	Method 2	130	131	95	3	0	0	133	131	95	17	12	0.71	0.71
	Method 3	130	203	130	3	0	0	133	203	130	32	3	0.98	0.64
	Average	130	130	130	3	0	0	133	130	130	0	3	0.98	1.00
	Median	130	132	130	3	0	0	133	132	130	1	3	0.98	0.98
Movie	Method 1	302	256	246	11	3	3	313	259	249	3	38	0.80	0.96
	Method 2	302	362	250	11	4	4	313	366	254	41	23	0.80	0.96
	Method 3	302	458	299	11	7	7	313	306	465	54	2	0.98	0.66
	Average	302	307	300	11	4	4	313	311	304	6	4	0.97	0.98
	Median	302	306	299	11	6	6	313	312	305	5	6	0.98	0.98

thresholding method (Zang et al., in Furht, 1995), (2) twin-comparison method (Furht, 1995) and (3) entropic thresholding method (Yu, 2001) are used for comparison. Table 2 shows the results of SBD obtained by the three methods and our proposed method using the average-based (average) and median-based (median) divergence measures, respectively.

The numbers of N_g , N_c and N_d are presented for each category of shot boundaries, that is, shot cuts, gradual transitions and all shot boundaries (both shot cuts and gradual transitions). We have measured the shot boundary detection performance by the over-segmentation (N_g), under-segmentation (N_d), recall and precision for all shot boundaries. The results are tabulated in Table 2.

It can be observed from Table 2, as the features used by method 1 (Zhang et al.'s adaptive thresholding method) and method 3 (entropic thresholding method) are based on IFHDs, their detection recalls and precisions depend very much on the detection thresholds used. In general, when the precision is high, the recall is low; when the recall is high, the precision is low. By using a twin-comparison method, the recall of SBD, especially that for gradual transitions, is improved. However, the improvement is usually gained at the expense of lower precision. In comparison, our proposed SBD method can obtain good detection performance in both recall and precision, especially for gradual transitions. The averaged recalls and precisions for all the shot boundaries are 0.97 and

0.99 by the average-based divergence measure, and 0.97 and 0.97 by the median-based divergence measure. At the same recall value, the average-based divergence measure can generally obtain marginally better precision than the median-based measure. Furthermore, our method can obtain smaller over-segmentations and under-segmentations of shots, when maintaining high correct detections.

We have also implemented the GLR algorithm proposed in Lelescu (2003) to detect shot boundaries for our test videos. The averaged detection performance is high in recall (0.82) and low in precision (0.36). The best result reported in Lelescu (2003) is 0.92 in recall and 0.72 in precision. Hence, our proposed SBD method is likely to perform better than the existing GLR algorithm due to the use of a different statistics assumption and different estimation of the underlying feature distributions.

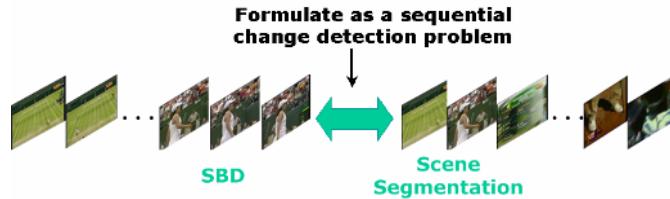
SEQUENTIAL METHODS FOR SCENE SEGMENTATION

Proposed Method

As scene segmentation and SBD share a similar objective of partitioning a sequence of observations (shots for scene segmentation and frames for SBD) into temporally adjacent groups where the visual contents are more similar within the same group than between different groups, we can also formulate scene segmentation as a sequential change detection problem similar to that of SBD. The formulation of SBD and scene segmentation as sequential change detection is shown in Figure 8. Note that the main difference between SBD and scene segmentation lies in the fact that the feature sequence used in SBD is extracted from frames and the one used in scene segmentation is extracted from shots. Specifically, in SBD, frame color histograms (FCHs) are used as features, and in scene segmentation, shot color histograms (SCHs), which are defined as the bin-wise average of the FCHs within the shots, are used.

Consider an example sequence of shots $ABABCD\bar{C}D$. The goal of scene segmentation is to detect the boundary between A, B shots and C, D shots. The distributions of observations before and after the scene boundaries are to be estimated based on A, B shots and C, D shots, respectively. Specifically, suppose that shot k is the center shot of the window and $X = \{X_{k-N}, \dots, X_{k-1}, X_k, X_{k+1}, \dots, X_{k+N}\}$ are the shot color histograms (SCHs) within the window, where N is set to 8 in our implementation. We consider both the

Figure 8. Formulation of scene segmentation as a sequential change detection problem similar to that of SBD



average-based and median-based divergence measures as defined in equation (2) and equation (3). Furthermore, the probabilities of each shot being generated from the two distributions can also be estimated by using the k_n -nearest neighbor method (k_n is set to 1).

Similarly, the existence of scene boundaries can be detected by a decision rule that identifies the peak values of the divergence measures as shown below:

$$\max_{k-W \leq i \leq k+W} J(i) \geq \max(T, \gamma z_m)$$

where $z_m = \text{median}_{k-W \leq i \leq k+W} J(i)$, the window size $2W+1$ is set to 11 and T and γ are determined by the PGF scheme.

Experimental Results for Scene Segmentation

We have conducted experiments to compare the scene segmentation performance of our proposed method against Kender and Yeo's method. Since the definition of scene is rather subjective, we first present in the following some visual results of scene segmentation.

Figures 9 and 10 show the scene segmentation measures and results for the first 100 shots of a test news video obtained by using the average-based divergence measure and Kender and Yeo's video coherence measure. In Figure 9, the ground truths of the scenes are shown in blocks with different background colors. Sample frames corresponding to the scene boundaries and that of the preceding and succeeding shots are given in Figure 10 to show the change in visual contents from one scene to another. In this example, a scene often corresponds to a news story. Whether the anchor-person shot belongs to the preceding or the succeeding scene is determined by the visual contents of the anchor-person shot being more similar to one scene or the other.

Figure 9. (a) Proposed median-based divergence measure and (b) Kender and Yeo's video coherence measure for the first 100 shots of a test news video

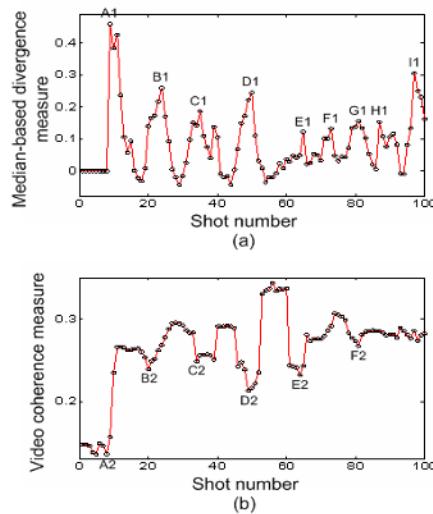
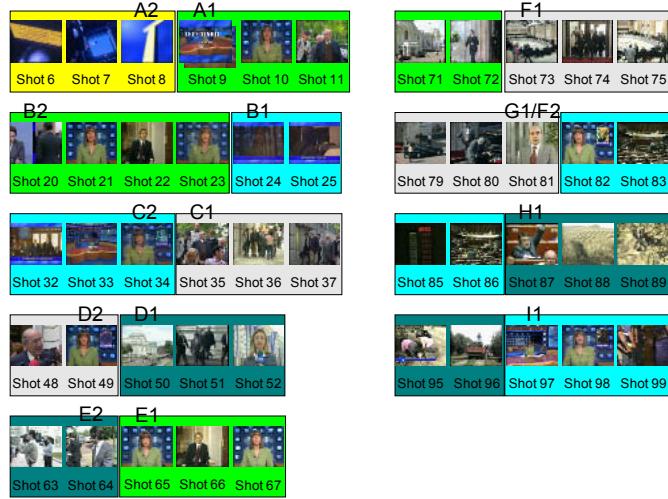


Figure 10. Scene segmentation result for the first 100 shots of a test news video



It can be observed from Figure 9 and Figure 10 that, although both our proposed method and Kender and Yeo's method can detect most of the scene boundaries for the test news video, our method is more accurate in detecting the location of scene boundaries. For example, the locations of scene boundaries A1, B1, C1, D1, and E1 detected by our proposed method are the same as the ground truths. These scene boundaries are detected by Kender and Yeo's method with one to a few shots difference from the locations of the ground truths (A2, C2, D2 and E2). For example, for scene boundary B2, the location of scene boundaries detected by Kender and Yeo's method is four shots away from the ground truth. Furthermore, scene boundaries F1, H1 and I1 are detected by our proposed method and missed by Kender and Yeo.

As the tennis and news videos in Table 1 cover only a few scenes, we also use the long tennis and the long news for scene segmentation, which are extracted from the same sources as the tennis and news but with longer time durations. The long tennis includes 17,982 frames, and is composed of 137 shots. The long news includes 71,379 frames, and is composed of 476 shots. The test videos dataset for scene segmentation comprises the two long videos, and the commercials, sitcom and movie test videos as shown in Table 1.

Table 3 shows the results of scene segmentation by using the Kender and Yeo memory-based method and our method using the average-based and median-based divergence measures. It can be observed that our proposed method can obtain more scene changes with higher recalls and precisions, and smaller over-segmented and under-segmented scenes than the memory-based method.

CONCLUDING REMARKS

This chapter reviews shot boundary detection (SBD) and scene segmentation for content-based video analysis. As SBD is the first and important step towards the

Table 3. Performance of scene segmentation by using the Kender and Yeo memory-based method (Kender, 1998) and the proposed method using the average-based and median-based divergence measures

Video	Method	N_g	N_d	N_e	N_o	N_u	Recall	Precision
Long Tennis	Memory Advantage	5	4	2	0	1	0.40	0.50
	Media	5	4	4	0	1	0.80	1.00
		5	5	3	0	1	0.60	
Long News	Memory Advantage	46	21	9	1	10	0.20	0.43
	Media	46	18	13	1	7	0.28	0.72
		46	22	7	2	10	0.15	0.32
Commercials	Memory Advantage	2	2	2	0	0	1.00	1.00
	Media	2	2	2	0	0	1.00	1.00
		2	2	2	0	0	1.00	1.00
Sitcom	Memory Advantage	11	3	5	0	0	0.30	0.60
	Media	10	5	5	0	4	0.50	1.00
		10	9	6	2	2	0.60	0.67
Movie	Memory Advantage	22	12	10	2	0	0.45	0.83
	Media	22	9	7	0	5	0.32	0.78
		22	16	13	1	4	0.59	0.81

understanding of high-level video semantics, a large variety of detection algorithms have been proposed over the past decade. However, we show in this chapter that the detection accuracy can still be further improved. By using a sequential change detection algorithm, we have developed a new SBD method and achieved consistent and improved detection accuracy in both recall and precision. By formulating the scene segmentation as a sequential change detection problem, we also extend the method proposed for SBD to segment video shots into scenes. Compared with the existing scene segmentation methods, such as the memory-based methods proposed by Kender and Yeo as well as Sundaram and Chang, the experimental results show that our proposed method can obtain more accurate scene segmentation results. The main innovation of the proposed methods is that SBD and scene segmentation can be performed in a unified manner.

As for the future research direction, SBD based on some learning methods has become popular and promising (Chua, 2003; Rong, 2005). Furthermore, differentiating types of shot boundaries can also help infer useful semantics in video content (Lu, 2005). For example, in sports video, wipes are normally used for replays or fouls. Also, fades normally correspond to the start or end of a program. In scene segmentation, since the definition of scene is subjective and could be different for different types of videos, more objective performance measures and algorithms that are customized to specific types of video are expected to be more and more important and useful.

ACKNOWLEDGMENTS

This work was supported in part by Natural Science Foundation of China under contracts 60533100, 60373020 and 60402007, and The Open Research Foundation of Shanghai Key Laboratory of Intelligent Information Processing.

REFERENCES

- Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes: Theory and application*. Englewood Cliffs, NJ: Prentice Hall.
- Brown, P., Smeaton, A. F., Murphy, N., O'Connor, N., Marlow, S., & Berrut, C. (2000). Evaluating and combining digital video shot boundary detection algorithms. *Irish Machine Vision and Image Processing Conference, UK*, Belfast, Northern Ireland (pp. 93-100).
- Chua, T.-S., Feng, H., & Charndrashekara, A. (2003). An unified framework for shot boundary detection via active learning. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2, 845-848.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: Wiley-Interscience.
- Furht, B., Smoliar, S. W., & Zhang, H. J. (1995). *Video and image processing in multimedia systems*. Boston: Kluwer Academic Publisher.
- Gargi, U., Kasturi, R., & Strayer, S. H. (2000). Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1), 1-13.
- Hoover, A., J.-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., et al. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, 673-689.
- Ide, I., Hamada, R., Sakai, S., & Tanaka, H. (2000). Scene identification in news video by character region segmentation. *ACM International Conference on Multimedia*, 195-200.
- ISO/IEC JTC1/SC29/WG11/N2467. (1998). *Description of MPEG-7 content set*. Atlantic City.
- Kender, J. R., & Yeo, B.-L. (1998). Video scene segmentation via continuous video coherence. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 367-373.
- Kenney, C., Deng, Y., Manjunath, B., & Hewer, G. (2001). Peer group image enhancement. *IEEE Transactions on Image Processing*, 10, 326-334.
- Koprinska, I., & Carrato, S. (2001). Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16, 477-500.
- Lelescu, D., & Schonfeld, D. (2003). Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream. *IEEE Transactions on Multimedia*, 5(1), 106-117.
- Lu, H., Wang, B., Xue, X., & Tan, Y.-P. (2005). Effective shot boundary classification using video spatial-temporal information. *International Symposium on Circuits and Systems* (pp. 3837-3840).
- Miyamori, H., & Iisaku, S.-I. (2000). Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 320-325).
- Nack, F., & Lindsay, A. T. (1999a). Everything you wanted to know about MPEG-7: Part 1. *IEEE Multimedia*, 6(3), 65-77.
- Nack, F., & Lindsay, A. T. (1999b). Everything you wanted to know about MPEG-7: Part 2. *IEEE Multimedia*, 6(4), 64-73.

- Nam, J., Alghoniemy, M., & Tewfik, A. H. (1998). Audio-visual content-based violent scene characterization. In *Proceedings of the IEEE International Conference on Image Processing* (Vol. 1, pp. 353-357).
- O'Toole, C., Smeaton, A., Murphy, N., & Marlow, S. (1999). *Evaluation of automatic shot boundary detection on a large video test suite*. Presented at the 2nd UK Conference on Image Retrieval: The Challenge of Image Retrieval, Newcastle.
- Pereira, F. (1999). MPEG-7: A standard for describing audiovisual information. *IEE Colloquium on Multimedia Databases and MPEG-7* (Ref. No. 1999/056), 6, 1-6.
- Quenot, G. M., Moraru, D., & Besacier, L. (2003). CLIPS at TRECVID: Shot boundary detection and feature detection. In *Proceedings of the 2003 Text Revival Conference (TREC 2003)*, Gaithersburg, MD.
- Rong, J., Ma, Y.-F., & Wu, L. (2005). Gradual transition detection using EM curve fitting. In *Proceedings of the 11th International Multimedia Modeling Conference* (pp. 364-369).
- Rui, Y., Huang, T. S., & Mehrotra, S. (1999). Constructing table-of-content for videos. *ACM Journal of Multimedia Systems*, 7(5), 359-368.
- Smoliar, S. W., & Zhang, H. J. (1994). Content-based video indexing and retrieval. *IEEE Multimedia*, 1(2), 62-72.
- Sundaram, H. (2002b). *Segmentation, structure detection and summarization of multimedia sequences*. PhD thesis, Columbia University.
- Sundaram, H., & Chang, S.-F. (2002a). Computable scenes and structures in films. *IEEE Transactions on Multimedia*, 4(4), 482-491.
- Wang, Y., Liu, Z., & Huang, J.-C. (2000). Multimedia content analysis: Using both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6), 12-36.
- Wu, L. D., Guo, Y. F., Qiu, X. P., Feng, Z., Rong, J. W., Jin, W. J., et al. (2003). Fudan University at TRECVID 2003. In *Proceedings of the 2003 Text Revival Conference (TREC 2003)*, Gaithersburg, MD.
- Xu, P., Xie, L., Chang, S.-F., Divakaran, A., Vetro, A., & Sun, H. (2001). Algorithms and system for segmentation and structure analysis in soccer video. *IEEE International Conference on Multimedia and Expo*, Tokyo (pp. 721-724).
- Yeung, M., Yeo, B.-L., & Liu, B. (1996). Extracting story units from long programs for video browsing and navigation. *IEEE International Conference on Multimedia Computing and Systems* (pp. 296-305).
- Yu, J., & Srinath, M. D. (2001). An efficient method for scene cut detection. *Pattern Recognition Letters*, 22(13), 1379-1391.
- Zhai, Y., Rasheed, Z., & Shah, M. (2003). University of Central Florida at TRECVID 2003. In *Proceedings of the 2003 Text Revival Conference (TREC 2003)*, Gaithersburg, MD.
- Zhang, H. J., Wu, J. H., & Smoliar, S. W. (1997, June 3). *System for automatic video segmentation and key frame extraction for video sequences having both sharp and gradual transitions* (U.S. Patent, 5,635,982).
- Zhong, D., & Chang, S.-F. (2001). Structure analysis of sports video using domain models. *IEEE International Conference on Multimedia and Expo*, Tokyo (pp. 713-716).
- Zhong, D. (2001). *Segmentation, index and summarization of digital video content*. Doctoral dissertation, Columbia University.

Section IV: Segmenting Particular Images

Chapter X

Color Image Segmentation in Both Feature and Image Spaces

Shengyang Dai, Northwestern University, USA

Yu-Jin Zhang, Tsinghua University, Beijing, China

ABSTRACT

One critical problem in image segmentation is how to explore the information in both feature and image space and incorporate them together. One approach in this direction is reported in this chapter. Watershed algorithm is traditionally applied on image domain but it fails to capture the global color distribution information. A new technique is to apply first the watershed algorithm in feature space to extract clusters with irregular shapes, and then to use feature space analysis in image space to get the final result by minimizing a global energy function based on Markov random field theory. Two efficient energy minimization algorithms: Graph cuts and highest confidence first (HCF) are explored under this framework. Experiments with real color images show that the proposed two-step segmentation framework is efficient and has been successful in various applications.

INTRODUCTION

Color image segmentation can be modeled as a labeling problem for each pixel in the entire image. In the optimal assignment, pixels having the same label should have the following two properties: one, that they must have small distance in features (such as color, texture, etc.) space, the other is that they must be spatially coherent. One natural way of combining the above properties together is treating spatial position as two

additional feature domains, and retrieve clusters in the extended feature space. However, recent research (Zabih, 2004) shows that segmentation tends to split large coherent regions, thus leading to poor results.

Treating the above two properties separately will lead to a two-phase framework. Pixels are clustered in feature and spatial spaces in different phases.

Many feature space clustering algorithms have been proposed in the literature (Jain, 2000). To be specific, for color space analysis, algorithms can be roughly classified into two groups: parametric and nonparametric methods. Due to the irregular shape of clusters, more attention has been devoted to nonparametric methods. Watershed has been widely used in spatial domain analysis. It can be naturally applied on feature space because it is particularly suitable to describe clusters with hill-like shapes.

Continuous regions are usually preferred for segmentation tasks. Using feature space analysis alone will not satisfy this requirement; spatial coherence should be added for this purpose. Markov random field (MRF) theory is very powerful in modeling image spatial coherence. In MRF, image segmentation problems are considered as a realization of a random field defined on pixels. Energy is defined on cliques, and stable state can be achieved when the total energy reaches the minimum (Li, 1995). Both local smoothness and global feature properties may be captured. This capturing is commonly used as a refining technique for initial labeling results. Computation complexity is a big issue when using MRF. Two algorithms can achieve a nice suboptimal result efficiently. Highest confidence first (Chou, 1990) is a greedy minimization algorithm. Graph cuts is very popular recently due to its high speed and theory soundness (Boykov, 2001). They are selected in the spatial analysis phase of our segmentation algorithm.

BACKGROUND AND RELATED WORKS

Watershed and Feature Space Analysis in Image Segmentation

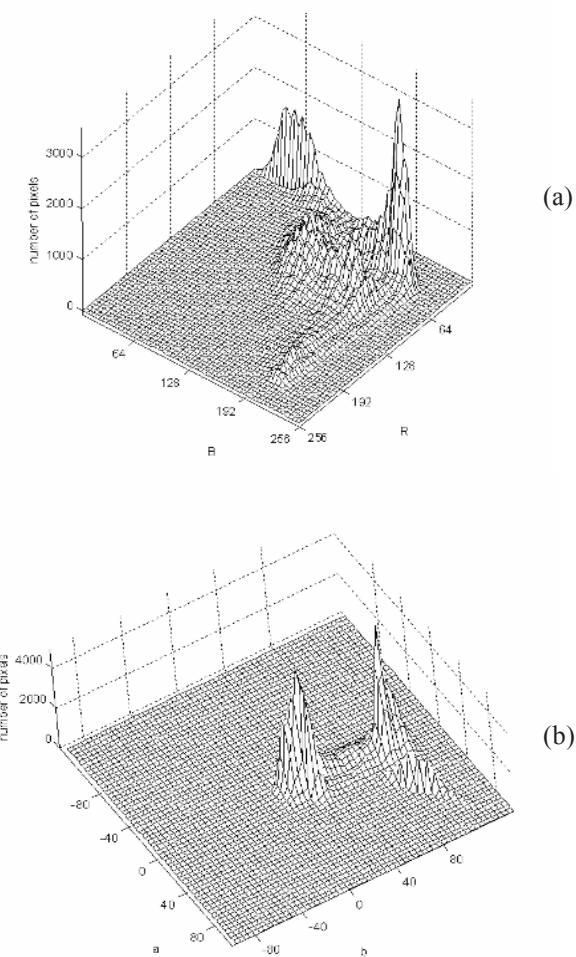
The watershed transform, originally proposed by Beucher (1982) and improved with fast implementation methods by Vincent (1991), is traditionally classified as a region-based segmentation approach (Roerdink, 2000). It is a classic segmentation algorithm, and has been widely used (Patras, 2001; Wang, 1998).

The idea underlying watershed transform comes from geography. It is imagined that a landscape is immersed in a lake, with holes pierced in local minima. Basins will be filled up with water starting at these local minima. At points where water coming from different basins meet, dams will be built. When the water level has reached the highest peak in the landscape, the process is stopped. As a result, the landscape is partitioned into regions separated by dams, called watershed lines or watersheds (Roerdink, 2000).

In most cases, the watershed algorithm is applied on image domain (usually on the edge image). It focuses on local color feature instead of global color distribution. In edge image, the local minima exist in the interior of objects and high altitude appears on the boundary of objects. After a flooding process, dams (watershed lines) will be constructed on object boundary and different objects are separated. In this way, it captures only information of local color feature instead of global color distribution.

In feature space, one obstacle of segmentation is the difficulty of reliance on color clustering. Researchers have noticed (Park, 1998; Pauwels, 1999) that color distribution in 3-D color space cannot be well approximated by the traditional parameter-based clustering algorithm (such as K -mean model or Gaussian mixture model). For example, K -mean is unable to handle unbalanced or elongated clusters. Gaussian mixture model is not appropriate for clusters with irregular shapes. In Figure 1, we project the pixel distribution of PEPPERS in 3-D color space onto 2-D plane. The result on RGB space is shown in Figure 1(a), the distribution has irregular shapes; some clusters are sharp and

*Figure 1. Pixel distribution in color space of PEPPERS (projection on BR or a^*b^* plane)*



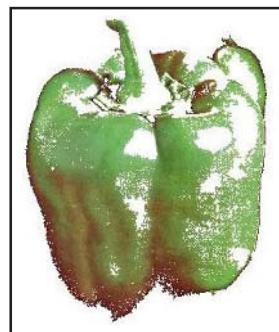
compact and some are flat. Figure 1b shows the result on $L^*a^*b^*$ space; the clusters in this space seems more salient, but it is still hard to get the boundary between clusters with parametric models. Besides, the determination of cluster number is another problem for these models. So people resort to some form of non-parameter clustering techniques. One algorithm based on mathematical morphology in 3-D color space is described below (Park, 1998). A closing operation and adaptive dilation are used to extract the number of clusters and their representative values. Various color spaces such as RGB, XYZ, YIQ, UVW and $I_1I_2I_3$ are tested; however, $L^*a^*b^*$ color space has not been included. A non-parameter algorithm to get color cluster in feature space is also proposed (Pauwels, 1999).

Watershed can be used to find clusters in feature space. If the 3-D histogram is reversed, the local minima will be located at the places where there are locally maximum numbers of corresponding pixels. The dams (watershed lines) will be constructed on the dividing lines of different color clusters to split the color histogram into several parts with different dominant colors. By mapping the clusters back to image domain, separated regions can be obtained to complete the segmentation. This technique has also been applied on RGB space to get the coarse segmentation result (Geraud, 2001). However, the RGB space is not proportional to human perception, thus limiting the performance.

The underlying idea of applying watershed on clustering color histograms is to conform to intuition. For nature images, the color changeover inside a region is usually smooth, so the pixel distribution in color space for the region with gradually changed color tends to form a (generalized) hill-like shape. The watershed algorithm is suitable to model such shapes (after reversion). Figure 2 shows one of the continuous regions of a color cluster extracted by watershed on $L^*a^*b^*$ color space, the color transition part is well captured.

It is known that watershed transform may produce quite thick watersheds due to the plateau between basins. If a plateau exists, watershed points may pervade and form thick dams (edges) in the flooding process. This problem has little influence on feature space watershed, because a plateau tends to appear in the histogram with fewer pixels belongs to it (after histogram reversion). So, since there is only a very small portion of

*Figure 2. One continuous region of a color cluster captured by watershed on $L^*a^*b^*$ color space*



pixels corresponding to the plateau, they can be processed in the second stage with other techniques in image space analysis.

Image Space Analysis by Markov Random Field

To get coherent regions, image space analysis is commonly used as the second step of image segmentation to refine the result from feature space analysis.

Markov random field describes an image by local interactions between adjacent pixels. Due to the local property of this theory, it is very suitable to conduct the image space analysis in segmentation applications. The feature property can also be easily incorporated by adding the observation term in the energy function.

Usually an initial state is required to start, and many techniques have been developed to get an optimal solution of Markov random field by updating the labeling state. So it is particularly suitable to refine a coarse result, and get a final result with desirable property on both feature and image space.

Markov random field has been used to improve the initial segmentation result obtained from a region-growing algorithm (Mukherjee, 2002). The energy function provides the measure of interclass and intra-class deviation. Simulated annealing technique is employed for minimizing the energy function.

The coarse segmentation result can be obtained from the one-step mean shift algorithm (Yang, 2001). The multilevel logistic (MLL) distribution model is employed for the purpose of smoothing regions with its characteristic of region forming, and the boundary information is added to the energy function of MLL distribution to preserve the discontinuity at boundaries. Also, the coarse image segmentation result is refined through the process of simulated annealing.

Two factors limit the efficiency of MRF-based algorithms: First, the commonly used simulated annealing technique consists of a large number of iteration steps at each temperature. In each step, energy related to all pixels needs to be computed, and the temperature itself needs to decrease gradually, so the computation load is usually very high. Secondly, all pixels in the image are usually updated simultaneously, thus the pixels at the border of regions are likely to be improperly segmented in the process in order to meet the continuity constraint, which will negatively affect the segmentation process and increase the computation load in the meanwhile.

Besides simulated annealing, there are other optimization techniques, such as iterated conditional modes (ICM) (Besag, 1986), relaxation labeling (Chou, 1990) and mean field annealing (Geiger, 1991), however they also suffer from heavy computational load.

Efficient Energy Minimization

How to minimize the energy efficiently is a key problem. Actually, it is NP-hard and has many local minima. Various efficient algorithms have been proposed to find a suboptimal solution.

Highest confidence first (HCF) (Chou, 1990) is an efficient technique for energy minimization. It is suitable for the case of assigning labels to pixels with unknown labels, because it introduces an uncommitted label. A stability measurement is defined for each uncommitted pixel. In each step, the greedy algorithm is applied to update the label of the least stable uncommitted pixel to get the most energy reduction. It is a deterministic

algorithm, having good performance on minimizing the energy. In this chapter we used a modified version of HCF, which can enforce the region continuity smoothly and efficiently.

Graph cuts is widely used in energy minimization (Greig, 1989) and clustering (Wu, 1993). When there are only two labels, the energy minimization problem can be reduced directly to a problem of computing the max-flow (or min-cut) of a graph, which can be solved efficiently in polynomial time (Boykov, 2001). This method was further generalized by allowing arbitrary label sets (Boykov, 2001; Kolmogorov, 2004). They developed very efficient graph-based methods to find the optimal α - β -swap and α -expansion given an initial labeling state. By using α -expansion iteratively, the NP-hard minimization problem can be approximately solved within a guaranteed optimality boundary. The only constrain is the metric and semi-metric of the smoothness energy function, which can cover fairly general classes of function.

THE PROPOSED ALGORITHM

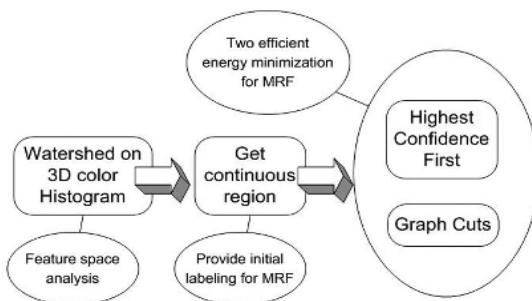
The entire framework is outlined in Figure 3. The watershed algorithm on a 3-D color histogram captures the feature space information. The initial labeling state is achieved after continuous region analysis. Two efficient energy minimization techniques for MRF are followed to get the final segmentation result.

Watershed Based on Color Histogram

Getting the Histogram and Pre-Processing

$L^*a^*b^*$ color space is adopted in the proposed algorithm, as the color difference in this space is proportional to human perception. The following equations are used to convert color from RGB color space to $L^*a^*b^*$ color space (Cheng, 2001):

Figure 3. Flow chart



$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$L^* = 116 \times (Y/Y_0)^{1/3} - 16$$

$$a^* = 500 \times [(X/X_0)^{1/3} - (Y/Y_0)^{1/3}]$$

$$b^* = 200 \times [(Y/Y_0)^{1/3} - (Z/Z_0)^{1/3}]$$

where $X/X_0 > 0.01$, $Y/Y_0 > 0.01$, $Z/Z_0 > 0.01$ and X_0, Y_0, Z_0 are the X, Y, Z values for the standard white.

The $L^*a^*b^*$ values of each pixel in the input image correspond to one point in the $L^*a^*b^*$ color space. Then, all pixels with the same $L^*a^*b^*$ values are accumulated to form the 3-D histogram for the input image. L^* , a^* and b^* components are uniformly quantized into u, v, w intervals, respectively. If this 3-D space were regular in all directions, the total number of bins would be $u \times v \times w$ (in our experiment, $u=50, v=w=60$, so there are 1.8×10^5 bins in total). However, as the $L^*a^*b^*$ space is not cubic-like, so not all the combinations of L^*, a^* and b^* values are valid. In other words, the number of meaningful bins would be rather smaller than $u \times v \times w$ (about 4.6×10^4 in our experiment).

The 3-D Gaussian filter is used to smooth the 3-D $L^*a^*b^*$ histogram. Then, any color bins with corresponding pixel numbers smaller than a predefined threshold are labeled as 0 (pixels labeled 0 are called uncommitted pixels, this is the first source of the uncommitted pixels). These preprocessing steps are aimed to remove the spurious and noisy components. Some other pixels such as those near sharp edges will also be labeled 0 and the following step will decide to which region they belong.

Now, we get the preprocessed histogram $H(x, y, z)$ ($0 \leq x < u, 0 \leq y < v, 0 \leq z < w$).

Watershed Algorithm

The 3-D watershed algorithm is performed on the $L^*a^*b^*$ histogram $H(x, y, z)$. An immersing watershed algorithm (Vincent, 1991) with 26-neighborhood is used on the reverse $L^*a^*b^*$ histogram to obtain the correct result because this immersing watershed algorithm begins with each local minimum. The following procedure is used to cluster the color histogram by a labeling process:

1. Get the reverse histogram $H'(x, y, z) = -H(x, y, z)$ ($0 \leq x < u, 0 \leq y < v, 0 \leq z < w$);
2. Get all local minimum (26-neighborhood in 3-D histogram) of the reverse histogram H' , label them as $1, 2, 3, \dots, m$;
3. Find the unlabeled bin in H' with minimum value and label it according to its neighbors:
 - a. If more than one label appears in its 26-neighborhood, it is a “dam” bin, and we label it as 0 (the second source of the uncommitted pixels).
 - b. If else, label it the same as its labeled neighbor. (Note that it is impossible that all of its neighbors are unlabeled, because all local minimums are already labeled in step (2)).

4. Go to step (3) until all non-zero bins are labeled.

Now, all color histograms with non-zero values are labeled as $0, 1, 2, \dots, n$.

Post-Processing

After obtaining watershed in the color histogram, we bring the result back to the image space. The following post-process steps are used to get continuous homogeneous regions with meaningful size:

1. Get all pixels with corresponding color bins labeled as 0, and labeled them as 0 in image domain;
2. Get all continuous (4-neighborhood) regions that are composed of pixels belonging to histograms labeled the same (uncommitted pixels are not taken into account);
3. All pixels in regions with size smaller than T_{size} (in our experiment, it is set to a very small proportion of the entire image size, say, 3.0×10^{-4}) are labeled as 0 (the third source of the uncommitted pixels); and
4. Label the left regions as $1, 2, \dots, m$, and all pixels in the labeled regions are labeled the same as their correspondent regions. Label the rest pixels the same as the label of their correspondent histogram.

There are two different methods in step 4: (a) get continuous regions such that all pixels with the same label are connected together; it will be followed by highest confidence first algorithm to get the final result; while (b) do not enforce the continuous constraint. It will be followed by a graph cuts algorithm to get the final result. The reason will be discussed later.

The post-process steps are used to get the initial segmentation result, which is composed of two parts, some continuous homogeneous regions with meaningful size, and some scatteredly distributed pixels labeled as zero, which are called uncommitted pixels.

Markov Random Field

Generally speaking, image segmentation assigns labels to each pixel. Let $S = \{(i, j) \mid 1 \leq i \leq width, 1 \leq j \leq height\}$ indexes pixels in an image with size $width \times height$. An image segmentation result is a realization of a random field $F = \{f_s \mid s \in S\}$, where f_s is a discrete random variable taking values in the set $L = \{1, \dots, m\}$.

A neighborhood system is defined by associating to each pixel $s \in S$ a symmetric set of neighbors N_s . A clique is either a single pixel or a set of mutually neighboring pixels. The set of all cliques is $C = \cup C_i$, where C_i is the set of cliques that contain i mutually neighboring pixels.

According to the Hammersley-Clifford theorem, the random field F is an MRF on S with respect to C if and only if its distribution on the state space is a Gibbs distribution:

$$P(f) = Z^{-1} \times e^{-U(f)/T}$$

$$Z = \sum_f e^{-U(f)/T}$$

where Z is a normalizing constant called the partition function. T is a constant called the temperature. $U(f)$ is the energy function, it is defined on all considered cliques as follows:

$$U(f) = \sum_{c \in C} V_c(f).$$

So the image segmentation problem (or generally, labeling problem) can be transformed into an energy minimization problem. The optimal solution is correspondent to the lowest energy state.

Cliques with different order have their own physical meaning. A zero-order clique contains only one pixel. The correspondent energy can be defined as fitness between the observed data and the model for assigned label. Information from feature space analysis can be smoothly integrated into this term. Higher order cliques usually contain information of interaction between neighboring pixels. It is very natural to assume that neighbor pixels tend to have the same label. For image segmentation, these terms reflect our preference of continuous regions. Local properties are integrated here.

We can rewrite the energy function in the following form by separating energy terms correspondent to cliques with different order:

$$U(f) = E_{smooth}(f) + E_{data}(f)$$

where E_{smooth} is the energy from high order cliques and E_{data} is the energy from zero-order cliques, which measures the distance between the assigned label f for each pixel and the local observation (the feature of this pixel). By minimizing the total energy, we can find the optimal labeling fitting the underlining model for each label, and also having the smoothness property.

More specifically, the smoothness term of energy function can be written in the following form:

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{p,q}(f_p, f_q)$$

where N is the set consisting of all neighboring pixel pairs (here, each clique contains at most two pixels). The simplest form of the potential energy prefers smoothness everywhere, (such as in the multilevel logistic distribution model). The result may be over-smoothed due to the poor result on the boundary. Some discontinuity preserving energy functions are proposed to deal with this problem (Yang, 2001; Grimson, 1985).

While the energy from observation usually takes the form:

$$E_{data}(f) = \sum_{p \in P} D_p(f_p)$$

Function D_p measures the fitness of the observed data in pixel p to the extracted model from label f_p . The specific form of this function depends on models of different applications. For example, in an image restoration problem, it can be defined as the color distance.

In the proposed framework, the feature domain watershed algorithm has already provided a good initial labeling state. Assume the site set (which contains all pixels in image) $S = S_0 \cup S'$, where S_0 is defined as the set of site which is labeled 0 in the initial state, and S' is defined as the set of site which has a non-zero label in the initial state. The problem left is how to decide the labels for sites in S_0 . In the following steps, two efficient energy minimization algorithms are applied to get suboptimal labeling state for the MRF, thus arriving at the final segmentation result.

Energy Minimization by Highest Confidence First (HCF)

Highest Confidence First (Chou, 1990)

HCF is a deterministic minimization algorithm. It introduces uncommitted labels and augments the label set to $L^+ = \{0, 1, \dots, m\}$. A label f_s is said to be uncommitted if $f_s = 0$ (or $s \in S_0$), or committed if $f_s \in L$ (or $s \in S'$).

For each pixel s , a potential energy function can be defined if we assigning it a label f_s :

$$E_s(f_s) = D_s(f_s) + \sum_{s' \in N_s} V_{s,s'}(f_s, f_{s'})$$

where D_s is the distance function and V is the smoothness energy term. N_s is the set of neighboring pixels of s . In the proposed framework, the second order neighborhood system (also known as the 8-neighborhood system) is used, which is commonly used in MRF-based algorithms (Mukherjee, 2002; Yang, 2001). $V=0$ if there is at least one pixel in c that is uncommitted. Therefore, in each step of the HCF algorithm, only committed pixels can propagate information to their neighbors, an uncommitted pixel will have no effect on its neighbors.

The stability of pixel s is defined as:

$$S_s(f) = \begin{cases} -\min_{l \in L, l \neq l_{\min}} [E_s(l) - E_s(l_{\min})] & \text{if } f_s = 0 \\ \min_{l \in L, l \neq f_s} [E_s(l) - E_s(f_s)] & \text{else} \end{cases}$$

$$l_{\min} = \arg \min_{l \in L} E_s(l)$$

The stability $S_s(f)$ measures the potential energy reduction for updating the label of a committed pixel or a possible future potential energy reduction for labeling an uncommitted one. This is called the confidence for updating this pixel. Highest confidence first means in each step, the pixel with lowest stability value will be updated.

At each step, only the least stable pixel will change its label or be committed. Suppose $k = \arg \min_s S_s(f)$ is the least stable pixel, we update f_k to f'_k as follows:

$$f'_k = \begin{cases} \arg \min_{l \in L} E_k(l) & \text{if } f_k = 0 \\ \arg \min_{l \in L, l \neq f_k} [E_k(l) - E_k(f_k)] & \text{else} \end{cases}$$

In each iteration step, we only need to update the stability for related pixels (the updated pixel itself and neighboring pixels). The iteration step stops until a pre-defined stop condition is satisfied.

Implementation Issues

In the proposed framework, the potential function is defined as follows:

$$D_s(f_s) = \alpha \rho^2(C(s), \mu(f_s))$$

where $C(s)$ is the color of pixel s , $\mu(l)$ is the average color of all pixels labeled l , which remains the same during the entire process because uncommitted pixels only take a small proportion. ρ is the Euclidean distance of two colors. ρ is a predefined constant.

Potential function is defined as:

$$V_{\{s, s'\} \in N}(f_s, f_{s'}) = \begin{cases} 0 & f_s = 0 \text{ or } f_{s'} = 0 \\ -\beta & f_s = f_{s'} \neq 0 \\ \beta & \text{else} \end{cases}$$

That is to say, for one clique, the potential energy is 0 if there is a pixel belonging to this clique that is uncommitted. This is derived from the definition of the HCF algorithm. The idea for assigning the $+\beta$, and $-\beta$ values for other cases is derived from the multi-level logistic (MLL) model, it is the simplest pair-wise smoothness constrain, tending to get smooth and continuous regions (Geiger, 1991).

The proposed framework has the following differences from the standard HCF algorithm:

1. The label updating process is executed only for uncommitted pixels. Label for committed pixels will not change once determined. The reason is that the watershed algorithm for color space has already obtained a satisfying result and need not be updated further. Reduction of computation load is another consideration.
2. $E_s(f_s)$ is set to infinity if f_s doesn't appear in its neighbor, that is, for an uncommitted pixel, it is only possible to be set to the labels which appear in its neighborhood, thus to enforce continuous regions.

The above modifications are designed to achieve the energy minimization efficiently. The entire process will stop until all pixels are committed.

The HCF algorithm of MRF in the proposed framework is summarized as follows:

1. For each region with label l , compute the average color;
2. Compute the stability value for all uncommitted pixels;
3. Update the uncommitted pixel with the lowest stability;
4. Update the stability for the left uncommitted pixels (note that this is only needed for pixels neighbor to the latest updated pixel until now); and
5. Go to step 3 until all pixels are committed.

The segmentation is completed after all the above steps. Pixels are clustered into regions according to their labels. The computation load is lower in comparison with other energy minimization algorithms (such as simulated annealing) because the uncommitted pixels only take a small proportion of the entire image, and only one of them is updated in each step.

Energy Minimization by Graph Cuts

A graph cuts algorithm to get a suboptimal solution for the energy minimization problem of a Markov random field has been developed by Boykov (2001). This algorithm can be applied directly in our framework. It is outlined as follows:

1. For $\alpha = 1, 2, \dots, m$: keep the label of sites in S' , update the label of sites in S_α with α -expansion.
2. Repeat step (1) until the total energy converges.

For given label state f and a label α , an α -expansion allows any set of pixels to change their labels to α . The optimal expansion move can find the optimal labeling state within one expansion move. By using graph cuts, the optimal expansion move can be found very efficiently.

The problem of finding optimal expansion can be reduced to a problem of finding max-flow (or min-cut) of a graph. The structure of the graph is shown in Figure 4 (Boykov, 2001). The two terminal nodes are α and $\bar{\alpha}$, every pixel in the image is correspondent to one node in the graph and they are connected with both terminal nodes (α and $\bar{\alpha}$). P_l ($l = 1, 2, \dots, m$) represents the set of nodes correspondent to pixels with current label l (for example: $q, r \in P_2$, because $f_q = f_r = 2$). Nodes representing neighboring pixels with the same label are connected by an edge (for example, q and r are representing a pair of neighboring pixels, so they are connect by an edge). Auxiliary nodes are added for each pair of neighboring nodes with different labels, and they are connected to both of the correspondent neighboring pixels and the sink node $\bar{\alpha}$ (for example, node a is added because $f_p \neq f_q$, and it is connected to node p , q and sink node $\bar{\alpha}$).

Figure 4. Illustration of graph structure for finding optimal expansion operation

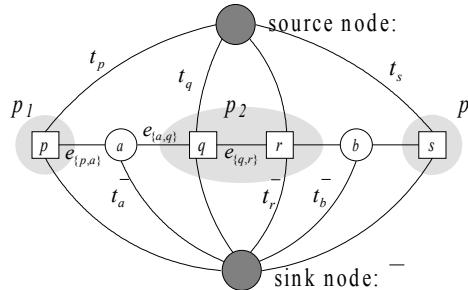


Table 1. Capacity settings for graph in Figure 4

edge	weight	for
t_p^α	∞	$p \in S', f_p \neq \alpha$
t_p^α	$D_p(\alpha)$	else
t_p^α	∞	$p \in P_\alpha$
t_p^α	$D_p(f_p)$	$p \notin P_\alpha$
$e_{\{p,a\}}$	$V(f_p, \alpha)$	$\{p, q\} \in N, f_p \neq f_q$
$e_{\{\alpha,q\}}$	V_{α, f_q}	
t_a^α	$V(f_p, f_q)$	$\{p, q\} \in N, f_p = f_q$
$e_{\{p,q\}}$	$V(f_p, \alpha)$	$\{p, q\} \in N, f_p = f_q$

The capacity of edges between the terminal node and node representing pixels are defined according to the observation energy. Capacities of other edges are based on a smoothness energy term. Please refer to Table 1 for more details. It is proved (Boykov, 2001) that under the constraint of metric property of smoothness energy, graph cuts based on the above graph structure correspondent to the optimal expansion move. Having the min-cut, a pixel is assigned label α if the cut separates this pixel from terminal α , otherwise, it just keeps its original label. In our experiment, the observation penalty energy is the same as in HCF algorithm; while the smoothness energy is 0 for a same label, and a fixed value \bar{a} for different labels. The infinity values in the first row of Table 1 are set to enforce the constraint that pixels in S' do not change their labels in this step.

If we apply the optimal expansion moves iteratively on every label until there is no energy reduction after one round of labeling update, a suboptimal global energy within a known factor of the global optimum can be achieved (Boykov, 2001). In our experiment, the energy usually converges in about two to five rounds of label updating.

There are no uncommitted pixels in the graph cuts algorithm; we can just assign random labels to uncommitted pixels for the initial state. The initial labeling state of pixels in S_0 does not have much influence on the final result. We do not have continuity constraint for initial labels. There are two reasons for this, one is that the graph cuts algorithm allows region discontinuity in the result, if the observation energy overwhelms the smoothness energy. Another reason is that it can greatly reduce the number of labels, which has great impact on the overall computation load (because each label will be updated iteratively in graph cuts, so the complexity is approximately linear to the number of labels).

Experiment Result and Applications

Comparison Between Different Color Spaces

Experimentations with a large variety of images are carried out. Some of them are shown in Figure 5.

Images (a-2) and (b-2) are the result after the watershed step on the L*a*b* color histogram. Each region is shown with the average color of pixels in it. The dominant colors

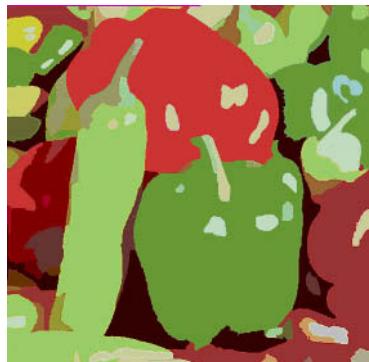
Figure 5. Result comparison between $L^*a^*b^*$ and RGB color space for PEPPERS and BEANS



(a-1) Original image
(PEPPERS 512 x 512)



(a-2) Labeling after Watershed
($L^*a^*b^*$ space)



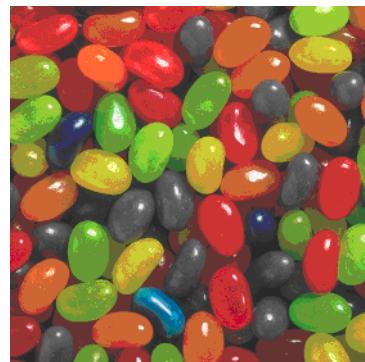
(a-3) Final result by HCF
($L^*a^*b^*$ space)



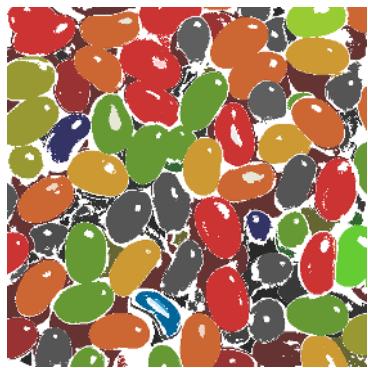
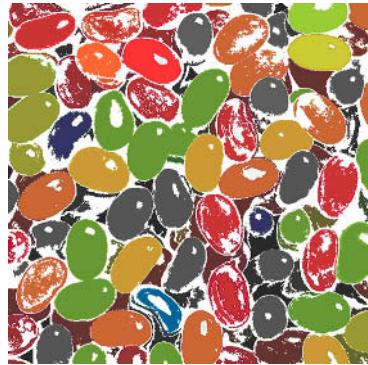
(a-4) Labeling after Watershed
(RGB space)



(a-5) Final result by HCF
(RGB space)



(b-1) Original image
(BEANS 512 x 512)

Figure 5. continued(b-2) Labeling after Watershed
($L^*a^*b^*$ space)(b-3) Final result by HCF
($L^*a^*b^*$ space)(b-4) Labeling after Watershed
(RGB space)(b-5) Final result by HCF
(RGB space)

are captured efficiently with the watershed algorithm in this color space. Results for some regions with nonuniform distributed colors (but with a smooth transition between colors) are satisfying, such as the largest green pepper region in PEPPERS and the salient bean regions in BEANS.

All uncommitted pixels are colored white here. It will be noticed that most uncommitted pixels have noisy color or a small uniformly colored region (small glistening border or shadow areas in BEANS, some shadow areas in PEPPERS). The results in this step are relatively coarse, many pixels are still uncommitted. The final segmentation results are shown in (a-3) and (b-3). The HCF algorithm for Markov random field is used to refine the coarse results by assigning the uncommitted pixels to existing regions. It is in some sense an ordered region growing process. Moreover, the results are promising. Uncommitted pixels are assigned reasonably.

Results based on RGB color space are shown in (a-4, 5) and (b-4, 5) for comparison. The entire procedure is the same as for $L^*a^*b^*$ color space. The RGB color space is also quantized uniformly into nearly the same number of meaningful (non-zero) histogram bins. The result is not very good, because the RGB color space is not consistent with human perception and the distance between perceptually similar colors is quite different (such as the green region in the PEPPERS and the over-segmented regions in BEANS).

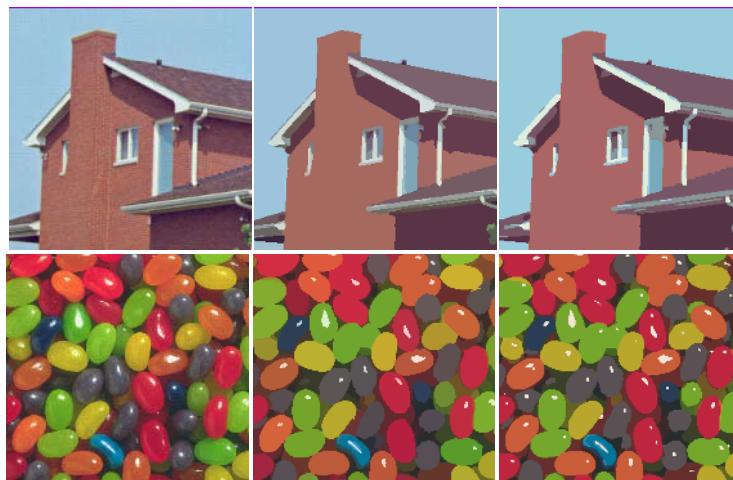
Comparison Between HCF and Graph Cuts

HCF and graph cuts have different continuity constraints. From Figure 6, we can see that HCF provides a better border. Because, in HCF, the observation is only determined by this continuous region, it contains more local information than a label in graph cuts, which may be distributed in the entire image. Graph cuts can find some salient small regions (such as the shining spot in BEANS), while HCF loses these regions because the strict constraint of region continuity. It is hard to compare the energy directly, because these two algorithms use different orders of neighboring system.

In terms of computational speed, the computation complexity of a proposed framework is not high. For a 256×256 color image in Pentium 2.8G PC, the watershed algorithm takes less than one second. Both graph cuts and the HCF algorithm take about four seconds. Graph cuts usually has 2-5 rounds of label updating, and the computational overload of the HCF algorithm depends on the number of uncommitted pixels.

An image retrieval system (Dai, 2005) is implemented based on image segmentation with the feature space clustering. Each image is described by two-level description. The watershed algorithm on color space provides a detailed description of image, and a merge

Figure 6. Result comparison between HCF and graph cuts for HOUSE and BEANS (the first column shows the initial images, the second column gives the results from HCF and the third column gives the results from graph cuts) applications



procedure is designed to get a rough description. This segmentation algorithm shows powerful color representation ability and helps to get satisfying retrieval results. It is also applied to medical image segmentation (Jiang, 2003). After some post-processing procedures, satisfying results can be achieved.

FUTURE TRENDS AND CONCLUSIONS

In this chapter a novel segmentation framework is proposed, which consists of two consecutive stages. Watershed on a 3-D histogram of $L^*a^*b^*$ color space is used in the stage of feature space analysis, providing a rough segmentation result. This technique can solve the problem caused by color clusters with irregular shapes in images and capture the dominant colors in images efficiently. Markov random field is followed to refine the previous coarse result to get the continuous regions. Two efficient energy minimization techniques are applied to get suboptimal results of MRF. Highest confidence first can enforce continuity properties, while graph cuts can avoid over-smoothness, and is more efficient.

Vision problems usually need clustering data in both feature and spatial spaces. However, a good cluster in feature space may not correspond to a continuous region in image space; researchers are paying more attention to clustering with joint feature and spatial coherence (Marroquin, 2003; Matas, 1995) and Markov random field is suitable to model both of them. Feature property can be integrated in first order energy, while spatial coherence can be modeled by higher order energy.

Graph cuts provide a very efficient energy minimization framework for MRF. Recently, this algorithm has been successfully applied in various vision problems, including: stereo (Kim, 2003), motion layer extraction (Xiao, 2004) and image segmentation (Rother, 2004) and achieving promising results. The central problem is how to construct an appropriate graph model. Iteration algorithms can also be integrated in this framework to improve the feature and spatial cluster simultaneously. It has been successfully applied in segmentation tasks (Zabih, 2004; Rother, 2004; Comaniciu, 2002). This is quite similar in spirit to the expectation-maximization algorithm (Dempster, 1977), where feature model and spatial coherency are updated by turns. Very promising results are achieved under this framework.

ACKNOWLEDGMENTS

This work has been supported by the Grant NNSF-60172025 and SRFDP-20050003013.

REFERENCES

- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48, 259-302.
- Beucher, S. (1982). Watersheds of functions and picture segmentation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 1928-1931).

- Boykov, Y., & Kolmogorov, V. (2001). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Proceedings of International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* (pp. 359-374).
- Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 1-18.
- Carson, C., Belongie, S., Greenspan, H., & Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 1026-1038.
- Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. (2001). Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12), 2259-2281.
- Chou, P. B., & Brown, C. M. (1990). The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3), 185-210.
- Comaniciu, D., & Meer, P. (2002). Mean-shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603-619.
- Dai, S. Y., & Zhang, Y. J. (2005). Unbalanced region matching based on two-level description for image retrieval. *Pattern Recognition Letters*, 26(5), 565-580.
- Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1-38.
- Geiger, D., & Girosi, F. (1991). Parallel and deterministic algorithms for MRFs: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), 401-413.
- Geiger, D., & Yuille, A. (1991). A common framework for image segmentation. *International Journal on Computer Vision*, 6(3), 227-243.
- Geraud, T., Strub, P. Y., & Darbon, P. Y. (2001). Color image segmentation based on automatic morphological clustering. In *Proceedings of the IEEE International Conference on Image Processing* (Vol. 3, pp. 70-73).
- Greig, D., Porteous, B., & Seheult, A. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2), 271-279.
- Grimson, W. E. L., & Pavlidis, T. (1985). Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 30, 316-330.
- Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37.
- Jiang, K., Liao, Q. M., & Dai, S. Y. (2003). A novel white blood cell segmentation scheme using scale-space filtering and watershed clustering. In *Proceedings of the Second IEEE International Conference on Machine Learning and Cybernetics* (Vol. 5, 2820-2825).
- Kim, J., Kolmogorov, V., & Zabih, R. (2003). Visual correspondence using energy minimization and mutual information. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1033-1040).
- Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), 147-159.

- Li, S. Z. (1995). *Markov random field modeling in computer vision*. New York: Springer.
- Marroquin, J., Santana, E., & Botello, S. (2003). Hidden Markov measure field models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11), 1380-1387.
- Matas, J., & Kittler, J. (1995). Spatial and feature space clustering: Applications in image analysis. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns* (pp. 162-173).
- Mukherjee, J. (2002). MRF clustering for segmentation of color images. *Pattern Recognition Letters*, 23(8), 917-929.
- Park, S. H., Yun, I. D., & Lee, S. U. (1998). Color image segmentation based on 3-D clustering: Morphological approach. *Pattern Recognition*, 31(8), 1061-1076.
- Patras, I., Hendriks, E. A., & Lagendijk, R. L. (2001). Video segmentation by MAP labeling of watershed segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), 326-332.
- Pauwels, E. J., & Frederix, G. (1999). Finding salient regions in images: Nonparametric clustering for image segmentation and grouping. *Journal of Computer Vision and Image Understand*, 75(1/2), 73-85.
- Roerdink, J., & Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamental Informatica*, 41, 187-228.
- Rother, C., Kolmogorov, V., & Blake, A. (2004). Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH'04)*, 23(3), 309-314.
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), 583-598.
- Wang, D. (1998). Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on Circuits System Video Technology*, 8(5), 539-546.
- Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1101-1113.
- Xiao, J., & Shah, M. (2004). Motion layer extraction in the presence of occlusion using graph cut. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 972-979).
- Yang, X., & Liu, J. (2001). Unsupervised texture segmentation with one-step mean shift and boundary Markov random field. *Pattern Recognition Letters*, 22(10), 1073-1081.
- Zabih, R., & Kolmogorov, V. (2004). Spatially coherent clustering using graph cuts. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (Vol. 2, pp. 437-444).

Chapter XI

Optimizing Texture Primitive Description, Analysis, Segmentation, and Classification Using Variography

Assia Kourgli, U.S.T.H.B. University, Algeria

Aichouche Belhadj-Aissa, U.S.T.H.B. University, Algeria

ABSTRACT

There are many approaches dealing with various aspects of texture analysis, segmentation and classification. One of the characteristics among most of these approaches and those using neighborhood is that they require applying a template to a given image, pixel by pixel, to yield a new image. Although the selection of an appropriate window size is critical and affects directly the results obtained, it is often done arbitrarily. We present, in this chapter, a new approach based on the concept of variography to achieve the selection of the optimal window. We develop some direct applications including textural primitives description, mathematical morphology, textural segmentation and textural classification. Promising results are obtained and presented.

INTRODUCTION

Texture plays an important role in visual information processing since it provides useful information about shape, orientation and depth of the objects. To develop a texture analysis system capable of dealing with all aspects of texture structure is a very difficult visual information processing task because of the complex nature of texture, and is, as yet, an unsolved problem. It is important to use properties of texture and to understand how the human visual system works for texture discrimination and grouping necessary to the analysis.

There are many approaches dealing with various aspects of texture analysis, segmentation and classification (Reed, 1993; Sharma, 2001; Tuceyran, 1998). One of the characteristics among most of these approaches and those using neighborhood is that they require applying a template to a given image, pixel by pixel, to yield a new image. Successful textural analysis relies on the careful selection of adapted window size because it determines the amount of information that has to be evaluated. On one hand, window size should be large enough to provide as many details as possible about texture pattern and, on the other hand, be small enough to take only the relevant information. So, the selection of an appropriate window size is critical because it affects directly the results obtained.

Our purpose in this chapter is to achieve the selection of the optimal window using a concept belonging to geostatistics (Cressie, 1993). Indeed, geostatistics is the analysis of spatially continuous data. It treats geographic attributes as mathematical variables depending on their locations. One of the central techniques among geostatistical approaches is the variogram, which describes to what extent spatial dependence exists between sample locations. We show that the variogram measure provides a description of the placement rules and the unit patterns of texture. Indeed, local minima of the variogram measure can be used to describe size, shape, orientation and placement rules of unit patterns of a texture (Kourgli, 2000). Using hole-effects, we can determine the size and shape of the optimum window and obtain a description of texture primitives. The second application is mathematical morphology, where we apply variography to find the optimal structuring element applied with morphological operators. The results obtained are evaluated using a contour enhancement algorithm. The others applications are textural segmentation and textural classification. We propose to use some further features extracted from the variogram representation (range of influence, variogram values, etc.) in a segmentation scheme. The segmentation and classification processes are performed on Brodatz' textures (Brodatz, 1966) to be validated. Then, they are performed on photographic urban images for the identification of built-up environment and cover lands. Promising results are obtained and presented.

In section 1, we review research in the area of texture analysis, texture segmentation and classification. One of the key aims of literature review is to learn of what other research has taken place in finding the optimal window size for texture description. In section 2, we provide a detailed methodology of how we aim to apply variography to texture analysis. In section 3, we describe the processes used for texture primitives detection, mathematical morphology, texture segmentation and classification. Some algorithms are presented to show in more detail the processes that are applied to images before final results are obtained. These algorithms can be easily programmed, with some additional reading, using C++. We also show the results of detailed experiments with

Brodatz' textures and photographic images. The chapter concludes with a final summary. As a part of this chapter, we discuss the importance of our results to texture analysis research. We also suggest how our results can be further improved. A bibliography is provided at the end for further reading.

BACKGROUND AND RELATED WORK

As a fundamental step in the understanding and description of natural imagery, texture analysis is one of the most important techniques used in analysis and classification of images representing repetition or quasi-repetition of fundamental elements. Texture is very difficult to define. This difficulty is demonstrated by the number of different texture definitions attempted by vision researchers. It is simply defined as the relationships between gray levels in neighboring pixels that contribute to the overall appearance of an image. It can also be viewed as a global pattern arising from the repetition, either deterministically or randomly, of local sub-patterns.

A number of techniques for texture analysis and discrimination have been proposed and achieved considerable success, although generally under well defined and rather limited operating conditions (Zhang, 2003). Approaches to texture analysis are usually categorized into structural, statistical, model based and transforms methods. The structural approach is based on the description of unit patterns and their placement rules by using geometrical methods such as Voronoi Tessellation, mathematical morphology, image filtering, tree grammars, etc. In the statistical approach, the aim is to characterize the stochastic properties of the spatial distribution of gray levels in an image by estimating first and higher order statistics from a local neighborhood by auto correlation functions, frequency domain analysis, edge operators, grey level co-occurrence matrices, grey level run length, grey level sum and difference histograms and filter masks (Haralick, 1986). While transforms methods, such as Fourier, Gabor and wavelets transforms (Arivazaghan, 2003), represent an image in space whose co-ordinate system has an interpretation which is closely related to the characteristics of texture (frequency, size, direction, etc.). In the early eighties, there had been a new interest in model-base techniques using fractal and stochastic models (Autoregressive, Markov, Gibbs, Gaussian, etc.) which attempted to interpret an image by use of, respectively, generative and stochastic models (Materka, 1998).

In Materka (1998), Ohanian (1992), Reed (1993), and Tuceyran (1998), concise surveys of many recent texture segmentation and features extraction techniques are presented and compared. Recently, Singh (2002) compared some traditional, and some fairly new techniques of texture segmentation—the MeasTex and VisTex benchmarks—to illustrate their relative abilities. The methods considered include autocorrelation, co-occurrence matrices, edge frequency, Laws' masks, run length, binary stack method (Chen, 1995), texture operators (Manian, 2000) and texture spectrum (Kourgli, 1997). These different approaches are evaluated using the linear classifier and the nearest neighbor. In her thesis, Sharma (2001) studied and compared four image segmentation algorithms (fuzzy c-means clustering, histogram based thresholding, region growing and split and merge), five texture analysis algorithms (autocorrelation, co-occurrence matrices, edge frequency, Laws' masks and primitive run length) and two classifiers (linear classifier and nearest neighbor classifiers). She concluded there were no leading texture

approaches because the results obtained depend on the segmentation technique used, combined with the texture analysis approach chosen. In Clausi (2004), the discrimination ability of texture features derived from Gaussian Markov random fields and grey level co-occurrence probabilities are compared and contrasted. More specifically, the role of window size in feature consistency and separability, as well as the role of multiple textures within a window are investigated. On the one hand, co-occurrence matrices are demonstrated to have improved discrimination ability relative to MRFs with decreasing window size, an important concept when performing image segmentation. On the other hand, they are more sensitive to texture boundary confusion than GMRFs.

The problem pointed out by this latest study is common to most segmentation techniques. Indeed, they require the selection of the optimal window size yielding to the best rate of recognition. Most texture measures utilize a moving array of cells with a variety of mathematical measures to derive texture values for the center cell of the moving array. It is unclear how the size of the moving window influences classification accuracy; it can have a positive or negative effect depending on the intended application. Generally, the solution found is adapted to the specific problem addressed. It seems likely that the influence of various window sizes may be a function of surface characteristics and spatial resolutions (Haack, 2000). In his thesis, Glotefelty (1999) examined the relationship between texture and scale, and its effects on image classification. He created a model that automatically selected windows of optimal size according to the location of a pixel within a land cover region and the texture of the surrounding pixels. Large windows were used to get a representative sample of within-class variability in the interior of these regions. Smaller windows were used near the boundaries of land cover regions in order to reduce edge effect errors due to between-class variability. This program was tested using a Maximum Likelihood classification scheme against spectral data and texture from fixed-size windows to determine if there were any improvements in classification accuracy. Images obtained by incorporating the optimal size window program were superior in accuracy to the other ones. Puig (2001) described an algorithm for determining the optimal window size for a texture feature extraction method in order to maximize both its discrimination and segmentation capabilities. The algorithm applies a multi-criterion decision technique in order to select the best window size based on two criteria: normalized number of votes and average discrimination percentage. Grigorescu (2003) investigated a texel identification method based on the search of the smallest window through which the minimum number of different visual patterns is observed when moving the window over a given texture. Such a window has the property of minimizing Rényi's generalized entropies. In Khotanzad (2003), the segmentation algorithm used relies on scanning the image with a sliding window and extracting texture and color features from each window. The optimum window size is obtained by sweeping the image with varying window sizes, and choosing the smallest one out of at least two consecutive window sizes that produce the same number of clusters. In Kourgli (2000), we have shown that the variogram measure provides a description of the placement rules and the unit patterns of texture. Indeed, local minima, computed from the variogram measures in different directions, can be used to describe size, shape, orientation and placement rules of unit patterns of a texture and thus exploit these features to customize windows for use in texture analysis. The variogram which is commonly used in geostatistics is essentially a "variance of differences" in the values as a function of the separation distance. This variance therefore changes as the separation distance increases, where repetitive

structures are described as hole-effects. In semivariance analysis, the semivariogram parameters (range, sill and nugget) have been applied first to remote sensed data (Atkinson, 2000; Berberoglu, 2003; Chica-Olmo, 2000; Ji, 2004; Maillard, 2000). Recently, variogram analysis has successfully been exported from geostatistics to other fields, for example, ecology (Daley, 1999) and epidemiology (Zucca, 1997) that make use of spatially referenced variables including edge detection in biomedical images, classification of fingerprints, default detection and urban segmentation (Hamami, 2001; Kourgli, 2004; Mardia, 1997).

VARIOGRAPHY

Spatial continuity is the cornerstone of geostatistics, which may be loosely defined as the statistical study of spatial correlation. In its ability to provide a quantitative measure of the intuitive sense that points, which are closer together, are more related than points farther apart, geostatistics offers a great form of validation to environmental science (Atkinson, 2000; Chandler, 2004).

Geostatistics

Geostatistics is built upon concepts of probability theory, in particular, the regionalized variable. Technically, a random variable is a function defined on a sample space. There are two minimum conditions on the data, which are sufficient for geostatistical analysis, as follows: The data are intrinsically stationary and the spatial correlation can be defined with a mathematical function, or model. The stationarity condition assumes that the values in the data set represent the same statistical population. That is, the property measured is stationary, or stable, over the area measured. It is required to ensure that the spatial correlation may be modeled with an appropriate function (that is, a positive definite function) and states that the expected value, noted E, which may be considered the mean of the data values, is not dependent upon the distance 'd' separating the data points. Mathematically, this assumption states that the expected value of the difference between two random variables is zero:

$$E[Z(x+d) - Z(x)] = 0 \text{ for all } x, x + d \quad (1)$$

where, $Z(x), Z(x+d)$ = random variables, x = sampled location and d = distance between sampled locations.

Variogram

Probability theory defines the variogram (sometimes referred to as the semivariogram) in terms of the variance of the difference between two random variables (Cressie, 1993; Matheron, 1963). The term variance refers to the measure of dispersion within a random variable (that is, all possible values).

The variogram may be defined as follow:

$$2\gamma(d) = E\{[Z(x) - Z(x+d)]^2\} \text{ for all } x, x + d \quad (2)$$

So, semi-variograms are essentially graphs measuring the difference between grade values relative to the distance separating them in a particular orientation. They provide a description of the similarity or dissimilarity between pairs of values as a function of their separation vector ' d '. For certain applications, such as earth sciences, remote sensing or image processing, whose data sets contain huge amounts of closely spaced and regularly gridded information, summarizing the pattern of spatial continuity is, by itself, an important goal.

Numerically, the variogram function is calculated in units of distances. It is typically estimated by the "sample" or experimental variogram:

$$2\gamma(d) = \frac{1}{N(d)} \sum_i [Z(x_i) - Z(x_i + d)]^2 \quad (3)$$

where d = the inter-sample spacing distance (in pixels), $N(d)$ = the number of gray level pairs within an image and $Z(x_i)$, $Z(x_i + d)$ = the Gray level pairs.

The result is a variogram value, which is plotted against distance.

Variogram Representation and Features

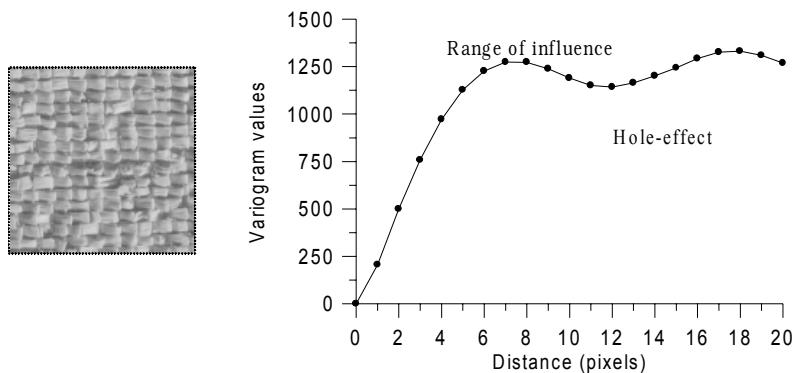
The basic philosophy behind a semi-variogram is that at a particular orientation within a given domain, as the distance between sample points increases, so does the expected difference between the grade values. This relationship is illustrated in Figure 1, which shows the variogram of the raffia texture (Brodatz, 1966).

A typical semi-variogram has three characteristics which enhance the understanding of the phenomenon behavior. These are as follows:

Range of Influence: The distance between the samples at which the semi-variogram appears to level off (7 pixels for raffia texture).

Nesting Structures: repetitive structures are described as hole-effects (12 pixels for raffia texture).

Figure 1. Variogram values versus distance in the horizontal direction for the raffia texture



Sill: The level of variability that indicates that there is no further correlation between pairs of samples.

The range can be used as a measure of spatial dependency, or homogeneity, whereas the sill reflects the amount of spatial variability. Nesting structure on the semi-variogram defines the separation distance up to which the samples can be considered repetitive. To gain a more complete understanding of the overall behaviour of the texture, it is necessary to plot a selection of semi-variograms at different orientations for various texture patterns.

Two kinds of experiments were performed. The first kind was designed to show that the variogram measure did indeed vary with the amount of structure in the texture pattern, while the second involved the comparison of variogram and χ^2 measures.

To show the relation between the variogram measure computed over an image and the degree of structure present in the underlying texture, we simulated a texture pattern of 64×64 pixels. This texture is illustrated in Figure 2 and is composed of pattern units sized 16×16 pixels and texture primitives composed of sub-patterns sized 4×4 .

The variogram $\gamma(d)$ values (various distances over the four angles 0° , 45° , 90° and 135°) were computed for the simulated image (Figure 2), and the result is plotted against distance. The experimental variogram, shown in Figure 3a, is fitted through the plotted

Figure 2. Simulated texture

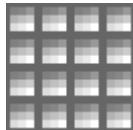
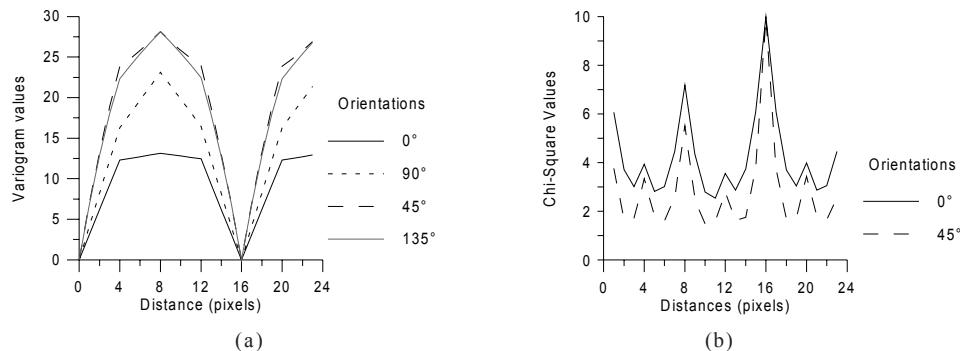


Figure 3. Graphs of (a) $\gamma(d)$ measures, and (b) χ^2 measures versus distance at the four orientations for the simulated texture of Figure 2



points to assess the spatial correlation. Variations in the variogram measure can be observed for a single image as a function of the spatial relation ‘d’. For values of distance ‘d’ that capture texture structure very well, variogram values will be low (hole-effect), as expected; $\gamma(d)$ values decrease when the inter-sample spacing distance ‘d’ is equal to texture primitives size, and minima occur for $d = 16, 32, 48$ and 64 pixels. A further property of the variogram measure is that the range of influence is half the size of texture pattern ($d=8$ pixels). We note, also, that the variogram behaviour changes each 4 pixels’ displacement, so the behaviour of the variogram is related to the size of the primitive and its sub-patterns. Hence the above observations tell us that the variogram measure can be used for texture description.

Comparing Variogram to χ^2 Measure

To further corroborate our observations, we compared the variogram measure and the χ^2 measure proposed by Zucker (1980), who used a statistical approach to find those spatial relations that best capture the structure of textures when grey-level, co-occurrence matrices are used (Haralick, 1986). The model adopted is based on a χ^2 (chi-square) measure of independence of the rows and columns of the co-occurrence matrix. The measure of independence is formulated by interpreting the co-occurrence matrix and its elements P_{ij} as a contingency table. This leads, after some algebra, to the following expression:

$$\chi^2 = N \left(\sum_{i=1}^m \sum_{j=1}^n \frac{p_{ij}^2}{r_i c_j} - 1 \right) \quad (4)$$

$$\text{where } r_i = \sum_{j=1}^n p_{ij} \text{ and } c_i = \sum_{i=1}^m p_{ij}.$$

Figure 3b shows the results obtained by applying the χ^2 measure for the texture simulated above (Figure 2).

Similar variations in the variogram measures and the χ^2 measures can be observed for a single image (Figure 3a and Figure 3b), for values of ‘d’ that capture texture structure very well; variogram measures will be low while χ^2 measures will be high. However, variogram measures do not involve computation of the grey level co-occurrence matrix, so it takes less time computation and they can be applied for any textural analysis that involves the choice of optimal window. In the following section, we focus on the use of variography in texture primitives’ detection, textural segmentation and classification.

APPLICATIONS OF VARIOGRAPHY TO TEXTURE ANALYSIS

The purpose of the following section is to provide an optimal description of texture primitives and thus to customize window sizes (structuring element shape) for use in mathematical morphological texture analysis and textural segmentation and classification.

Extracting Texture Primitives

In texture analysis the first and most important task is to extract texture features which most completely embody information about the spatial distribution of intensity variations in an image. In order to evaluate the performance of the variogram measure for texture characterization, we use the following Brodatz textures: raffia, woollen cloth, herringbone weave and wood grain (Figure 4). The texture images were taken from *USC-SIPI Image Database* and are of size 128×128 with 256 gray levels.

The variogram values (Figure 5a, Figure 5b, Figure 6a, and Figure 6b) were computed for these textures.

For natural texture, the variogram measure is never equal to zero but local minima represent candidates' points from which the structure size can be computed. So, the minimum variogram values being significantly different from zero indicates that the replication process and/or the unit patterns are not always the same. Figure 5a shows that the points where variogram values are minimum (hole-effect) correspond to inter-sample spacing distance values ' d ', which are integer multiples of 12 pixels for the horizontal direction and of 8 pixels for the vertical one. Hence, raffia texture is composed of primitives whose size is 12×8 .

Figure 4. Textures: (a) raffia, (b) woollen cloth, (c) herringbone weave, (d) wood grain

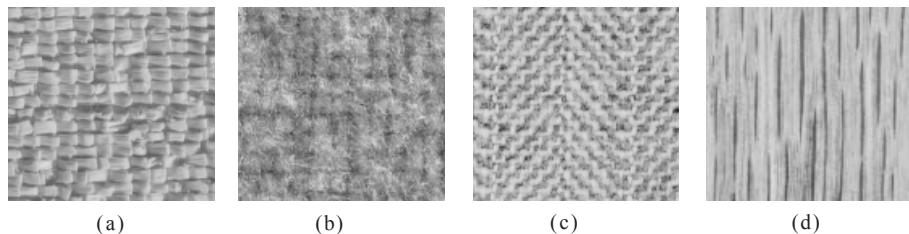


Figure 5. Graphs of $g(h)$ measures versus distance at the four orientations: (a) raffia texture, (b) herringbone weave texture

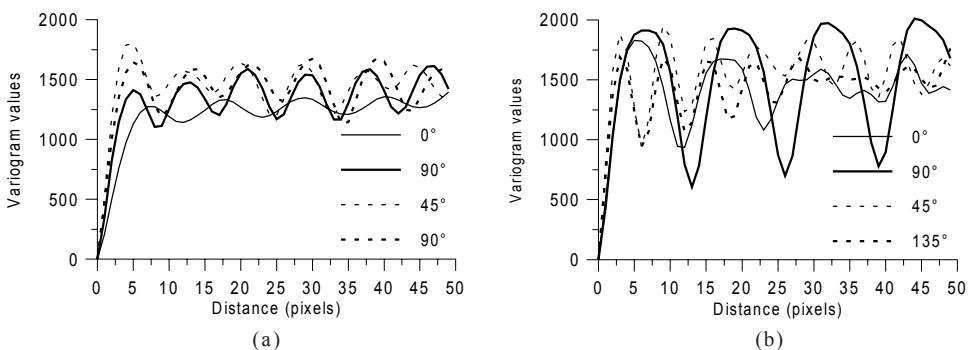
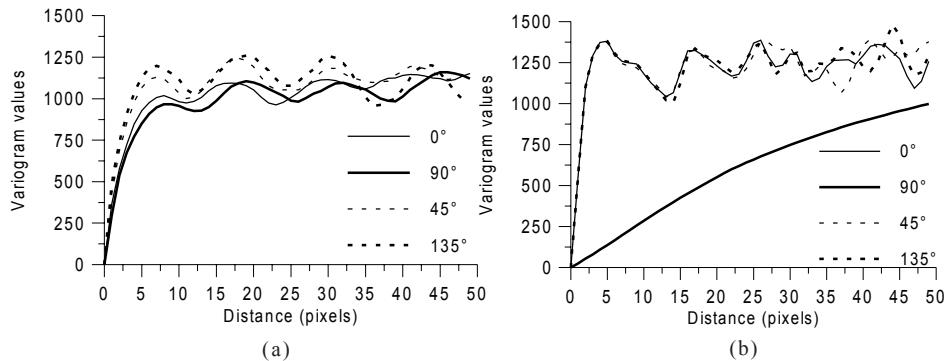


Figure 6. Graphs of $g(h)$ measures versus distance at the four orientations: (a) woolen cloth texture, (b) wood grain texture



The plots shown in Figure 5b indicate that we have a structure repetition every six pixels in both diagonal directions (45° and 135°) for herringbone weave texture. We can also estimate the size of the unit pattern of woollen cloth texture using the graphs of variogram measures. The plots shown in Figure 6a indicate that the unit pattern is a square of 12 by 12 pixels. While for wood grain texture, the nonappearance of hole-effect in the vertical direction (Figure 6b) indicates the existence of a vertical structure. These four examples demonstrate how the variogram measure can be used to compute the size and orientation of unit patterns. Other textures from Brodatz's album have also been analyzed. The textures examined include grass, beach sand, pigskin and pressed calf leather.

Variography and Mathematical Morphology

Both linear convolution and morphological methods are widely used in texture image processing. One of the characteristics common between them is that they both require applying a template to a given image, pixel by pixel, to yield a new image. In the case of convolution, the template is linear and is usually called a convolution window or mask, while in mathematical morphology, it is referred to a structuring element. Mathematical morphology involves the study of the different ways in which a structuring element interacts with a given set, modifies its shapes and extracts the resultant set (Lee, 1997). Structuring elements can vary greatly in their weights, sizes and shapes, depending on the specific applications (Sussner, 1997).

So, the structuring element is used as a tool to manipulate the image using various operations. The basic ones are erosion and dilatation (Jackway, 1997). We use the notation of Krishnamurthy et al. (1994) to introduce morphology on functions.

Suppose that a set A in the Euclidean N -space (E^N) is given. Let F and K be $\{x \in E^{N-1} | \text{for some } y \in E, (x, y) \in A\}$, and let the domains of the gray-scale image be f and the structuring element (kernel/template) be k , respectively.

The dilatation of f by k , which is denoted by $f \oplus k$, is defined as:

$$(f \oplus k)(x, y) = \max \{f(x + m, y + n) + k(m, n)\} \quad (5)$$

for all $(m, n) \in K$ and $(x + m, y + n) \in F$.

The erosion of f by k , which is denoted by $f \ominus k$, is defined as:

$$(f \ominus k)(x, y) = \min \{f(x + m, y + n) - k(m, n)\} \quad (6)$$

for all $(m, n) \in K$ and $(x + m, y + n) \in F$.

Based on these operations, closing and opening are defined. The closing operation is a dilatation followed by an erosion with the same structuring element while the opening operation is an erosion followed by a dilatation.

The selection of a structuring element k used by the dilatation and erosion functions is very important to the system, as this determines the manner in which the individual objects are supposed to be connected.

So, mathematical morphology is a structural method of processing images according to the images' topological properties. It has been successfully used in many applications including object recognition, image enhancement, texture analysis and industrial inspection (Verly, 1993). Several adaptive techniques have been used for finding optimal morphological structuring elements. Some techniques use neural networks and fuzzy systems (Lee, 1997; Verly, 1993). We suggest using variography to optimize the shape and size of structuring elements to fit the shape of the unit patterns that form a texture.

The plots shown in Figure 5b indicate that we have a structure repetition every six pixels in both diagonal directions (45° and 135°) for herringbone weave texture, hence, we choose a structuring element k_1 whose size is half the distance for which we have a nesting structure (3×3 pixels) and shape with respect to the main directions (both diagonal directions) of the texture. While the plots shown in Figure 6a for woolen cloth indicate that the unit pattern is a square of 12×12 pixels, we choose a square structuring element k_2 whose size is 6×6 pixels. And for raffia texture, the corresponding structuring element k_3 is a rectangle of 6×4 pixels.

We obtain the following structuring elements k_1 , k_2 and k_3 :

$$k_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad k_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad k_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (7)$$

Erosion, dilatation, opening and closing, with the above structuring elements were applied to the texture images shown in Figure 4. The target images for these operations are shown in Figures 7, 8, and 9.

Figures 7b and 7c show that the results of erosion and opening of raffia texture (Figure 7a) by the structuring element k_1 allow enhancing unit patterns which look

Figure 7. (a) Raffia texture; results of (b) erosion; (c) opening

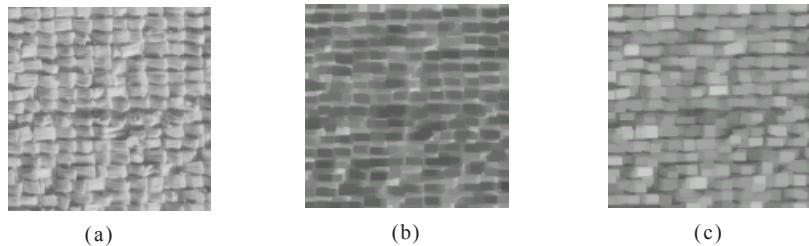


Figure 8. (a) Woolen cloth texture; results of (b) opening; (c) closing

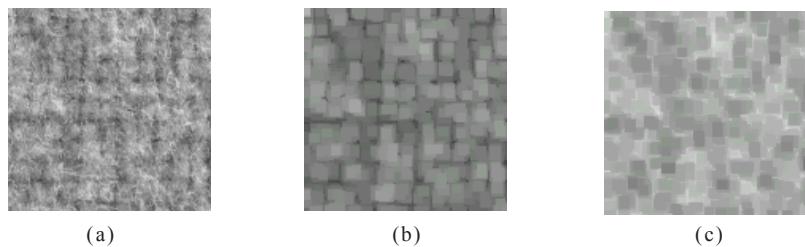
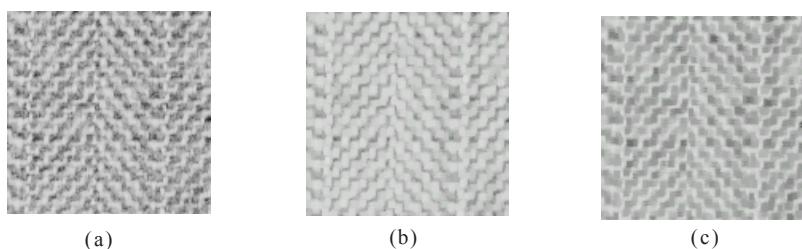


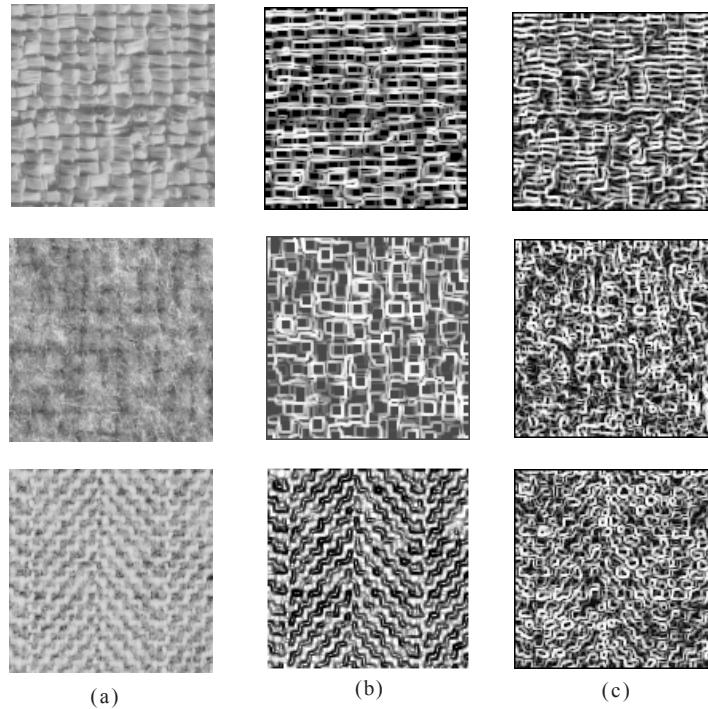
Figure 9. (a) Herringbone weave texture; results of (b) dilatation; (c) closing



homogenous, while contours are well preserved. Similar results (Figure 8b, Figure 8c, Figure 9b, and Figure 9c) are obtained for woolen cloth (Figure 8a) and herringbone weave textures (Figure 9a); we note that the shape and the boundaries of the unit patterns are enhanced; hence, the structuring elements found are well adapted.

To assert these observations, we applied a contour enhancement algorithm (Sobel filter, Coquerez, 1995) to the original texture images and to the morphological transforms. The results obtained are illustrated by Figure 10, where the original images are repre-

Figure 10. (a) Original images, (b) contour enhancement applied to the eroded images using the optimal structuring elements, (c) contour enhancement applied to the eroded images with an arbitrary structuring element



sented in the first column (Figure 10a) the second column's images (Figure 10b) illustrate the results of contour enhancement applied to the eroded texture using the optimal structuring elements found (Equation 7) and the last one (Figure 10c) represents the results of contour enhancement applied to the eroded texture using an arbitrary structuring element (a 3×3 size square).

So, good results can be obtained when the variogram measure is used to compute the size and shape of structuring elements. Indeed, texture primitives (Figure 10b) are well characterized and their micro-contours are preserved with the optimal structuring elements.

Applying Variography to Textural Segmentation

Texture segmentation is a difficult problem because we often don't know *a priori* what types and how many textures exist in an image. It consists of splitting an image into regions of uniform texture. Therefore, while a variety of methods exist that have been demonstrated to provide good segmentation, it remains quite difficult to produce a

generally usable segmentation scheme since many of the details of the process must be specifically chosen for the particular discrimination task to be considered.

Indeed, textural segmentation is usually performed by applying two stages. In the first one textural features are extracted. These features are used in the second stage to segment the image. Despite the fact that the windows chosen in both stages directly affect the quality of the final segmentation, most texture segmentation algorithms that have been proposed in the literature define those sizes experimentally. Hence, determining a suitable size for those windows is an open problem.

The two general approaches to performing texture segmentation are analogous to methods for image segmentation: region-based approaches or boundary-based approaches.

In a region-based approach, one tries to identify regions of the image which have a uniform texture. Pixels or small local regions are merged based on the similarity of some texture. The regions having different textures are then considered to be segmented regions. These methods have the advantage that the boundaries of regions are always closed and therefore, the regions with different textures are always well separated. In many region-based segmentation methods, one has to specify the number of distinct textures present in the image in advance. There are many techniques related to this approach including clustering (Jain, 1999), Gabor filters (Manthalkar, 2003), Wavelet transform (Hsin, 2002; Liapsis, 2004), Statistical geometrical features (Dai, 2004), generative model (Stainvas, 2003), Watershed algorithm (Malpica, 2003), association rules (Rushing, 2001), and so forth.

The boundary-based approaches, including independent component analysis (Chen, 2002), Markov fields (Li, 2003) and Gabor filters (Ji, 2004) are based upon the detection of texture differences in adjacent regions. In this method, one needn't know the number of textured regions in the image in advance. However, the boundaries may have gaps, and two regions with different textures are not identified as separate closed regions.

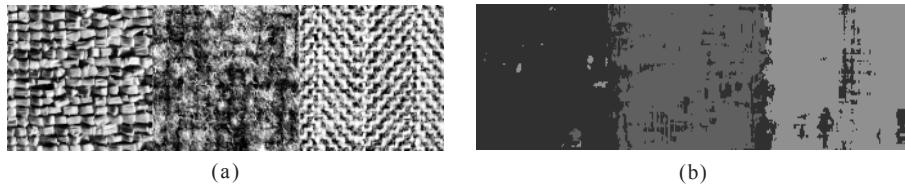
We performed a segmentation scheme related to region-based approaches using the textural information provided by the variogram. We exploited a discriminating property of the variogram measure which is the range of influence. For the determination of the window size, we took account of the fact that there is a trade-off between the application of too large and too small a window. On the one hand, a large window leads to a variogram encompassing adjacent textures. On the other hand, within a small window, the range of the variogram may be incorrectly estimated.

The following is an example of the kind of results that can be obtained (Figure 11a and Figure 11b). The procedure of segmentation is described as follows:

- Select from each texture image a sample of size $M \times M$ pixels. The window size chosen must be large enough to allow computation of the variograms with a reasonable number of lags to detect range of influence or hole-effect.
- Calculate the variogram measure $\gamma_{i,\theta}(d)$ ($i=1, 2, 3$) for each sample according the four main directions ($q=0^\circ, 90^\circ, 45^\circ$ and 135°) and determinate the “range of influence” $R_{i,\theta}(d_{i,\theta})$ for each one:

$$R_{i,\theta}(d_{i,\theta}) = \{\gamma_{i,\theta}(d)/\text{first max of } \gamma_{i,\theta}(d)\} \quad (8)$$

Figure 11. (a) Mosaic textures (raffia, woollen cloth and herringbone weave), (b) segmented textures



- Scan the mosaic texture using a window of $M \times M$ pixels with a step of one pixel in the row and column directions, and calculate the “range of influence” of the variogram plots $R_\theta(d_\theta)$ for each window.
- Calculate the absolute difference between the four “range of influence” plots $R_\theta(d_\theta)$ and distances d_θ ($\theta = 0^\circ, 90^\circ, 45^\circ$ and 135°) of each window and the ones $R_{i,\theta}(d_{i,\theta})$ and $d_{i,\theta}$ of each sample:

$$D_\theta(i) = (R_\theta(d_\theta) - R_{i,\theta}(d_{i,\theta}))^2, \quad i=1,2,3. \quad (9)$$

$$E_\theta(i) = (d_\theta - d_{i,\theta})^2, \quad i=1,2,3. \quad (10)$$

The central pixel of the window considered will be assigned to the class ‘i’, such most of $D_\theta(i)$ and $E_\theta(i)$ ($\theta = 0^\circ, 90^\circ, 45^\circ$ and 135°) are minimum among all the $D_\theta(i)$ and $E_\theta(i)$ for $i=1, 2, 3$.

The method developed then produces an image partition chosen (Figure 11a) according to the textural feature; we note that the three textures can be easily distinguished. The result, illustrated in Figure 11b, shows 97.97% correct classification for raffia, 82.15% for woolen cloth and 89.68% for herringbone weave.

Textural Classification

Texture classification involves deciding what texture category an observed image belongs to. In order to accomplish this, one needs to have an *a priori* knowledge of the classes to be recognized. Still, textural image classification is far from being widely used in automated object identification. This is due to the fact that there is no general rule as to what size of a sliding window should be chosen for the estimation of textural parameters. Usually, they are derived from a moving kernel of a fixed size, where window size is chosen experimentally in the pre-processing phase, and greatly depends on the image to be classified.

Franklin et al. (1996) proposed to use a geographic window, where dimensions of a moving kernel are customized and are determined by a variogram range. Indeed, Franklin generated geographic windows corresponding to the scale of observation to provide

forest inventory, forest structure characteristics and land cover classes. Iacozza (1999) used the variogram to model the statistical pattern of snow distribution and to provide an estimate of the type and change of spatial dependence of snow depths over various types of sea-ice. In Atkinson (2000), various methods of incorporating spatial information into the classification of remotely sensed images are considered. They focus on the variogram in classification both as a measure of texture and as a guide to the choice of smoothing function. In Jakomulska (2000), experimental variograms were derived for each image pixel from a moving geographic window, the size of which was determined locally by a range. A set of parameters was derived from the variograms (sill, range, mean and sum of variances) derived from each spectral band of IKONOS satellite and multivariate variograms (cross variograms and pseudo-cross variograms) calculated between pairs of the spectral bands. Spectral bands, as well as parameters derived from univariate and multivariate variograms, were used in supervised image classification. Accuracy of textural classification was compared with classic per-pixel classification. Berberoglu (2003) found that the addition of variogram texture information, in particular, variogram lags of 1, 2, and 3 pixels increased the classification of IKONOS images. In Lévesque (2003), variogram analysis combined with conventional image measures was integrated in stepwise multiple regression modeling of forest structure (canopy and crown closure, stem density, tree height and crown size) and health (a visual stress index). Zhang (2003) used the simplified variograms of horizontal direction in each training area as the classification vectors in a thematic pixel by pixel classification of SIR C images. While Kuplich (2003) used it for the estimation of tropical forest biomass using SAR images.

In Kourgli (2004), our objective was to assess the potential of variogram-derived texture measures applied to the classification of high ground resolution imagery. The method was applied to detect land cover in a built-up environment, which has a much finer texture than land cover in general. The technique employed was a combination of the approaches used in a segmentation scheme, with few improvements. Figure 14 shows the results of textural classification of aerial photography (Figure 12). Urban or built-up land is composed of areas of intensive use with much of the land covered by structures. These structures include cities, strip developments along highways, airports, road network, areas occupied by industrial structures and universities that may even be isolated from the urban areas.

We chose to identify four samples (training sites) of homogenous regions (uncovered lands, covered lands, dense built-up environment and less dense built-up environment). These homogenous regions of varying texture must be large enough to allow computation of the semi-variogram with a reasonable number of lags to detect variogram features. In fact, in order to obtain a good texture characterization, it is desirable to work with large windows, since they obviously contain more information than small ones. In terms of the window size, clearly it should not be too large, as compared with the average land feature sizes. If the moving window is larger than the patch of variogram associated with particular land-cover type, the variogram curve will be biased because of the presence of non-typical pixels from other land-cover types. Meanwhile, the moving window should not be too small either. That is why experimental variograms were derived for each training site for different window sizes till ranges for each direction are reached. We obtained a size of 24×24 pixels and computed the variograms of every land-cover type. Variograms of uncovered lands (Figure 13a) and covered lands (Figure 13b) were

Figure 12. Aerial photography of an Algiers' area at a scale of 1/10000

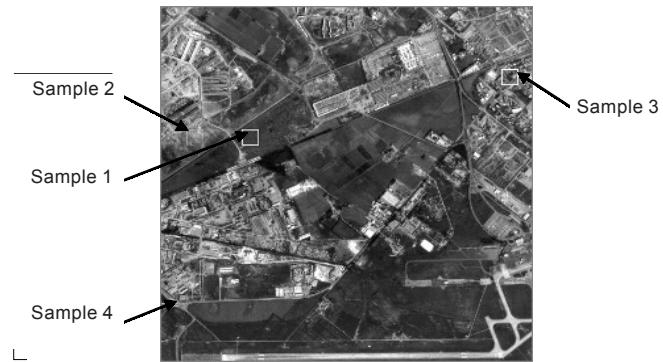


Figure 13. Graphs of $\gamma(d)$ measures of (a) Sample 1 (uncovered lands), (b) Sample 2 (covered lands), (c) dense built-up environment, (d) less dense built-up environment

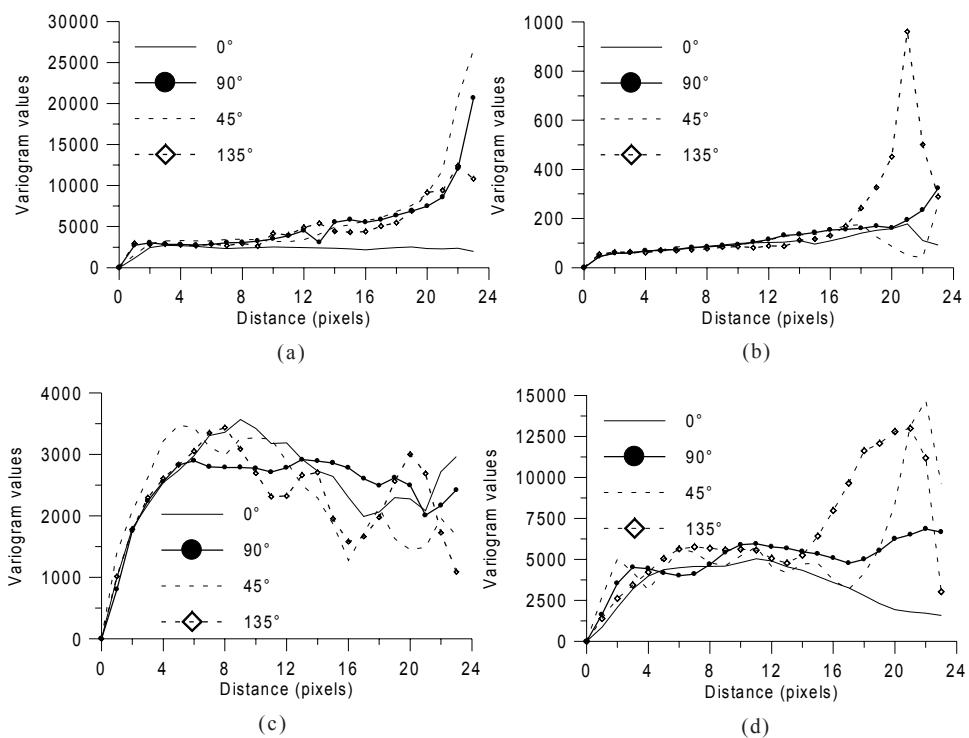
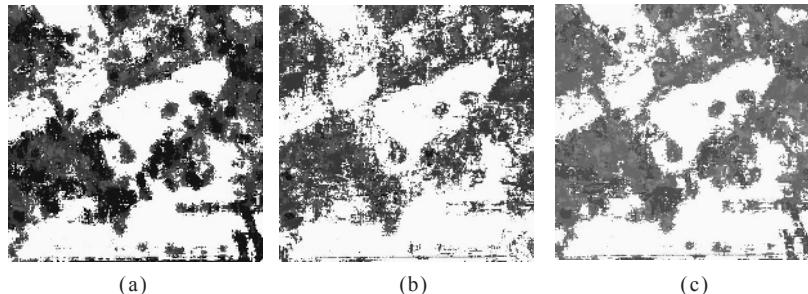


Figure 14. Classification of the aerial photography of Figure 10 using: (a) range of influence, (b) hole-effect, (c) range of influence and hole-effect distances



white (sample 1) = uncovered lands; medium gray (sample 3) = dense built-up environment; darker spots (sample 2) = covered lands; black (sample 4) = less dense built-up environment

essentially flat, exhibiting little or no spatial correlation for lag distances further than two pixels. While variograms of built-up environments (Figure 13c and Figure 13d) present more variations and are quasi-periodic. After determining the moving window size, the semivariance can be calculated by shifting the moving window throughout the entire image. Next, a set of parameters (range and hole-effect) was derived from the variograms and their discriminative potential was assessed using a minimum distance decision. The results obtained show that land cover classes can be well differentiated by range only (Figure 14a) or hole-effect only (Figure 14b), and that both parameters are distinctive for different land types. Furthermore, for a particular class, range seems to give a more homogenous segmentation, while with hole-effect, some confusion between built-up classes is still present. Finally, textural classification combining range and hole-effect (Figure 14c) produces less dispersed classes and best exploits the differences between classes tested.

FUTURE TRENDS AND CONCLUSIONS

This chapter has reviewed the basic concepts, various methods and techniques for processing textured images using an optimal window size. In image interpretation, pattern is defined as the overall spatial form of related features, and the repetition of certain forms is a characteristic pattern found in many natural surfaces. Texture processing has been successfully applied to practical application domains such as automated inspection problems, medical image analysis, document processing and remote sensing.

The primary goal of this chapter was to use variography to customise window size for textural applications. We exploited the variogram features to obtain relationships among the pixels that belong to a similar texture. These relationships allow for the discrimination of every distinctive texture from the others. The study has managed to

address some important applications related to texture analysis. We have investigated a morphological approach to texture analysis which attempts to optimize the structuring of element size and shape using the variogram representation for primitive texture descriptions. Some characteristics (nesting structures) of the variogram representation have been computed, which was introduced for efficient implementation of mathematical morphological functions. Indeed, we demonstrate the ability of the variogram to identify the features of texture and its underlying structure. Because the recognition of various textures in natural scene images is a complex problem, it becomes very important that the choice of image segmentation and texture analysis algorithms is as close to optimal as possible in order to achieve reasonable results. Experimental findings show that interesting results can be obtained when variogram features are applied to textural segmentation and classification, which both require the application of a template in the processing scheme. It has been demonstrated that measures incorporating variogram parameters are robust. Furthermore it has been established that discrimination between textures is possible if variogram features are used. We must point out the fact that the use of variograms is simple and can be easily implemented; we use C++ to do it.

Of course, it is far from constituting a complete independent analysis method. Some further study is required, including the adaptive structuring elements for mathematical morphology. Indeed, whereas the structuring elements used in most applications remain constant as they probe an image, there are situations where structuring elements must change their size, shape and orientation during probing. These changes can be made on the basis of the variogram to obtain an adaptive structuring element; for textural segmentation and classification, evaluating the algorithms on real natural images is quite difficult. Incorporating other features derived from the variogram, such as mean and sum of variances, should increase the rate classification. These hypotheses will be tested in the future.

ACKNOWLEDGMENT

The authors would like to thank Professor Youcef Smara for his critical reading. They acknowledge the contributions of the reviewers for their suggestions and helpful comments.

REFERENCES

- Arivazhagan, S., & Ganesan, L. (2003). Texture classification using wavelet transform. *Pattern Recognition Letters*, 24(9-10), 1513-1521.
- Atkinson, P. M., & Lewis, P. (2000). Geostatistical classification for remote sensing: An introduction. *Computers & Geosciences*, 26, 361-371
- Berberoglu, A., Alphan, H., & Yilmaz, K. T. (2003). Remote sensing approach for detecting agricultural encroachment on the eastern Mediterranean coastal dunes of Turkey. *Turkish Journal of Agriculture and Forestry*, 27, 135-144.
- Brodatz, P. (1966). *Textures: A photographic album for artists and designers*. New York: Dover Publications.

- Chandler, J. H., Rice, S., & Church, M. (2004, July). Colour aerial photography for riverbed classification. In *Proceedings of the Remote Sensing and Spatial Information Sciences Congress*, Turkey, 34(3), 12-23.
- Chen, Y. Q., Nixon, M. S., & Thomas, D. W. (1995). Statistical geometrical features for texture classification. *Pattern Recognition*, 28(4), 537-552.
- Chen, Y.-W., Zeng, X.-Y., & Lu, H. (2002, August). Edge detection and texture segmentation based on independent component analysis. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, Quebec, Canada (Vol. 3, pp. 11-15).
- Chica-olmo, M., & Abarca-Hernandez, F. (2000) Computing geostatistical image texture for remotely sensed data classification. *Computers & Geosciences*, 26(4), 373-383.
- Clausi, D. A., & Yue, B. (2004, August). Texture segmentation comparison using grey level co-occurrence probabilities and Markov random fields. *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR '04)*, UK (Vol. 1, pp. 584-587).
- Cocquerel, J.-P., & Philipp, S. (1995). *Analyse d'images: Filtrage et segmentation*. Paris: Masson.
- Cressie, N. A. (1993). *Statistics for spatial data* (Revised ed.). New York: Wiley-Interscience.
- Dai, X., & Maeda, J. (2004). Unsupervised texture segmentation applied to natural images containing man-made objects. *Transactions of Information Processing Society of Japan*, 45(1), 346-349.
- Daley, N. M. A., Burnett, C. N., Wulder, M., Niemann, K. O., & Goodenough, D. G. (1999, July 6-10). Comparaison of fixed size and variable-sized windows for the estimation of tree crown position. In *Proceedings of IGARSS '98, Geoscience and Remote Sensing Society*, New York (pp. 1323-1325). IEEE.
- Franklin, S. E., Wulder, M. A., & Lavigne, M. B. (1996). Automated derivation of geographic window sizes for use in remote sensing digital image texture analysis. *Computers and Geosciences*, 22(6), 665-673.
- Glotfelty, J. E. (1999). *Automatic selection of optimal window size and shape for texture analysis*. Master's thesis. West Virginia University, USA.
- Grigorescu, S. E., & Petkov, N. (2003, September 14-17). Texture analysis using Renyi's generalized entropies. In *Proceedings of the IEEE International Conference on Image Processing*, Spain (pp. 241-244).
- Haack, B., & Bechdol, M. (2000). Integrating multisensor data and RADAR texture measures for land cover mapping. *Computers and Geosciences*, 26, 411-421.
- Hamami, L., & Lassouaoui, N. (2001). An approach fractal and analysis of variogram for edge detection of biomedical images. In *Lecture Notes In Computer Science: Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks: Bio-inspired Applications of Connectionism-Part II* (Vol. 2085, pp. 336-344).
- Haralick, R. M. (1986). Statistical image texture analysis. In *Handbook of pattern recognition and image processing*. New York: Academic Press.
- Hsin, H.-S., & Li, C.-C. (2002). The adaptive modulated wavelet transform image representation. *Pattern Recognition Letters*, 23(14), 1817-1823.
- Iacozza, J., & Baber, D. G. (1999). An examination of the distribution of snow on sea-ice. *Atmosphere-Ocean*, 37(1), 21-51.

- Jackway, P. T., & Deriche, M. (1996). Scale-space properties of the multiscale morphological dilatation-erosion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2), 38-51.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3), 264-323.
- Jakomulska, A., & Clarke, K. C. (2000, September 19-23). Variogram-derived measures of textural image classification. In *Proceedings of the 3rd European Conference on Geostatistics for Environmental Applications*, Poland (Vol. 1, pp. 181-202).
- Ji, Y., Chang, K. H., & Hung, C.-C. (2004). Efficient edge detection and object segmentation using Gabor filters. In *Proceedings of the 42nd Annual Southeast Regional Conference*, AL (pp. 454-459).
- Khotanzad, A., & Hernandez, O. J. (2003). Color image retrieval using multispectral random field texture model and color content features. *Pattern Recognition*, 36(8), 1679-1694.
- Kourgli, A., & Belhadj-aissa, A. (1997). Approche structurale de génération d'images de texture. *International Journal of Remote Sensing*, 18(17), 3611-3627.
- Kourgli, A., & Belhadj-aissa, A. (2000, August 31-September 2). Characterizing textural primitives using variography. In *Proceedings of IMVIP2000*, Belfast, Ireland (pp. 165-175).
- Kourgli, A., & Belhadj-aissa, A. (2004, September 29-October 1). Optimizing texture primitives' description based on variography and mathematical morphology. *Lecture Notes in Computer Science: Image Analysis and Recognition, ICIAR 2004*, Porto, Portugal (pp. 866-873).
- Kourgli, A., & Belhadj-aissa, A. (2004, October 10-13). Texture primitives description and segmentation using variography and mathematical morphology. In *Proceedings of the IEEE SMC 2004*, The Hague, The Netherlands (pp. 866-873).
- Krishnamurthy, S., Iyengar, S. S., Hoyler, R. J., & Lybanon, M. (1994). Histogram-based morphological edge detector. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4), 4759-767.
- Kuplich, T. M., & Curran, P. J. (2003). Estimating texture independently of tone in simulated images of forest canopies. In *Anais XI Simosio Brasileiro Sensriamento Remoto, INPE*, Avril, Brasil (pp. 2209-2216).
- Lee, K.-H., Morale, A., & Ko, S.-J. (1997). Adaptive basis matrix for the morphological function processing opening and closing. *IEEE Transactions on Image Proceedings*, 6(5), 769-774.
- Lévesque, J., & King, D. J. (2003). Spatial analysis of radiometric fractions from high-resolution multispectral imagery for modelling individual tree crown and forest canopy structure and health. *Remote Sensing of Environment*, 84, 589-602.
- Li, C.-T. (2003). Multiresolution image segmentation integrating Gibbs sampler and region merging algorithm. *Signal Processing*, 83(1), 67-78.
- Liapsis, S., Sifakis, E., & Tziritas, G. (2004). Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms. *Journal of Vision Commun and Image*, 15, 1-26.
- Maillard, P. (2000). *Texture in high resolution digital images of the earth*. PhD thesis, University of Queensland, Australia.

- Malpica, N., Ortuno, J. E., & Santos, A. (2003). A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24(9-10), 1545-1554.
- Manian, V., Vasquez, R., & Katiyar, P. (2000). Texture classification using logical operators. *IEEE Transactions on Image Analysis*, 9(10), 1693-1703.
- Manthalkar, R., Biswas, P. K., & Chatterji, B.N. (2003). Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters* 24(12), 2061-2068.
- Mardia, K. V., Baczkowski, A. J., Feng, X., & Hainsworth, T. J. (1997). Statistical methods for automatic interpretation of digitally scanned fingerprints. *Pattern Recognition Letters*, 18, 1197-1203.
- Materka, A., & Strzelecki, M. (1998). *Texture analysis methods—a review*. Technical University of Lodz, COST B11 Report.
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58, 1246-1266.
- Ohanian, P. P., & Dubes, R. C. (1992). Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8), 819-833.
- Puig, D., & Garcia, M. A. (2001, May). Determining optimal window size for texture feature extraction methods. In *IX Spanish Symposium on Pattern Recognition and Image Analysis 2001*, Spain (Vol. 2, pp. 237-242).
- Reed, T. R., & Du Buf, J. M. H. (1993). A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*, 57(3), 359-372.
- Rushing, J. A., Ranganath, H. S., Hinke, T. H., & Graves, S. J. (2001). Using association rules as texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), 845-858.
- Sharma, M. (2001). *Performance evaluation of image segmentation and texture extraction methods in scene analysis*. Master thesis. University of Exeter, UK.
- Singh, M., & Singh, S. (2002, August 11-15). Spatial texture analysis: A comparative study. In *Proceedings of the 15th IEEE International Conference on Pattern Recognition (ICPR '02)*, Quebec, Canada (pp. 676-679).
- Stainvas, I., & Lowe, D. (2003). A generative probabilistic oriented wavelet model for texture segmentation. *Neural Processing Letters*, 17(3), 217-238.
- Sussner, P., & Ritter, G. X. (1997). Decomposition of gray-scale morphological templates using the rank method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6), 649-658.
- Tuceyran, M., & Jain, A. K. (1998). Texture analysis. In C. H. Chen, L. F. Pau, & P. S. P. Wang (Eds.), *Handbook of pattern recognition and computer vision*. Singapore: World Scientific (pp. 235-276).
- Verly, J. G., & Delanoy, R. L. (1993). Adaptive mathematical morphology for range imagery. *IEEE Transactions on Image Processing*, 2(2), 272-275.
- Zhang, Y. (2003). *Spatial autocorrelation in remote sensing image analysis: A report*. Environmental Remote Sensing Laboratory Electrical Engineering Department, University of Nebraska, Lincoln, Nebraska.
- Zucca, A., Diaz, G., Setzu, M. D., & Cappai, C. (1997). Chromatin pattern by variogram analysis. *Microsciences Research Technology*, 1 39(3), 305-11.
- Zucker, S. W., & Terzopoulos, D. (1980). Finding structure in co-occurrence matrices for texture analysis. *Computer Graphics and Image Processing*, 12, 286-308.
- USC-SIPI Image Database. (n.d.). Retrieved January 7, 2000, from <http://sipi.usc.edu/database.cgi>

Chapter XII

Methods and Applications for Segmenting 3D Medical Image Data

Hong Shen, Siemens Corporate Research, USA

ABSTRACT

In this chapter, we will give an intuitive introduction to the general problem of 3D medical image segmentation. We will give an overview of the popular and relevant methods that may be applicable, with a discussion about their advantages and limits. Specifically, we will discuss the issue of incorporating prior knowledge into the segmentation of anatomic structures and describe in detail the concept and issues of knowledge-based segmentation. Typical sample applications will accompany the discussions throughout this chapter. We hope this will help an application developer to improve insights in the understanding and application of various computer vision approaches to solve real-world problems of medical image segmentation.

INTRODUCTION

The advances in medical imaging equipment have brought efficiency and high capability to the screening, diagnosis and surgery of various diseases. The 3D imaging modalities, such as multi-slice computer tomography (CT), magnetic resonance imaging (MRI) and ultrasound scanners, produce large amounts of digital data that are difficult and tedious to interpret merely by physicians. Computer aided diagnosis (CAD) systems will therefore play a critical role, especially in the visualization, segmentation, detection, registration and reporting of medical pathologies. Among these functions, the segmen-

tation of objects, mainly anatomies and pathologies from large 3D volume data, is more fundamental, since the results often become the basis of all other quantitative analysis tasks.

The segmentation of medical data poses a challenging problem. One difficulty lies in the large volume of the data involved and the on-time requirement of medical applications. The time constraints vary among applications, ranging from several tens of milliseconds for online surgical monitoring, to seconds for interactive volumetric measures, to minutes or hours for off-line processing on a PACS server. Depending on the application, this puts a limit on the types of methods that may be used. Another major hurdle is the high variation of image properties in the data, making it hard to construct a general model. The variations come from several aspects. First, the complexity of various anatomies maps to the large variation of their images in the medical data. Second, the age, gender, pose and other conditions of the patient lead to high inter-patient variability. Last, but not the least, are the almost infinite variations in an anatomy due to pathology or in the pathological structures. On the other hand, medical applications usually have a strong requirement of robustness over all variations. Beside the above challenges, system issues exist for the major modalities, such as noise, partial volume effects, non-isotropic voxel, variation in scanning protocols, and so forth. These all lead to more difficulties for the medical segmentation problem.

Knowledge-Based Segmentation

Medical image segmentation has the advantage of knowing beforehand what is contained in the image. We also know about the range of size, shape, and so forth, which is extracted from expert statements. In other fields of computer vision, such as satellite image analysis, the task of segmentation sometimes contains a recognition step. Bottom-up strategy is usually used, which starts with the low-level detection of the primitives that form the object boundaries, followed by merging. One sophisticated development is Perceptual Organization (Sarkar & Boyer, 1994; Guy & Medioni, 1996; Mohan & Nevatia, 1989), which attempts to organize detected primitives into structures. It is regarded as the “middle ground” between low-level and high-level processing. In 3D medical data, grouping is much more difficult due to the complexity of medical shapes. Because of this, top-down strategies prevail in 3D medical image analysis.

Knowledge-based segmentation makes strong assumptions about the content of the image. We use prior knowledge to find and tailor a general segmentation method to the specific application. Global priors are applied when the local information is incomplete or of low quality. It is a top-down strategy that starts with knowledge or models about high-level object features and attentively searches for their image counterparts.

The past several decades witnessed dramatic advances in the fields of computer vision and image analysis, from which the area of medical image analysis is derived. Various methods and frameworks for segmentation have been proposed, and many are driven by the needs of medical image analysis. These provide valuable theoretical thoughts as the basis of knowledge-based segmentation, but only at a high level. Typically, such a method is shown to be generic as it works on a number of cases from various applications with reasonable successes. This is quite different from the requirement of medical image segmentation in the real world, which depend heavily on the specific application—the workflow of the medical procedure, the anatomy and pathology

of interest, the performance and accuracy requirements and the user inputs. Given all the priors and conditions of a medical application, we need to design the algorithm that will be the best compromise between accuracy, speed, robustness and user inputs. The resulting system will be specific to the given application; not only the algorithm, but also the parameter settings. A medical image analysis algorithm will be put to tests on thousands of data sets before it can be made into a clinical product. Hence, even if it is application specific, such an algorithm is general in that it has to cover all possible variations of the application.

A developer in this field not only needs to master the various methods in computer vision, but also understand the situations each of them may be applied to, as well as their limitations. In the next sections, we will focus on real world issues of medical image segmentation. We will first give an overview of the relevant low-level methods with examples. We will then discuss the popular frameworks for incorporating global priors and their limitations. We will present the principles of a novel strategy that was developed recently (Shen, Shi, & Peng, 2005) for an sample application. Finally, we will conclude the chapter with an outlook on the future trends in this field.

APPLICATIONS USING LOW-LEVEL SEGMENTATION METHODS

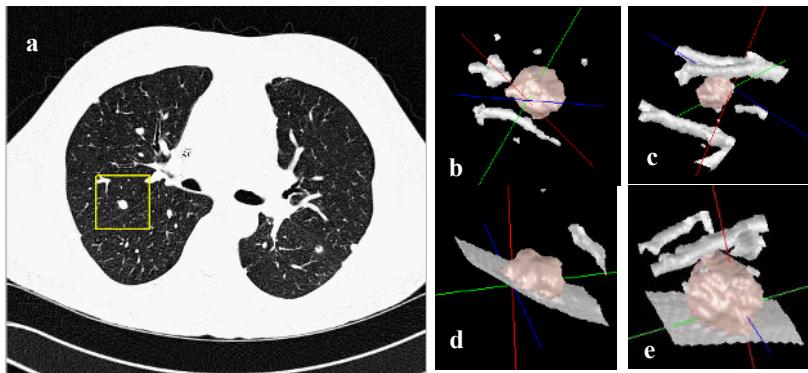
In this section we will give 3D medical application examples that use low-level segmentation methods. We define low-level methods as those that do not incorporate global shape priors of the object to be segmented, but mostly rely on local image properties. Many surveys and textbooks (Sonka, Hlavac, & Boyle, 1998) give detailed descriptions of all the segmentation methods, and this short chapter will not serve as a duplication of these efforts. Instead, we will give examples to show where each category of low-level method is adequate, and also review the recently developed methods.

Segmentation Using Regional and Morphological Methods

Region-based methods group together voxels that have some common properties. They are applicable when there is high contrast between object and background and intra-object variation is minimal. As shown in Figure 1, in a chest CT data, the lung area maps to dark regions, while the chest wall, the vessels, the airways and nodules map to bright regions of the same intensity range. A typical application is the segmentation of a lung nodule given a click point in the nodule. Since there is high contrast between object and background, a simple 3D region grow (Sonka et al., 1998; Adams & Bischof, 1984) is the best method to obtain the foreground voxels that are above a clear-cut threshold. The challenges are as follows: First, the nodule can be connected to a vessel and/or the chest wall, which has the same intensity range as the nodule. Second, it is required that the segmentation result be completely independent of the click point as long as it is within the nodule. Last, but not the least, is the high expectation from the user of success rate on a large number of cases due to the fact this looks like an easy problem.

These challenges are met by incorporating prior knowledge about the application. First, the click point is guaranteed to be in the nodule. Second, the nodule size range is

Figure 1. Lung nodule segmentation from multi-slice chest CT data



(a) An axial slice of the volume data. At the center of the marked box is the nodule to be segmented. Surface shaded display of the segmentation results are shown for nodules that are (b) solitary, (c) connected to vessels, (d) connected to chest wall, and (e) connected to both vessel and chest wall

known. Third, the nodule is known to be compact, while a vessel is assumed to be of elongated shape, and the chest wall is mostly a smooth surface. Given the click point and size range, we define a cubic volume of interest (VOI) centered at the click point, and hence isolate out the object and its immediate neighbors. In this VOI, methods such as mathematical morphology are applied to consistently find the nodule center and separate the nodule from the connected structures.

Mathematical morphology utilizes local shape information (Sonka et al., 1998; Haralick, Sternberg, & Zhuang, 1987). Although grayscale operations are also defined, this type of method is mostly used in binary images that are generated from an earlier process such as region grow. As illustrated in Figure 1(c), the foreground object contains a nodule connected to a vessel. The basic operators such as dilation, erosion, opening and closing can be used for boundary modifications and to separate two connected objects. However, the more effective separation techniques are based on the important concept of geodesic distance. A geodesic distance of a point inside the foreground object is the shortest travel distance to the background by any path that is completely inside the object.

Using a geodesic distance map, the vessel voxels can be removed from the nodule since their distance values are relatively small. Again, the knowledge about relative sizes between the nodule and the vessel are used to determine the distance thresholds. We could use more sophisticated watershed algorithms (Vincent & Soille, 1991) to find and separating distance basins, if the over-segmentation issue can be overcome (Meyer & Beucher, 1990).

The region-based methods are based on the similarity or connectivity of voxels, therefore they can be extended to more general situations by redefinition of the similarity

or connectivity. Recently, the concept of fuzzy connectivity has been introduced (Udupa, Saha, & Lotufo, 2002; Herman & Carvalho, 2001), which is defined between every possible pair of voxels. All objects are initialized with a reference point and then grow by competition to acquire voxels according to the strengths of fuzzy connectivity.

Overall, region-based segmentation is robust, simple to implement and fast when the object is small. In some cases, region merging and splitting is necessary to the grown region according to other criterions (Sonka et al., 1998; Chang & Li, 1994). However, the more heuristics that have to be applied in order to get a decent result, the less reliable the algorithm will become. The bottom line is that region-based, low-level methods should be applied to regions with relatively homogenous intensities.

Segmentation Based On Primitive Detection

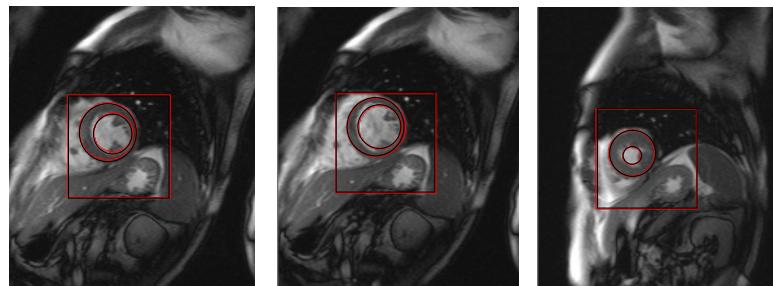
Image primitives are local properties such as edges, ridges and corners. Instead of looking for properties of all voxels inside the region, primitive-based methods look for the region boundaries. There are many recent developments in primitive detection, especially edge detection, which is the basis of most segmentation schemes. The detected primitives are usually fragmented, and the challenging problem of linking edges and lines into boundaries was the focus of many works in the early and middle 90s (Basak, Chanda, & Majumder, 1994; Wu & Leahy, 1992).

Basic edge detection is achieved by applying an edge operator (Sonka et al., 1998) followed by non-maxima suppression. A systematic treatment of edge detection was provided by Canny (1986), which is an optimal detector for step edges, and can be adapted to detect other features when different filters are selected. Recently, Brejl and Sonka (2000) invented a 3D edge detector that is particularly fit for anisotropic situations, which are typical in 3D medical data. Freeman and Edward (1991) proposed a method of systematically detecting all linear primitives, including step edges, ridges and valleys, as well as their orientations. They designed a basis of filters from which an arbitrarily-oriented filter can be generated to selectively detect the primitives of a specified orientation. This is particularly useful for high-level feature detection. For instance, if we know beforehand there should be a long ridge at a certain orientation and location, a ridge filter of that direction can be applied.

As a special type of primitive-based method, Hough Transform (Sonka et al., 1998; Shankar, Murthy, & Pal, 1998) utilizes a very powerful scheme for noisy medical images—voting. It is applicable when the object roughly fits a simple parametric model, such as a rectangle. Although in medical applications precisely regular shape is rare, some are roughly spherical or with a circular cross-section. For instance, shown in Figure 2 are the segmentation results of the left ventricle from cardiac MR data. The outer and inner boundaries approximate a circle, which has three free parameters in its parametric equation. Edge detection followed by non-maxima suppression is performed within the region of interest, and each edge pixel casts one vote for every circle equation it satisfies. The circle with the highest vote is taken as the object boundary. To achieve good results, the parameter space needs to be carefully divided into cells. Sometimes the vote can be weighted by edge strengths.

The power of this algorithm comes from its strong assumption. It works in spite of noise, weak boundaries and even occlusions, as long as the good edges can form a majority over the outliers. Certainly, the parametric model has to be simple before the

Figure 2. Segmentation of the left ventricle wall from cardiac MRI data. The white circles mark the inner and outer boundaries of the left ventricle.



algorithm becomes too complex. However, in some cases even when the object shape is not regular, we may still use it to find at least the correct centroid of the object.

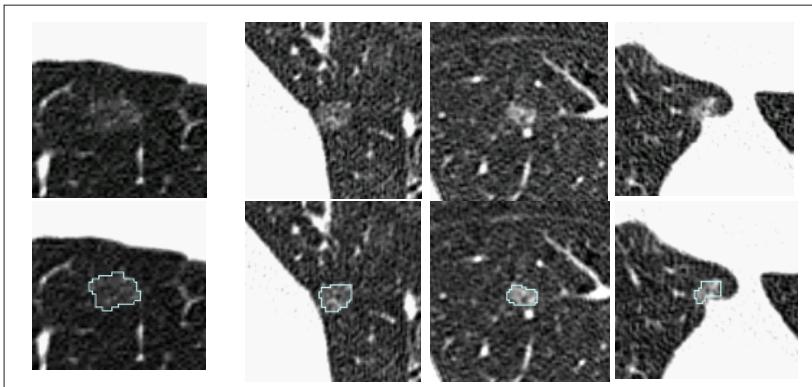
The Hough Transform is a type of robust estimation method. Its principle of utilizing the majority of pixels can be widely applied. In the presence of abnormality, noise and high complexity, the robustness of a segmentation algorithm is usually determined by the number of voxels the decision is made upon.

Statistical Segmentation Methods

Statistical methods treat the intensity of each voxel as an event, and model the local intensity variations by the estimation of probabilistic distributions. The goal is to use classification frameworks to determine whether a voxel belongs to a certain object. The set of voxels that are labeled as object points forms the segmented object. A simple scheme is to estimate the probability density functions (pdf) of different objects or background by computing the frequency of histograms from sample data. The estimated pdfs can then be applied to the test data. Each voxel can be classified using conventional classification methods, such as Bayesian classification or maximum likelihood decision. A more sophisticated algorithm would estimate the joint pdfs of the voxels in the neighborhood to capture the spatial local interaction properties between image labels. The most popular method is the Markov Random Field (MRF) framework, in which the pdfs are estimated with advanced algorithms such as EM algorithm. The segmentation is often obtained using a MAP estimator (Sun & Gu, 2004).

These type of methods can be applied when there is little knowledge about object shape and the intensities are nonhomogenous. A typical case, as shown in Figure 3, is the segmentation of ground glass nodules (GGN) from chest CT volume data, which has strong clinical significance due to the high malignancy rate of GGNs (Zhang, Zhang, Novak, Naidich, & Moses, 2005). Due to large inter-nodule variations, an iterative approach is applied to gradually capture the local image properties. A rough intensity model is generated from training data and applied to the first round of classification. The resultant segmentation is then used to estimate a more specific distribution for the

Figure 3. Segmentation of ground glass nodules (GGN) from chest CT data using Markov Random Field. Top row: Slices of volumes of interest (VOI). Bottom row: Slices of segmented GGN from the VOI above it.



(Courtesy of Li Zhang from Siemens Corporate Research)

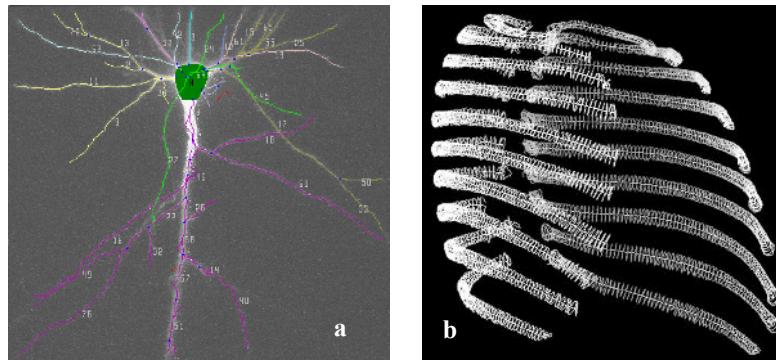
current nodule, which is in turn applied to the next round of classification. Several iterations combined with other heuristics give consistent segmentation results with respect to different user clickpoints. The same iterative strategy was used by Marroquin, Vemuri, Botello, and Calderon (2002) in the segmentation of brain MRIs. The establishment of pdfs is a learning process that requires large sample sets for each object, which may not be available for medical image segmentation. Further, large variations in image properties often lead to overlapped pdfs. Usually, heuristics have to be combined to achieve a decent result.

SEGMENTATION INCORPORATING GLOBAL PRIORS

For more complex situations, low-level methods will not be successful. This is because the objects to be segmented in a medical application usually have high internal nonhomogeneity and strong internal edges. In these cases, a global prior is needed.

We first give as intuitive examples two special applications that utilize the global priors of object shapes—the tracing-based segmentation of tubelike structures. As shown in Figure 4, 3D tracing algorithms were developed for the segmentation of neurons from laser-scanning confocal image stacks (Al-Kofahi et al., 2002), and of rib structures from chest CT data (Shen, Liang, Shao, & Qing, 2004). A tracing algorithm uses the global prior knowledge that the object has piecewise linear shapes with smooth surfaces. First, seed points are automatically detected for all objects by techniques such as sparse grid search or 2D cross-section analysis. From each seed, tracing proceeds iteratively along

Figure 4. 3D exploratory tracing-based segmentation



(a) Segmentation result of neuron from confocal image stacks, the centerlines are overlaid on original image (2D projection) (Courtesy of Khalid Al-kofahi from Thomson Legal & Regulatory, Eagan, MN); (b) Segmentation result of rib structures from chest CT image shown in Figure 1. Shown are the rib contours and centerlines. They are naturally separated. A graphical surface reconstruction algorithm can be applied to obtain the rib surfaces.

each object, using edge information to determine the tracing directions. Carefully designed stopping criteria are applied to terminate a trace.

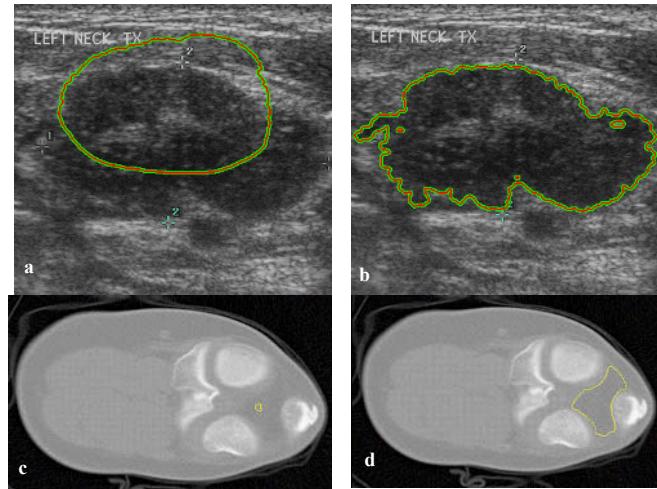
This approach has the advantage of obtaining object centerlines and boundaries simultaneously, and all tube structures are naturally separated. Another significant advantage is the low computational cost. This type of algorithm is exploratory. Only the voxels on or near the objects are processed, which is a small portion of the image data.

Deformable Models

The deformable models provide a framework that allows the incorporation of global priors. It is widely applied to medical data, and the potential is far from fully explored. Before global prior is introduced, a deformable model is a unified approach to the capture of local image properties through front evolution, without the need of local primitive grouping. Further, by the constraints that a smooth shape is to be maintained during evolution, the frontier points move with interaction rather than independently. Figure 5a and 5b show the two example segmentations using the level set method, which is one popular deformable model.

Kass, Witkin, and Terzopoulos (1988) proposed one of the first deformable models—snakes—in which they introduced the energy minimizing mechanism. By using gradient decent and variational approach, the 2D contour is attracted to the local energy minimum. Through contour or surface evolution, a deformable model integrates three factors to jointly determine the segmentation. First, the smoothness requirement of the shape constrains the relative motion of frontier points. These points are only allowed to

Figure 5. Surface evolving in the level set segmentation. The left column shows initialization, and the right column shows converged boundary.



(a) and (b) Larynx tumor in ultrasound data (c) and (d) Hoffa pad in CT knee data.
Courtesy of Gozde Unal and Greg Slabaugh from Siemens Corporate Research

move such that a smooth front is maintained. Second, the surface actively adapts to the local image primitives, such as edges, ridges, etc. Finally, it is possible to impose global geometrical constraints on the shape. In the initial treatment of snakes, these three factors are integrated as energy terms in the partial differential equation (PDE).

Accordingly, three issues are to be addressed when designing a deformable model. First of all, we need to decide how to represent a shape or surface. The other two major issues are the design of local image forces and the incorporation of global priors to constrain the front evolution.

The issue of shape representation becomes more challenging in a 3D situation. Pentland and Sclaroff (1991) used an approach based on finite element method (FEM) to establish a parametric model of shapes. Lorenz and Krahnstover (1999) used a 3D mesh to represent the vertebra shape. Recently, Slabaugh and Unal (2005) developed a more general deformable mesh for surface evolving.

The Level Set Method for Shape Representation and Evolution

In the past decades, the level set methods (Malladi, Sethian, & Vemuri, 1995; Adalsteinsson & Sethian, 1995) received much attention. The surface is represented implicitly. Typically, for a close surface, a signed distance function is defined in a volume

that contains that surface. The value of each voxel is the geodesic distance of that voxel to the closest point on the surface. Since this value on the surface is zero, the set composed of all surface points is named the zero level set of the signed distance function. The distance values of voxels in and outside the surface are set to be negative and positive, respectively. This implicit representation of the surface by the zero level set of the signed distance function provides great simplicity at the cost of an increase in dimension. The implementation is relatively simple, and provides a flexible platform to design various algorithms. Further, level set is transparent to dimension, which makes it much more desirable than some other methods that are hard to adapt to 3D.

Until recently, most level set researchers focused their efforts on the problem of surface evolving under local image properties, such as defining local image force or speeds, either edge-based or region-based (Chan & Vese, 2001; Chakraborty, Staib, & Duncan, 1996). A recent survey by Suri, Liu, and Singh (2002) gave a systematic summary of all the important efforts that address in depth the local surface evolvement. Without a prior shape model, however, a deformable model such as level set only reacts to local properties. For edge-based force, the surface will converge at local edges. For region-based force, the surface will converge when the average intensity of voxels in the object and background differs the most. A medical structure usually has complex local image properties, such as incomplete boundaries, noise, nonuniform internal intensities, etc. Therefore the most common problems for low-level segmentation methods also bother a deformable model—the leaking at weak boundaries or being caught by internal edges. The surface will truthfully conform to whatever local information it encounters. Even with the most advanced local constraining schemes, these problems persist.

Statistical Model as Global Prior Representation

Cootes, Taylor, Cooper, and Granham (1995) proposed the popular active shape models (ASM). Cootes, Edwards, and Taylor (2001) later proposed the active appearance models (AAM). Both of the two methods used statistical models to represent prior knowledge. The surface was represented by a point distribution model (PDM), which was a set of landmark points that were distributed on the object boundaries. For a given shape, the locations of all the landmarks were concatenated to form long vectors as the representation. The success of a PDM depends on correspondence of landmarks, which is challenging in 3D. Overall, the PDM models are difficult to implement in 3D. It is a very active field that attracts many contributions to solve these difficulties (Hill & Taylor, 1994; Walker, Cootes, & Taylor, 2002; Horkaew & Yang, 2003).

On the other hand, this was among the first systematic efforts to introduce statistical representation of global shape priors into a deformable model. Sample vectors were extracted from a set of registered training shapes and principal component analysis (PCA) is performed on the set of sample vectors. The dimension of the covariance matrix equals that of the sample vectors, but only the eigenvectors with the largest eigenvalues are taken to form the basis of the linear model space. The segmentation process first transforms the mean shape to the data, and then finds the correspondence of the mapped model landmarks by searching for strong edges in the contour's normal direction. Subsequently the shape vector computed from the correspondent points is projected onto the model basis to find the best matched model. This means a large number of residual components are discarded. The matched model is again matched onto the image

data for next iteration. In a PDM, the representation of surface and the incorporation of global priors are combined and a shape is only allowed to vary within the model space. The local surface evolution is limited and for the only purpose of finding the matched model.

For the level set framework, recently Leventon, Grimson, and Faugeras (2000) introduced the similar PCA model. PCA analysis is performed on a set of registered training samples, each represented by a signed distance function defined in a volume. Compared to the PDM, the linear PCA model on signed distance function is only valid within the limited space centered at the mean shapes, which is clearly a disadvantage.

Limitations of the PCA Statistical Model

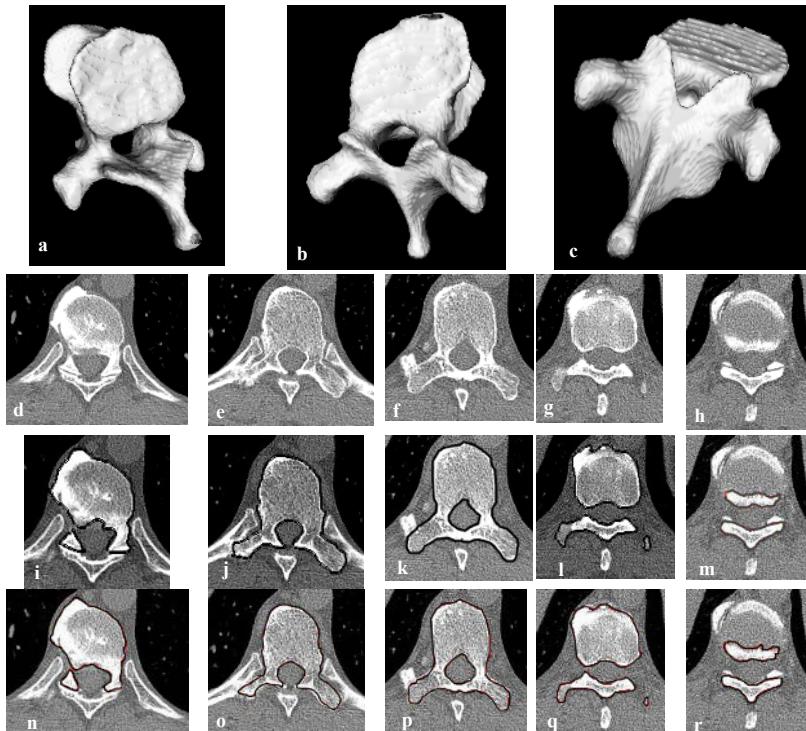
The PCA statistical models have limitations both in the representation and application of global priors. The representation is incomplete, especially when the structure is complex and with high variations due to age, gender, race and almost infinite types of pathological abnormalities. Coverage of these variations requires a large number of modes, in addition to the need for high-valued coefficients. Setting up such a large and high-dimension model space is impractical due to the limited size and coverage of the training set, in which every sample needs to be obtained by arduous manual segmentation. Further, searching in such a complex model space is also impractical if any reasonable performance is desired. Therefore, any practical PCA model space will be restricted to that of a relatively low dimensional, and the model shape can only vary in the neighborhood of the mean shape.

The established PCA model is applied to segmentation as follows. In an iteration of surface evolution, a model is identified from the PCA model space such that it has the minimum shape difference from the current surface. The difference is then used as a penalty term and added to the evolving force. In such a scheme, how to determine the relative strengths of model and local force is a critical issue. In a PDM, the local image force is only used to evolve the surface to its best model, and the shape residues not covered in the limited model space will be ignored. A level set scheme better understands the limitations of a model, and allows the surface to converge to a shape not covered by a model under local image force. The high-level model force and the local image force are added together, with a multiplier on one of the two terms to adjust the relative strength.

For a PDM, precise segmentation of a complex structure is obviously not possible, since the object is completely restricted to an inaccurate model space. In the literature, PDM showed its success mostly in 2D cases. Mitchell et al. (2002) extended it to 3D segmentation of a left ventricle from cardiac MR and Ultrasound images, in which the object of interest has a relatively simple shape and hence the issue of inaccurate model space is not severe. In some applications, such as ultrasound, the shape is not well defined. Even humans cannot consistently delineate the object boundaries. The goal is then to find a shape that best fits the visual clues of boundaries. This is where PDM is useful but, strictly speaking, it is geometric modeling rather than precise segmentation. On the other hand, geometric modeling sometimes is the best we can do since validation of results is not possible. Depending on application, geometric modeling may be a satisfactory solution even if it does not represent the actual object boundaries. When there is consistent ground truth from human, precise segmentation should be achieved.

In comparison, the level set scheme is fit for precise segmentation, since surface evolution is allowed to converge to local image primitives. However, the PCA model (Tsai et al., 2003; Yang, Staib, & Duncan, 2004) introduced the same-level competition scheme between global prior and local image forces. Because of the inaccurate model, the global prior term will often disagree with the local image term. In a situation where the shape is not well defined, the model term can be made very strong by adjusting the multiplier, and the resulting surface boundaries will be very close to that of a model shape. A example is shown in Figure 5c and 5d, in which a prior shape model was applied. In such a case, the user has only vague visual clues to identify the actual object boundary, therefore the boundaries suggested by a model shape are well accepted. Much work in the literature follows these types of strategies.

Figure 6. 3D level set segmentation of the vertebra



(a)-(c): 3D view of the complex structure of a vertebra shape. (d)-(h): Vertebra on selected axial slices, with its neighboring vertebra and rib structures. While the shape looks well defined for a human observer, local boundaries are far from clear. Also notice the pathology induced abnormal shape on the top-left of (d). (i)-(m): The 2D boundaries delineated by a human observer. Regions outside the contours belong to the neighboring vertebrae and rib structures. (n)-(r): Converged 3D level set surface of segmented object projected onto selected axial slices.

In a situation where an object shape is well defined, the user would expect a precise segmentation. A good example is the human vertebra in high-contrast data, such as multi-slice CT images, as shown in Figure 6.

This 3D complex structure has a relatively well-defined shape, at least for a human observer, who can perform consistent delineation of the object boundaries on 2D slice images. However, there are many places where local boundaries are weak, diffused and have gaps. Further, such a structure is adjacent or even connected to a neighboring structure that has similar image properties. This is the typical case where we cannot afford an accurate model, and the inaccurate model will compete with the local image forces. If we make the model strong, the surface will not converge to some of the strong local features and abnormal shapes that are out of the model space. On the other hand, if we reduce the strengths of the model, then level set will leak out at places where local primitives are weak and hence need guidance. Obviously, this is a dilemma that cannot be completely addressed by adjusting the multiplier. In the work in which Leventon et al. (2000) defined this competition scheme, an attempt was made to segment the human vertebrae. The training shapes are from the vertebrae of the same data set, leaving the middle vertebrae out as the test data. The problem manifested itself. The not-well-fit model shape competes with local primitives to pull the evolving surface away from these strong features and the abnormal shapes. From the published result, the convergence was quite far from the object boundaries that can be easily delineated by a human observer.

A Better Strategy

The limitations of the PCA models for representing global prior knowledge can be understood from an alternative point of view. Mathematically, the PCA models are based on the moment average of the training samples, and hence high frequency information will not be preserved. On the other hand, human observers to identify a complex structure most effectively use high frequency information, such as abrupt ridges, corners, etc. These are singular points, lines or planes where derivatives do not exist. A surface model represents smooth surfaces well, but does not incorporate these singular features. In other words, this important portion of the prior knowledge has to be represented more explicitly rather than implicitly embedded in the shape model.

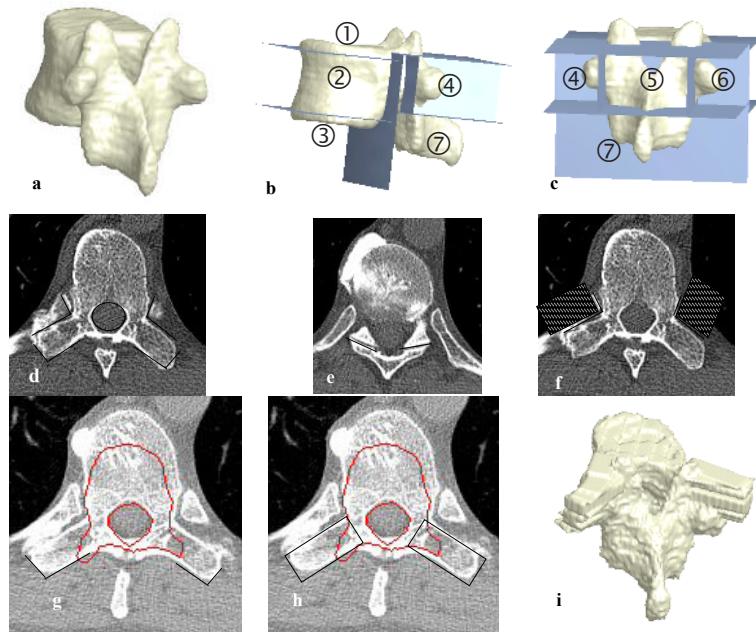
Further, global prior knowledge, such as surface models, should not be put in competition with local image properties at the same level. When simply added to the local image force, a global prior loses its “global” status and degrades into a local factor. A global prior should act as a high level guidance to local evolution, so that the surface will converge to the “correct” local primitives, hence avoiding the problems of local minima and leak-out.

To determine what global priors are needed, we want to find out what causes the failure of the surface evolution when no global prior is applied. As shown in Figure 6, the internal intensity of the vertebra is very nonhomogenous, and there are many internal edges. A level set front when initialized inside the vertebrae will ensure converge to these internal edges. The only way to prevent this from happening is to keep these internal edges away from the evolution path. Since the convergence of level set depends on the initial front, a major effort is to design the initial front such that it is close to the boundaries but outside of the vertebrae. The surface will then converge to the outside boundaries

before it even encounters the internal edges. Another problem is the weak boundary locations, from which the surface will leak out. We can carefully design the speed map according to the various local properties to avoid this problem. For instance, we can use region based speed at locations where edges are weak. Next, a more challenging issue is the adjacent, or even connected, ribs and vertebrae that have the same image properties. To separate them from the vertebrae of interest, we need to explicitly detect the high-level features formed between the vertebrae and its neighboring structures.

From the above, global priors should then include a surface model as a general description of shape, augmented by a set of known high-level features. The surface model is constructed as follows: First, a small number of vertebrae, roughly covering the spectrum of shape variations within the normal range, are selected and their surfaces are

Figure 7. Representation and application of prior knowledge



(a) The mean shape. (b)(c) The plane division of the volumes into regions. Each region has its specific speed design. (d)-(e) High-level boundary features and context features which are marked with black and gray, respectively. A context feature is not on object boundary, but is on the object interfaces. (f) Blocker speed regions. (g) Mean shape mapped to the image. Two high-level boundary features are detected. (h) The mean shape is augmented to make it close to the object boundaries. (i) 3D structure of the augmented mean shape to be used as the initial front for level set surface evolving.

manually segmented. These surfaces are transformed into a common coordinate system using similarity transformation such that the spinal channel and inter-vertebrae planes are registered. We do not intend to match the surfaces themselves, since that is unnecessary and will require nonlinear warping of the surfaces. Afterwards, the registered surfaces are represented as a set of signed distance functions. We then compute the mean signed distance function, whose shape is shown in Figure 7a. It has the basic look and structure that belong to and characterize a vertebra. The mean shape model is used as a rough guide or template and hence does not need to cover the large variations. This low requirement allows a small sample size.

Using the mean shape as a reference system, we define planes that divide the volume into regions. For each region, the proper speed design is applied, which varies in the type of speed and the constants. The region division is shown in Figure 7b and 7c.

The mean shape representation is augmented with a set of high-level features, which are formed by linear primitives including intensity edges, ridges and valleys. If the locations of the strong and same-type primitives in a local neighborhood roughly fit to a simple parametric model, such as a plane, that plane is recorded as a high-level feature. There are two types of high-level plane features, and both are recorded with reference to the common coordinate system. If fitted by the ridge or edge primitives from the object boundary, it is named a high-level boundary feature. A high-level context feature, on the other hand, is fitted by valley primitives located at the interface between the object and a neighboring structure. Plane is the simplest parametric model, which can be extended to piecewise linear models. Examples of high level boundary and context features are shown in Figure 7d and 7e.

Application of Global Priors in Surface Evolution

Similar to the registration process of training samples, the mean shape is mapped to the image data using the estimated similarity transformation, as shown in Figure 7g. This is, of course, specific to this application. However, a transformation can be estimated with other more general methods to map the initial shape, and we have seen examples in the literature (Baillard, Hellier, & Barillot, 2001). The boundaries of the mapped mean shape are usually close to the boundaries of the object, except that the surfaces at the two transverse processes are too far inside. Such an initial front will be captured by local minima and not converge to the outside true boundaries. It therefore needs to be modified, and we use the high level features to achieve this.

The high-level feature models are also mapped to the neighborhood of the features in the image data. The detection of high-level boundary and context features uses mainly parametric model-fitting techniques, such as Hough Transform. This is a typical knowledge-based strategy, which fits known models to local image data, rather than estimate models from detected image primitives. The detected high-level features have impacts on the correct convergence of the surface, as described below.

As shown in Figure 7g, the high-level boundary features at the two transverse processes are detected, and we use them to modify the mean shape into a good initial front. We place two rectangular parallelepipeds in connection with the mean shape to form the initial front, as shown in Figure 7h and 7i. This is a general idea that is applicable to many applications: If we are given a true knowledge of a partial boundary, we can use

it as part of an arbitrary shape to augment the initial front. After the modification, the initial front is pulled to the actual boundary, which helps in avoiding the internal local minimums.

The high-level context features, as shown in Figure 7d and 7e, are used for the introduction of a new type of speed to affect level set evolution—namely the blocker speed. The blocker speed serves as an energy potential that prevents the surface from evolving beyond the detected context feature. With this we can prevent the leakage to neighboring objects. As shown in Figure 7f, the two plane context features divide the space into two regions. Using the center of the spinal channel as the reference, we can determine the region which the surface should not evolve into. For that region, the speed terms are made negative, which prevents the surface from evolving beyond the context features. Note that the high-level context features do not belong to object boundary; rather, they belong to background. The surface will not converge to the context feature, but will converge to the object boundary that is on the inner side of the context feature.

From the above, the leakage and local minimum issues that frequently bother a level set surface evolution are principally addressed. Finally, the separation planes associated with the mean shape are also mapped to the image data. Local primitive detection is performed to generate the speed terms specially designed for each region. Careful design of the speed is also an important part of this knowledge-based segmentation scheme.

As shown in Figure 6n-6s, after iterations of surface evolution, the level set surface converges to the correct object boundaries despite the internal inhomogeneity, the weak boundaries and the presence of neighboring structures. From this example we showed the power of prior knowledge when it is well represented, organized and applied to medical image segmentation. We only presented high level concepts in this example, and the reader is referred to the work of Shen et al. (2005) for a detailed treatment including mathematical deductions.

CONCLUSIONS, DISCUSSION, AND OUTLOOK

We have made an attempt in this chapter to give an intuitive introduction to the various methods and frameworks that are useful in 3D medical image segmentation. We also discussed the concepts of our approach of knowledge-based segmentation, in which the effective representation and application of prior knowledge is most important. From the discussion, the most prominent high-level features are of most value and need to be intensively explored before other image properties come into play.

The state-of-the-art medical image segmentation methods still leave much to be desired. Other than the difficulties of the segmentation problem itself, the relatively short time in which the 3D medical data become available is also one factor. The earlier scanners produced data in which the axial resolution is much lower than the other two directions. Therefore 3D correlation between two adjacent axial slices is weak, and a 3D segmentation method is not particularly more beneficial than 2D segmentation on each slice image. In the past two years, multi-slice CT scanners have achieved true isotropic voxels. With the development of medical instrument technology, 3D medical image segmentation will become increasingly popular. 3D methods are inherently better for the segmentation of

complex structures, in that they implicitly involve the 3D structural information that is difficult to capture in 2D methods. On the other hand, it will still take time for people to have full awareness of the properties of these data and develop insights.

Medical imaging applications have now become an important driving force for the advance of computer vision. On the other hand, medical image analysis needs to address real-word issues that have been outside the realm of computer vision. These issues come largely from the fact that the end systems are mostly used by the physician. The human factor is essential, since any successful solution will have to be accepted by a physician and integrated into one's medical procedural workflow. This put strong constraints on the type of applicable methods. Because of this, there has been a discrepancy between the advanced frameworks presented in computer vision and the low-level and ad-hoc methods used by researchers working on real medical application solutions. Recently, we have seen the hope of practically applicable frameworks which are relatively simple in implementation, and reasonable in performance and robustness. In this respect, we foresee several trends. First, existing and coming frameworks from computer vision will improve themselves to be more practical and compete for their prevalence in medical image analysis. Second, more medical imaging researchers will make use of these frameworks, with their efforts focused on the incorporation of application-specific priors. Third, more efforts from the computer vision community will be attracted to solving real-world problems, and, hence, lead to the invention of more general methods for prior knowledge incorporation. As a byproduct, general frameworks will also be invented for the incorporation of user interaction into difficult segmentation problems.

ACKNOWLEDGMENTS

I would like to thank my colleagues at Siemens Corporate Research: Dr. Gozde Unal, Dr. Greg Slabaugh, Dr. Leo Grady, Dr. Li Zhang, Dr. Zhizhou Wang, Zhigang Peng, Dr. Marie-Pierre Jolly, Ludek Hasa and Julien Nahed for their support in writing this chapter. Gratitude also goes to Dr. Khalid Al-Kofahi from Thomson Legal & Regulatory and Yonggang Shi from Boston University. They have provided useful images and references, and most importantly, insightful discussions, without which this chapter would not be possible.

REFERENCES

- Adalsteinsson, D., & Sethian, J. (1995). A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118, 269-277.
- Adams, R., & Bischof, L. (1984). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6), 641-647.
- Al-Kofahi, K., Lasek, S., Szarowski, D., Pace, C., Nagy, G., Turner, J.N., et al. (2002). Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Transactions on Information Technology in Biomedicine*, 6(2), 171-187.
- Baillard, C., Hellier, P., & Barillot, C. (2001). Segmentation of brain images using level sets and dense registration. *Medical Image Analysis*, 5, 185-194.

- Basak, J., Chanda, B., & Majumder, D. D. (1994). On edge and line linking with connectionist models. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(3), 413-428.
- Brejl, M., & Sonka, M. (2000). Directional 3D edge detection in anisotropic data: Detector design and performance assessment. *Computer Vision and Image Understanding*, 77, 84-110.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
- Chakraborty, A., Staib, L. H., & Duncan, J. S. (1996). Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions on Medical Imaging*, 15(6), 859-870.
- Chan, T., & Vese, L. (2001). An active contour model without edges. *IEEE Transactions on Image Processing*, 10(2), 266-277.
- Chang, Y. L., & Li, X. (1994). Adaptive region growing. *IEEE Transactions on Image Processing*, 3(6), 868-72.
- Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 681-685.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., & Granham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61, 38-59.
- Freeman, W. T., & Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9), 891-906.
- Guy, G., & Medioni, G. (1996). Inferring global perceptual from local features. *International Journal of Computer Vision*, 20(1/2), 113-133.
- Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4), 532-50.
- Herman, G. T., & Carvalho, B. M., (2001). Multi-seeded segmentation using fuzzy connectedness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5), 460-474.
- Hill, A., & Taylor, C. J. (1994). Automatic landmark generation for point distribution models. *Proceedings of the 5th British Machine Vision Conference*, 2(2), 429-438.
- Horkaew, P., & Yang, G. Z. (2003). Optimal deformable surface models for 3D medical image analysis. *IPMI*, 13-24.
- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Active contour models. *International Journal of Computer Vision*, 1(4), 321-331.
- Leventon, M. E., Grimson, W. E., & Faugeras, O. (2000). Statistical shape influence in geodesic active contours. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 1(1), 316-323.
- Lorenz, C., & Krahnstover, N. (1999). 3D statistical shape models for medical image segmentation. *International Conference on 3-D Digital Imaging and Modeling*, (pp. 414-423).
- Malladi, R., Sethian, J. A., & Vemuri, B. C. (1995). Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2), 158-75.
- Marroquin, J. L., Vemuri, B. C., Botello, S., & Calderon, F. (2002). An accurate and efficient bayesian method for automatic segmentation of brain MRI. *Proceedings of ECCV* (pp. 560-574).

- Meyer, F. & Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1), 21-46.
- Mitchell, S. C., Bosch, J. G., Lelieveldt, B. P. F., van de Geest, R. J., Reiber, J. H. C., & Sonka, M. (2002). 3-D active appearance models: segmentation of cardiac MR and ultrasound images. *IEEE Transactions on Medical Imaging*, 21(9), 1167-1178.
- Mohan, R., & Nevatia, R. (1989). Using perceptual organization to extract 3-D structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11), 1121-1139.
- Pentland, A. P., & Sclaroff, S. (1991). Closed-form solutions for physically based modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7), 715-729.
- Sarkar, S., & Boyer, K. L. (1994). Computing perceptual organization in computer vision, *World Scientific Series on Machine Perception and Artificial Intelligence*. Singapore: World Scientific.
- Shankar, B. U., Murthy, C. A., & Pal, S. K. (1998). A new gray level based Hough transform for region extraction, an application to IRS images, *Pattern Recognition Letters*, 19, 197-204.
- Shen, H., Liang, L., Shao, M., & Qing, S. (2004). Tracing based segmentation for labeling of individual rib structures in chest CT volume data. *Proceedings of the 7th International Conference on Medical Image Computing and Computer Assisted Intervention* (pp. 967-974).
- Shen, H., Shi, Y., & Peng, Z. (2005). Applying prior knowledge in the segmentation of 3D complex anatomic structures. *Computer Vision for Biomedical Image Applications: Current Techniques and Future Trends, An International Conference on Computer Vision Workshop, Lecture Notes of Computer Science*, 3765 (pp. 189-199). Beijing, China.
- Slabaugh, G., & Unal, G. (2005). Active polyhedron: Surface evolution theory applied to deformable meshes. *Conference on Computer Vision and Pattern Recognition*.
- Sonka, M., Hlavac, V., & Boyle, R. (1998). *Image processing, analysis, and machine vision* (2nd ed.). PWS Publishing.
- Sun, J., & Gu, D. (2004). Bayesian image segmentation based on an inhomogeneous hidden Markov random field. *Proceedings of the 17th International Conference on Pattern Recognition* (pp. 596-599).
- Suri, J. S., Liu, K., Singh, S., Laxminarayan, S. N., Zeng, X., & Reden, L. (2002). Shape recovery algorithm using level sets in 2-D/3-D medical imagery, a state-of-the-art review. *IEEE Transactions on Information Technology in Biomedicine*, 6(1), 8-28.
- Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., et al. (2003). A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2), 137-154.
- Udupa, J. K., Saha, P. K., & Lotufo, R. A. (2002). Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11), 1485-1500.
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), 583-598.

- Walker, K. N., Cootes, T. F., & Taylor, C. J. (2002). Automatically building appearance models from image sequences using salient features. *Image and Vision Computing*, 20(5-6), 435-440.
- Wu, Z., & Leahy, R. (1992). Image segmentation via edge contour finding. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 613-619).
- Yang, J., Staib, H.S., & Duncan, J.S. (2004). Neighbor-constrained segmentation with level set based 3-D deformable models. *IEEE Transactions on Medical Imaging*, 23(8), 940-948.
- Zhang, L., Zhang, T., Novak, C. L., Naidich, D. P., & Moses, D. A. (2005). A computer-based method of segmenting ground glass nodules in pulmonary CT images: Comparison to expert radiologists' interpretations. *Proceedings of SPIE Medical Imaging*.

Chapter XIII

Parallel Segmentation of Multi-Channel Images Using Multi-Dimensional Mathematical Morphology

Antonio Plaza, University of Extremadura, Spain

Javier Plaza, University of Extremadura, Spain

David Valencia, University of Extremadura, Spain

Pablo Martinez, University of Extremadura, Spain

ABSTRACT

Multi-channel images are characteristic of certain applications, such as medical imaging or remotely sensed data analysis. Mathematical morphology-based segmentation of multi-channel imagery has not been fully accomplished yet, mainly due to the lack of vector-based strategies to extend classic morphological operations to multidimensional imagery. For instance, the most important morphological approach for image segmentation is the watershed transformation, a hybrid of seeded region growing and edge detection. In this chapter, we describe a vector-preserving framework to extend morphological operations to multi-channel images, and further propose a fully automatic multi-channel watershed segmentation algorithm that naturally combines spatial and spectral/temporal information. Due to the large data volumes often associated with multi-channel imaging, this chapter also develops a parallel implementation strategy to speed up performance. The proposed parallel algorithm is evaluated using magnetic resonance images and remotely sensed hyperspectral scenes collected by the NASA Jet Propulsion Laboratory Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS).

INTRODUCTION

The segmentation of an image can be defined as its partition into different regions, each having certain properties (Zhang, 1996). In mathematical terms, a segmentation of an image f is a partition of its definition domain D_f into m disjoint, non-empty sets s_1, s_2, \dots, s_m called segments, so that $\bigcup_{i=1}^m S_i = D_f$ and $S_i \cap S_j = \emptyset, \forall i \neq j$. Segmentation of intensity images in the spatial domain usually involves four main approaches (Haralick & Shapiro, 1985). Thresholding techniques assume that all pixels whose value lies within a certain range belong to the same class. Boundary-based methods assume that the pixel values change rapidly at the boundary between two regions. Region-based segmentation algorithms postulate that neighboring pixels within the same region have similar intensity values, of which the split-and-merge technique is probably the most well known. Hybrid methods combine one or more of the above-mentioned criteria. This class includes variable-order surface fitting and active contour methods.

One of the most successful hybrid segmentation approaches is the morphological watershed transformation (Beucher, 1994), which consists of a combination of seeded region growing (Adams & Bischof, 1994; Mehnert & Jackway, 1997) and edge detection. It relies on a marker-controlled approach (Fan et al., 2001) that considers the image data as imaginary topographic relief; the brighter the intensity, the higher the corresponding elevation. Let us assume that a drop of water falls on such a topographic surface. The drop will flow down along the steepest slope path until it reaches a minimum. The set of points of the surface whose steepest slope path reaches a given minimum constitutes the *catchment basin* associated with that minimum, while the watersheds are the zones dividing adjacent catchment basins. Another way of visualizing the watershed concept is by analogy to immersion (Vincent & Soille, 1991). Starting from every minimum, the surface is progressively flooded until water coming from two different minima meet. At this point, a watershed line is erected. The watershed transformation can successfully partition the image into meaningful regions, provided that minima corresponding to relevant image objects, along with object boundaries, are available (Shafarenko et al., 1997). Despite its encouraging results in many applications, morphological techniques have not been fully exploited in applications that involve multi-channel imagery, where a vector of values rather than a single value is associated with each pixel location.

Many types of multi-channel images exist depending on the type of information collected for each pixel. For instance, color images are multi-channel images with three channels, one for each primary color in the RGB space. Images optically acquired in more than one spectral or wavelength interval are called multispectral. These images are characteristic in satellite imaging and aerial reconnaissance applications. The number of spectral channels can be extremely high, as in the case of hyperspectral images produced by imaging spectrometers (Chang, 2003). Finally, all image types above can be extended to the class of multitemporal images or image sequences, which consist of series of images defined over the same definition domain, but collected at more than a single time. Examples include magnetic resonance (MR) images in medical applications and video sequences.

Segmentation of multi-channel imagery has usually been accomplished in the spectral/temporal domain of the data only. Techniques include well known data clustering algorithms such as ISODATA (Richards & Jia, 1999). Other techniques, such as

Soille's watershed-based multi-channel segmentation (Soille, 1996), are based on an initial spectral clustering followed by a post-classification using spatial information. This approach separates spatial information from spectral information, and thus the two types of information are not treated simultaneously.

In this chapter, we develop a novel watershed-based segmentation technique that naturally combines spatial and spectral/temporal information in simultaneous fashion. While such an integrated approach holds great promise in several applications, it also creates new processing challenges (Tilton, 1999). In particular, the price paid for the wealth of spatial and spectral/temporal information is an enormous amount of data to be processed. For that purpose, we develop a parallel implementation of the proposed segmentation algorithm that allows processing of high-dimensional images quickly enough for practical use. The chapter is structured as follows. The following section provides a mathematical formulation for multi-dimensional morphological operations, and relates the proposed framework to other existing approaches in the literature. A multi-dimensional watershed-based segmentation approach is described next, along with its parallel implementation. A quantitative segmentation evaluation comparison with regard to standard segmentation techniques is then provided, using both MR brain images and remotely sensed hyperspectral data collected by the 224-channel NASA Jet Propulsion Laboratory Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) system. The chapter concludes with some remarks.

MULTI-DIMENSIONAL MATHEMATICAL MORPHOLOGY

Our attention in this section focuses primarily on the development of a mechanism to extend basic morphological operations to multi-channel imagery. In the following, we provide a mathematical formulation for classic and extended morphological operations.

Classic Morphological Operations

Following a usual notation (Soille, 2003), let us consider a grayscale image f , defined on a space E . Typically, E is the 2-D continuous space R^2 or the 2-D discrete space Z^2 . The classic erosion of f by $B \subset Z^2$ is given by the following expression:

$$(f \ominus B)(x, y) = \min_{(s,t) \in Z^2(B)} \{f(x+s, y+t)\}, \quad (x, y) \in Z^2 \quad (1)$$

where $Z^2(B)$ denotes the set of discrete spatial coordinates associated to pixels lying within the neighborhood defined by a “flat” SE, designed by B . The term “flat” indicates that the SE is defined in the x-y plane. Similarly, the classic dilation of f by B is given by:

$$(f \oplus B)(x, y) = \max_{(s,t) \in Z^2(B)} \{f(x-s, y-t)\} \quad (x, y) \in Z^2 \quad (2)$$

Vector-Based Mathematical Morphology

Let us denote a multi-channel image with n channels as \mathbf{f} , where the values of each pixel $f(x,y)$ of the definition domain D_f are given by an n -dimensional vector $\mathbf{f}(x,y) = (f_1(x,y), f_2(x,y), \dots, f_n(x,y))$. Extension of monochannel erosion and dilation above the images defined on the n -dimensional space is not straightforward. A simple approach consists in applying monochannel techniques to each image channel separately, an approach usually referred to as “marginal” morphology in the literature. This approach is unacceptable in many applications because, when morphological techniques are applied independently to each image channel, there is a possibility for loss or corruption of information of the image due to the probable fact that new pixel vectors—not present in the original image—may be created as a result of processing the channels separately. In addition, no correlation between spectral/temporal components is taken into account. An alternative way to approach the problem of multi-dimensional morphology is to treat the data at each pixel as a vector. Unfortunately, there is no unambiguous means of defining the minimum and maximum values between two vectors of more than one dimension, and thus it is important to define an appropriate arrangement of vectors in the selected vector space.

Several vector-ordering schemes have been discussed in the literature (Plaza et al., 2004). Four classes of ordering will be shortly outlined here for illustrative purposes. Let us now consider an n -dimensional image \mathbf{f} and let $\mathbf{f}(x,y)$ and $\mathbf{f}(x',y')$ denote two pixel vectors at spatial locations (x,y) and (x',y') respectively, with $\mathbf{f}(x,y) = (f_1(x,y), \dots, f_n(x,y))$ and $\mathbf{f}(x',y') = (f_1(x',y'), \dots, f_n(x',y'))$. In marginal ordering (M-ordering), each pair of observations $f_i(x,y)$ and $f_i(x',y')$ would be ordered independently along each of the n channels. In reduced ordering (R-ordering), a scalar parameter function g would be computed for each pixel of the image, and the ordering would be performed according to the resulting scalar values. The ordered vectors would satisfy the relationship $\mathbf{f}(x,y) \leq \mathbf{f}(x',y') \Rightarrow g[\mathbf{f}(x,y)] \leq g[\mathbf{f}(x',y')]$. In partial ordering (P-ordering), the input multivariate samples would be partitioned into smaller groups, which would then be ordered. In conditional ordering (C-ordering), the pixel vectors would be initially ordered according to the ordered values of one of their components, for example, the first component, $f_1(x,y)$ and $f_1(x',y')$. As a second step, vectors with the same value for the first component would be ordered according to the ordered values of another component, e.g., the second component, $f_2(x,y)$ and $f_2(x',y')$, and so on.

In this chapter, we adopt a new distance-based vector ordering technique (D-ordering), where each pixel vector is ordered according to its distance from other neighboring pixel vectors in the data (Plaza et al., 2002). This type of ordering, which can be seen as a special class of R-ordering, has demonstrated success in the definition of multi-channel morphological operations in previous work. Specifically, we define a cumulative distance between one particular pixel $\mathbf{f}(x,y)$ and all the pixel vectors in the spatial neighborhood given by B (B -neighborhood) as follows:

$$C_B(\mathbf{f}(x,y)) = \sum_{(s,t)} \text{Dist}(\mathbf{f}(x,y), \mathbf{f}(s,t)), \quad \forall (s,t) \in Z^2(B), \quad (3)$$

where Dist is a linear point-wise distance measure between two N -dimensional vectors. As a result, $C_B(\mathbf{f}(x,y))$ is given by the sum of Dist scores between $\mathbf{f}(x,y)$ and every other pixel vector in the B -neighborhood. To be able to define the standard morphological

operators in a complete lattice framework, we need to be able to define a *supremum* and an *infimum*, given an arbitrary set of vectors $S = \{v_1, v_2, \dots, v_p\}$, where p is the number of vectors in the set. This can be achieved by computing $C_B(S) = C_B(v_1), C_B(v_2), \dots, C_B(v_p)\}$ and selecting v_j , such that $C_B(v_j)$ is the minimum of $C_B(S)$, with $1 \leq j \leq p$. In similar fashion, we can select v_k such that $C_B(v_k)$ is the maximum of $C_B(S)$, with $1 \leq k \leq p$. Based on the simple definitions above, the flat extended erosion of f by B consists of selecting of the B -neighborhood pixel vector that produces the minimum C_B value:

$$(f \ominus B)(x, y) = \{f(x + s', y + t'), (s', t') = \arg \min_{(s, t) \in Z^2(B)} \{C_B(f(x + s, y + t))\}\}, (x, y) \in Z^2, \quad (4)$$

where the $\arg \min$ operator selects the pixel vector that is most similar, according to the distance Dist , to all the other pixels in the in the B -neighborhood. On other hand, the flat extended dilation of f by B selects the B -neighborhood pixel vector that produces the maximum value for C_B :

$$(f \oplus B)(x, y) = \{f(x - s', y - t'), (s', t') = \arg \max_{(s, t) \in Z^2(B)} \{C_B(f(x - s, y - t))\}\}, (x, y) \in Z^2, \quad (5)$$

where the $\arg \max$ operator selects the pixel vector that is most different, according to Dist , from all the other pixels in the B -neighborhood. Using the above notation, the multi-channel morphological gradient at pixel $f(x, y)$ using B can be simply defined as follows:

$$G_B(f(x, y)) = \text{Dist}((f \ominus B)(x, y), (f \oplus B)(x, y)). \quad (6)$$

It should be noted that the proposed basic multi-dimensional operators are vector-preserving in the sense that vectors which are not present in the input data cannot be generated as a result of the extension process (Plaza et al., 2002). Obviously, the choice of Dist is a key topic in the resulting multi-channel ordering relation. A common choice in remote sensing applications is the spectral angle (SAD), an illumination-insensitive metric defined between two vectors s_i and s_j as follows:

$$\text{SAD}(s_i, s_j) = \cos^{-1}(s_i \cdot s_j / \|s_i\| \|s_j\|) = \cos^{-1} \left(\sum_{l=1}^N s_{il} s_{jl} / \left[\left(\sum_{l=1}^N s_{il}^2 \right)^{1/2} \left(\sum_{l=1}^N s_{jl}^2 \right)^{1/2} \right] \right) \quad (7)$$

In medical imaging, for instance, illumination effects not as relevant as noise or other types of interferers. In those cases, a most common choice is the Euclidean distance (ED). In the following, we respectively adopt SAD and ED as the baseline distances for remote sensing and medical imaging experiments discussed in this chapter.

MULTI-CHANNEL WATERSHED SEGMENTATION ALGORITHM

The segmentation paradigm of our multi-channel watershed segmentation algorithm consists of three stages. First, multi-dimensional morphological operations are used to collect a set of minima according to some measure of minima importance. Starting from the selected minima and using the multi-dimensional morphological gradient as a reference, a multi-channel watershed transformation by flooding is then applied. Finally, watershed regions are iteratively merged, according to a similarity criterion, to obtain the final segmentation.

Minima Selection

The key of an accurate segmentation resides in the first step, that is, the selection of “markers,” or minima, from which the transform is started. Following a recent work (Malpica et al., 2003), we hierarchically order all minima according to their deepness, and then select only those above a threshold. This approach has the advantage that it provides an intuitive selection scheme controlled by a single parameter. The concept can be easily explained using the immersion simulation. The deepness of a basin would be the level the water would reach, coming in through the minimum of the basin, before the water would overflow into a neighboring basin, that is, the height from the minimum to the lowest point in the watershed line of the basin.

Deepness can be computed using morphological reconstruction applied to the multi-channel gradient in Equation 6. Reconstruction is a special class of morphological transformation that does not introduce discontinuities (Vincent, 1993). Given a “flat” SE of minimal size, designed by B , and the multi-channel gradient $G_B(f)$ of an n -dimensional image f , morphological reconstruction by erosion of $G_B(f)$ using B can be defined as follows:

$$(G_B(f) \otimes B)^t(x, y) = \bigvee_{k \geq 1} [\delta_B^t(G_B(f) \otimes B | G_B(f))] (x, y), \quad (8)$$

where

$$[\delta_B^t(G_B(f) \otimes B | G_B(f))] (x, y) = \left[\overbrace{\delta_B \delta_B \cdots \delta_B}^{t \text{ times}} (G_B(f) \otimes B | G_B(f)) \right] (x, y) \quad (9)$$

and

$$[\delta_B(G_B(f) \otimes B | G_B(f))] (x, y) = \bigvee \{(G_B(f) \otimes B)(x, y), G_B(f(x, y))\} \quad (10)$$

In the above operation, $G_B(f) \otimes B$ is the standard erosion of the multi-channel gradient image, which acts as a “marker” image for the reconstruction, while $G_B(f)$ acts as a “mask” image. Reconstruction transformations always converge after a finite number of iterations t , that is, until the propagation of the marker image is totally impeded by the mask image. It can be proven that the morphological reconstruction $(G_B(f) \otimes B)^t$ of $G_B(f)$

) from $G_B(f) \otimes B$ will have a watershed transform in which the regions with deepness lower than a certain value v have been joined to the neighboring region with closer spectral properties, that is, parameter v is a minima selection threshold.

Flooding

In this section, we formalize the flooding process following a standard notation (Soille, 2003). Let the set $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k\}$ denote the set of k minimum pixel vectors selected after multi-dimensional minima selection. Similarly, let the catchment basin associated with a minimum pixel \mathbf{p}_i be denoted by $CB(\mathbf{p}_i)$. The points of this catchment basin which have an altitude less than or equal to a certain deepness score d (Malpica et al., 2003) are denoted by:

$$CB_d(\mathbf{p}_i) = \{f(x, y) \in CB(\mathbf{p}_i) \mid Dist(\mathbf{p}_i, f(x, y)) \leq d\} \quad (11)$$

We also denote by $X_d = \bigcup_{i=1}^k CB_d(\mathbf{p}_i)$ the subset of all catchment basins which contain a pixel vector with a deepness value less than or equal to d . Finally, the set of points belonging to the regional minima of deepness d are denoted by $RMIN_d(f(x, y))$. The catchment basins are now progressively created by simulating the flooding process. The first pixel vectors reached by water are the points of highest deepness score. These points belong to $RMIN_{p_j}(f(x, y)) = X_{D_B(\mathbf{p}_j)}$, where \mathbf{p}_j is the deepest pixel in P , that is, $D_B(\mathbf{p}_j)$ is the minimum, with $1 \leq j \leq k$. From now on, the water either expands the region of the catchment basin already reached by water, or starts to flood the catchment basin whose minima have a deepness equal to $D_B(\mathbf{p}_l)$, where \mathbf{p}_l is the deepest pixel in the set of $P - \{\mathbf{p}_j\}$. This operation is repeated until $P = \emptyset$. At each iteration, there are three possible relations of inclusion between a connected component Y and $Y \cap X_{D_B(\mathbf{p}_j)}$:

1. If $Y \cap X_{D_B(\mathbf{p}_j)} = \emptyset$ then it follows that a new minimum Y has been discovered at level $D_B(\mathbf{p}_l)$. In this case, the set of all minima at level $D_B(\mathbf{p}_l)$, that is, $RMIN_{p_l}(f(x, y))$ will be used for defining $X_{D_B(\mathbf{p}_l)}$.
2. If $Y \cap X_{D_B(\mathbf{p}_j)} \neq \emptyset$ and is connected, then the flooded region is expanding and Y corresponds to the pixels belonging to the catchment basin associated with the minimum and having a deepness score less than or equal to $D_B(\mathbf{p}_j)$, that is, $Y = CB_{D_B(\mathbf{p}_j)}(Y \cap X_{D_B(\mathbf{p}_j)})$.
3. Finally, if $Y \cap X_{D_B(\mathbf{p}_j)} \neq \emptyset$ and is not connected, then the flooded regions of the catchment basins of two distinct minima at level $D_B(\mathbf{p}_j)$ are expanding and merged together.

Once all levels have been flooded, the set of catchment basins of a multi-dimensional image f is equal to the set $X_{D_B(\mathbf{p}_m)}$, where \mathbf{p}_m is the least deep pixel in P , that is, $D_B(\mathbf{p}_m)$ is the maximum, with $1 \leq m \leq k$. The set of catchment basins after multi-dimensional watershed can be represented as a set $\{CB(\mathbf{p}_i)\}_{i=1}^k$, where each element corresponds to the catchment basin of a regional minimum of the multi-channel input image f . This is the final segmentation output for the algorithm. A parallel algorithm for implementing the proposed flooding simulation is described in the following section.

Region Merging

To obtain the final segmentation, some of the regions $\{\text{CB}(\mathbf{p}_i)\}_{i=1}^k$ resulting from the watershed can be merged to reduce the number of regions (Le Moigne & Tilton, 1995). This section briefly explains the region merging criteria and method employed. First, all regions are ordered into a region adjacency graph (RAG). The RAG is an undirected graph $G=(V,E)$, where $V=\{\text{CB}(\mathbf{p}_i)\}_{i=1}^k$ such that each region $\text{CB}(\mathbf{p}_i)$ is represented by a node, and $e(\mathbf{p}_i, \mathbf{p}_j) \in E$ if:

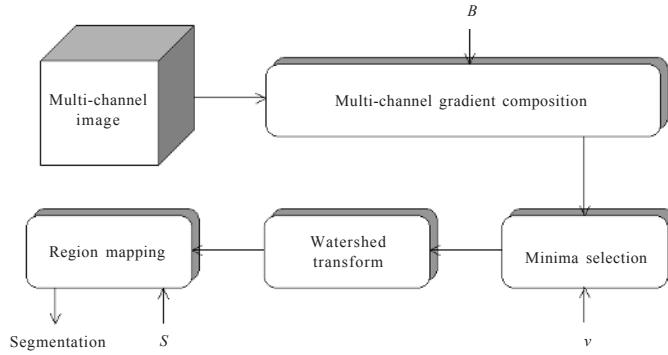
1. $\mathbf{p}_i, \mathbf{p}_j \in V$, and
2. $\text{CB}(\mathbf{p}_i), \text{CB}(\mathbf{p}_j)$ are adjacent, or
3. $\text{Dist}(\mathbf{p}_i, \mathbf{p}_j) < S$, where S is a pixel vector similarity threshold.

The merging process is based on graph G , where the weight of an edge $e(\mathbf{p}_i, \mathbf{p}_j)$ is the value of $\text{Dist}(\mathbf{p}_i, \mathbf{p}_j)$. Regions $\text{CB}(\mathbf{p}_i), \text{CB}(\mathbf{p}_j)$ can be merged attending to spatial properties in the case of adjacent regions, and also according to pixel vector similarity criteria in the case of non-adjacent regions. Similar merging procedures have been successfully used before in the literature (Tilton, 1999). Finally, Kruskal's algorithm can be applied to generate the minimum spanning tree, denoted as T , by adding one edge at a time. Initially, the edges of G are sorted in a non-decreasing order of their weights. Then, the edges in the sorted list are examined one-by-one and checked to determine whether adding the edge that is currently being examined creates a cycle with the edges that were already added to T . If it does not, it is added to T ; otherwise, it is discarded. It should be noted that adding $e(\mathbf{p}_i, \mathbf{p}_j)$ to T represents the merge of its two regions $\text{CB}(\mathbf{p}_i)$ and $\text{CB}(\mathbf{p}_j)$. On other hand, adding the edge with the minimum weight one-by-one in an increasing order to T using the sorted list is equivalent to the merge of the two most similar regions. Finally, when an edge is rejected because it creates a cycle in T , no merge is performed because its two regions have already been merged into one. The process is terminated when T contains k edges.

In order to summarize the different stages and parameters involved, Figure 1 shows a block diagram depicting our multi-dimensional morphological algorithm for segmentation of multi-channel images.

As noted, there are three input parameters: B , a “flat” SE of minimal size used in the morphological operations; v , a minima selection threshold used in the minima selection process and S , a pixel vector similarity threshold used in the region merging stage. First, a multi-channel gradient computation is performed by taking advantage of extended morphological operations. This step works as a multi-channel edge detector. Second, minima are selected from the resulting output by using the concept of deepness. Third, flooding from markers is accomplished by utilizing the spectral angle between pixel vectors. This operation makes use of the full spectral information (as opposed to traditional watershed-based segmentation algorithms), thus avoiding the problem of band selection from the input data. Finally, the resulting segmentation is refined by a region-merging procedure that integrates the spatial and spectral information. As can be deduced from the description above, one of the main contributions of the proposed algorithm is the fact that it naturally combines spatial/spectral information in all steps. The algorithm is fully automatic, and produces a segmentation output given by a set of

Figure 1. Block diagram summarizing the multichannel watershed segmentation algorithm



watershed regions after region merging that we will denote from now on as $\{WS_i\}_{i=1}^m$, with $\bigcup_{i=1}^m WS_i = D_f$ and $WS_i \cap WS_j \neq \emptyset, \forall i \neq j$.

PARALLEL IMPLEMENTATION

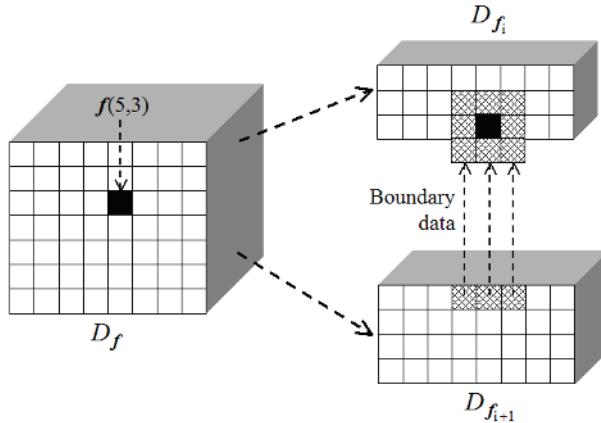
Parallelization of watershed algorithms that simulate flooding is not a straightforward task. From a computational point of view, these algorithms are representative of the class of irregular and dynamic problems (Moga & Gabbouj, 1997). Moreover, the watershed process has an extremely volatile behavior, starting with a high degree of parallelism that very rapidly diminishes to a much lower degree. In this section, our goal is to develop an efficient and scalable parallel implementation of the algorithm proposed in the previous section.

Partitioning

Two types of partitioning strategies can be applied to image segmentation problems (Seinstra et al., 2002). One may decide whether the computation associated with the given problem should be split into pieces (functional decomposition), or the data on which the computation is applied (domain decomposition). Functional decomposition is inappropriate for segmentation with watersheds, where a sequence of operators are applied in a chain to the entire image. Our parallel algorithm uses domain decomposition, that is, the original multi-channel image f is decomposed into subimages. It is important to note that the subimages are made up of entire pixel vectors, that is, a single pixel vector is never split amongst several processing elements (PEs). If the computations for each pixel vector need to originate from several PEs, then they would require intensive inter-processor message passing (Montoya et al., 2003). Thus, the global domain D_f is split among P processors in disjoint subdomains as follows:

$$D_f = D_{f_0} \cup D_{f_1} \cup \dots \cup D_{f_{P-1}}, \text{ with } D_{f_i} \cap D_{f_j} = \emptyset, \forall i \neq j \quad (12)$$

Figure 2. Morphological structuring element computation split between two processors



Task Replication

An important issue in SE-based morphological image processing operations is that accesses to pixels outside the domain D_f of the input image is possible. For instance, when the SE is centered on a pixel located in the border of the original image, a simple border-handling strategy can be applied (Seinstra et al., 2003). On the other hand, additional inter-processor communications may be required when the structuring element computation needs to be split amongst several different processing nodes, as shown by Figure 2. In the example, the computations for the pixel vector at spatial location (5,3) needs to originate from two processors. In order to avoid such an overhead, edge/corner pixels are replicated in the neighboring processors whose subdomains are thus enlarged with a so-called extension area. The extended subdomains are overlapping, and can be defined as follows:

$$D_{f_i}^e = \{f(x,y) \in D_f \mid N_B(f(x,y)) \cap D_{f_i} \neq \emptyset\}, \quad (13)$$

where $N_B(f(x,y))$ is the B -neighborhood of $f(x,y)$. Using the above notation, we can further denote the neighboring subimages of f_i as the set $N_B(f_i) = \{f_j \mid D_{f_j} \cap D_{f_i}^e \neq \emptyset\}$. It should be noted that the task-replication strategy above enhances code reusability, which is highly recommended in order to build a robust parallel algorithm. As will be shown by experiments, the amount of redundant information introduced by the proposed framework can greatly reduce communication time.

Implementation

Our implementation of the parallel multi-channel watershed algorithm uses a simple master-slave model. The master processor divides the multi-channel image f into a set of subimages f_i which are sent to different processors, so that the domain of each subimage is an extended subdomain given by $D_{f_i}^e$. The slave processors run the segmentation

algorithm on the respective subimages and also exchange data among themselves for uniform segmentation. After the segmented regions become stable, the slaves send the output to the master, which combines all of them in a proper way and provides the final segmentation. If we assume that the parallel system has p processors available, then one of the processors is reserved to act as the master, while each of the remaining $p-1$ processors create a local queue Q_i with $1 \leq i \leq p-1$. The minima selection algorithm is run locally at each processor to obtain a set of minima pixels surrounded by non-minima, which are then used to initialize each queue Q_i . Flooding is then performed locally in each processor as in the serial algorithm. However, due to the image division, flooding is confined only to the local subdomain. Therefore, there may exist parts of the subimage that cannot be reached by flooding since they are contained in other subimages. Our approach to deal with this problem is to first flood locally at every deepness score in the subimage. Once the local flooding is finished, each processor exchanges segmentation labels of pixels in the boundary with appropriate neighboring processors. Subsequently, a processor can receive segmentation labels corresponding to pixels in the extended subdomain. The processor must now “reflood” the local subdomain from those pixels, a procedure that may introduce changes in segmentation labels of the local subdomain. Communication and reflooding are again repeated until stabilization (i.e., no more changes occur). When the flood-reflood process is finished, each slave processor sends the final segmentation labels to the master processor, which combines them together and performs region merging to produce final set of segmentation labels.

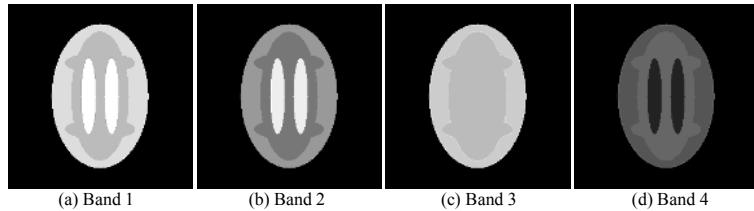
To conclude this section, we emphasize that although the flooding-reflooding scheme above seems complex, we have experimentally tested that this parallelization strategy is more effective than other approaches that exchange segmentation labels without first flooding locally at every deepness score. The proposed parallel framework guarantees that the processors are not tightly synchronized (Moga & Gabbouj, 1998). In addition, the processors execute a similar amount of work at approximately the same time, thus achieving load balance. Performance data for the parallel algorithm are given in the following subsection.

EXPERIMENTAL RESULTS

This section reports on the effectiveness of the proposed parallel segmentation algorithm in two specific applications. In the first one, phantom and real MR brain images are used to investigate the accuracy of multi-channel watershed segmentation in computer-aided medical diagnoses. In the second application, hyperspectral data collected by NASA’s Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) are used to illustrate a remote sensing classification scenario.

MRI Brain Image Experiments

In medical applications, the detection and outlining of boundaries of organs and tumors in magnetic resonance imaging (MRI) are prerequisite. This is one of the most important steps in computer-aided surgery. In this section, we present two sets of experiments, one set of computer-generated phantom images and another set of real MRI images, to show that the proposed algorithm has a good capability of segmentation.

Figure 3. Four band test phantoms for MRI simulation study*Table 1. Gray level values of the tissues of each band of the multichannel MRI phantom image*

Band	GM	WM	CSF
1	209	180	253
2	150	124	232
3	207	188	182
4	95	94	42

Computer Simulations for Phantom Experiments

In this subsection, computer simulations are used to conduct a quantitative study and performance analysis of the proposed multi-channel watershed algorithm. The computer-generated phantom MRI images, shown in Figure 3, consist of four bands. The ellipses represent structural areas of three interesting cerebral tissues corresponding to gray matter (GM), white matter (WM) and cerebral spinal fluid (CSF). From the periphery to the center, the distribution of tissues is simulated as follows: background (BKG), GM, WM and CSF, given by the gray level values in Table 1. The gray level values of these areas in each band were simulated in such a fashion that these values reflect the average values of their respective tissues in real MRI images. A zero-mean Gaussian noise was added to the phantom images in Figure 3 to achieve different levels of signal-to-noise ratios (SNRs) ranging from 10 dB to 30 dB. Despite the fact that such MRI phantom images may be unrealistic, they only serve for the purpose of illustration of the proposed algorithm. This is done by using available absolute ground-truth information at a pixel level, known from the controlled simulation scenario in which the data were simulated.

In order to assess contour-fitting precision of the proposed multi-channel watershed algorithm, we use the following statistical measures (Hoover et al., 1996): correct detection, over-segmentation, under-segmentation and missed and noise region. Let D be the total number of regions detected by the algorithm, and let G be the number of ground-truth regions (four in the phantom example). Let the number of pixels in each detected region, D_i , be denoted as P_{D_i} . Similarly, let the number of pixels in each ground-truth region, G_i , be denoted as P_{G_i} . Let $\theta_{D_i G_i} = P_{D_i} \cap P_{G_i}$ be the number of overlapped pixels between P_{D_i} and P_{G_i} . Thus, if there is no overlap between P_{D_i} and P_{G_i} , then $\theta_{D_i G_i} = \emptyset$, while if there is complete overlap, then $\theta_{D_i G_i} = P_{D_i} = P_{G_i}$. Let a threshold value T be a measure of

the strictness of the definition desired. With the above definitions in mind, the following segmentation accuracy metrics can be defined:

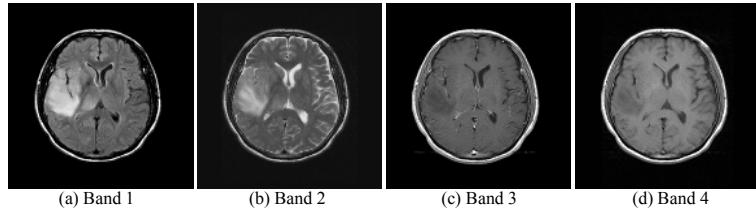
1. A pair made up of a detected region D_i and a ground-truth region G_i are classified as an instance of correct detection if $O_{D_i G_i} / P_{D_i} \geq T$, that is, at least T percent of the pixels in the detected region D_i are overlapped with the ground-truth region G_i ;
2. A ground-truth region G_i and a set of detected regions $D_j, j=1,\dots,n$ are classified as an instance of under-segmentation if: $\sum_{j=1}^n O_{D_j G_i} / P_{G_i} \geq T$, that is, the ground-truth region G_i is at least T -percent overlapped with the composition of the n detected regions, and, $\forall j \in [1, \dots, n] O_{D_j G_i} / P_{G_i} \geq T$, that is, all of the detected regions D_j are at least T -percent overlapped with the ground-truth region G_i ;
3. A set of ground-truth regions $G_j, j=1,\dots,m$ and a detected region D_i are classified as an instance of over-segmentation if: $\sum_{j=1}^m O_{D_i G_j} / P_{D_i} \geq T$, that is, the detected region D_i is at least T -percent overlapped with the composition of the m ground-truth regions, and, $\forall j \in [1, \dots, m] O_{D_i G_j} / P_{D_i} \geq T$, that is, all of the ground truth regions G_j are at least T -percent overlapped with the detected region D_i ;
4. A detected region D_i , not participating in any instance of correct detection, over-segmentation or under-segmentation, is classified as a missed region; and
5. A ground-truth region G_j not participating in any instance of correct detection, over-segmentation or under-segmentation is classified as noise region.

Using the five segmentation accuracy metrics above, Table 2 shows the number of correct detections, over-segments, under-segments, missed and noise regions obtained after applying the multi-channel watershed algorithm to the phantom image in Figure 3, corrupted by Gaussian noise in different proportions. The parameters used in experimental results were $B = B_5^{(\text{disk})}$, that is, a disk-shaped SE of radius equal to 5 pixels; v , a minima selection threshold automatically calculated from the data using the multi-level Otsu method (Plaza et al., 2002) and a pixel vector similarity threshold S that was set to 0.01 in experiments. The above values were selected empirically, although we

Table 2. Number of correct detections (C), over-segments (O), under-segments (U), missed (M) and noise (N) regions obtained after applying the multichannel watershed algorithm, using different tolerance (T) values, to the phantom image in Figure 3 corrupted by noise in different proportions

T	SNR = 30 dB					SNR = 20 dB					SNR = 10 dB				
	C	O	U	M	N	C	O	U	M	N	C	O	U	M	N
95%	4	0	0	0	0	2	1	1	0	2	2	0	2	0	4
90%	4	0	0	0	0	3	0	1	0	2	3	0	1	0	3
80%	4	0	0	0	0	4	0	0	0	1	3	0	1	0	1
70%	4	0	0	0	0	4	0	0	0	0	3	0	1	0	0
60%	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0
50%	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0

Figure 4. Four spectral bands of a real MRI brain image

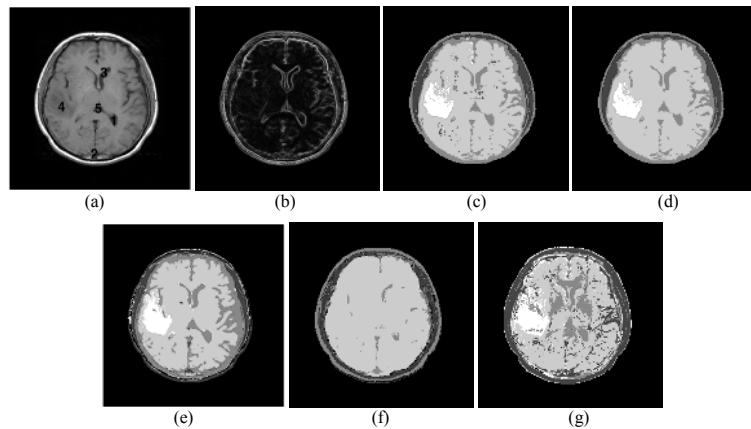


experimentally observed that the algorithm behaved similarly with other parameter settings. In addition, several tolerance threshold values were considered in the computation of the five statistical metrics above. As shown by the table, most of the regions detected by the multi-channel watershed algorithm were labeled as correct detections, even for very high tolerance thresholds or signal-to-noise ratios. The above results demonstrated that the proposed algorithm could produce accurate results in the presence of noise by integrating the information available from all the available channels in combined fashion.

Real MRI Image Experiments

In the following experiments, a real MRI multi-channel image was used for performance evaluation. Figure 4 shows the four different slices of the considered case study. Since in many MRI applications the three cerebral tissues, GM, WM and CSF, are of major interest, Figure 5a shows five markers with higher *deepness* associated with these regions. They are shown as numbers superimposed on band 3 of the real image, where the order relates to their deepness score (the pixel labeled as “1” has the highest deepness). On other hand, Figure 5b shows the morphological gradient obtained for this case study. The result of the multi-channel flooding-based watershed algorithm (before region merging) from the markers in Figure 5a is given in Figure 5c, where the considered parameter values were the same as those used for phantom experiments in the previous subsection. The final result after the region merging step is given in Figure 5d. For illustrative purposes, Figures 5e-g, respectively, show the segmentation result obtained by other standard algorithms. The first one is a standard single-channel watershed algorithm (Rajapakse et al., 1997) applied to the band with higher contrast, that is, band 2 in Figure 4b. The second one is a watershed clustering-based technique applied to the spectral feature space (Soille, 1996). This approach differs from our proposed combined method in that it separates the spectral and the spatial information. It first clusters the data in spectral space and then segments objects assuming that the pixels in a given class should have relatively homogeneous spectral intensities. The third approach is the well known ISODATA segmentation procedure that is regarded as a benchmark for most unsupervised segmentation algorithms. The algorithm uses the Euclidean distance as a similarity measure to cluster data elements into different classes (Richards & Jia, 1999).

Figure 5. (a) Five markers with highest deepness score, represented by numbers superimposed on band 3; (b) Multichannel morphological gradient; (c) Multichannel watershed segmentation (before merging); (d) Multichannel watershed segmentation (after merging); (e) Single-channel watershed segmentation; (f) Soille's watershed-based segmentation; (g) ISODATA segmentation



As observed in Figure 5, the multi-channel watershed algorithm in Figures 5c-d has several advantages over the other segmentation approaches tested. First, the proposed method was able to impose smoothness in the segmentation result. The segmentations produced by other methods often lack spatial consistency. For instance, the single-channel watershed algorithm in Figure 5e produced watershed lines which were rather noisy and jagged, even after region merging. This was because only one band was used, and the segmentation was dependent on intensity values at that particular band. In this work we selected the band with higher contrast for the single-channel watershed run; this selection might not be optimal in other cases. On the other hand, the watershed-based clustering approach in Figure 5f produced a smoother segmentation as a consequence of the (separate) use of spatial and spectral information. However, this approach missed some important regions clearly visible in the results produced by all other algorithms. Finally, the ISODATA segmentation in Figure 5g was produced using the spectral information only, that is, a pixel was classified depending on its spectral values whatever those of its neighbors. This resulted in a rather noisy and disconnected output. As can be seen in Figures 5c and 5d, the region merging stage implemented in the proposed method improved the segmentation by associating together some disconnected regions resulting from the flooding. Overall, results in Figure 5 demonstrate the importance of using spatial and spectral information in simultaneous fashion.

Finally, it should be noted that no quantitative analysis was conducted for the real MRI experiments due to the lack of ground-truth information. However, a visual evaluation of the results in Figure 5 by an expert radiologist indicated a “better

delineation of objects and superior spatial consistency of multi-channel watershed over the other tested approaches.” As for computational complexity, we used a PC with an AMD Athlon 2.6 GHz processor and 512 Megabytes of RAM to run all the experiments in this section, and it was found that the multi-channel watershed algorithm produced segmentation results in less than 30 seconds in all cases. This was mainly due to the limited number of bands available. Subsequently, no parallelization strategies were deemed necessary for computer-aided diagnosis of MRI image experiments. However, multi-channel image data in other applications, such as hyperspectral images in remote sensing, are characterized by high-dimensional images with hundreds of spectral bands. As a result, further experimentation using real hyperspectral data sets with high dimensionality is pertinent.

Remotely Sensed Hyperspectral Data Experiments

The image data set used in experiments is a hyperspectral scene collected by the 224-band Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), a remote sensing instrument operated by NASA’s Jet Propulsion Laboratory (Green et al., 1998). The data set was collected over an agricultural test site located in Salinas Valley, California, and represents a challenging segmentation problem. Fortunately, extensive ground-truth information is available for the area, allowing a quantitative assessment in terms of segmentation accuracy. Figure 6a shows the entire scene and a sub-scene of the dataset (called hereinafter Salinas A), dominated by directional regions. Figure 6b shows the 15 available ground-truth regions. The available data volume (over 50 Mb) creates the need for parallel watershed-based analysis able to produce segmentation results quickly enough for practical use.

Table 3 displays the number of correct detections, over-segmentations, under-segmentations and missed regions obtained after applying the proposed multi-channel segmentation algorithm using disk-shaped SEs with different radii, measured using different tolerance thresholds. For illustrative purposes, results by two other standard algorithms, ISODATA and Soille’s watershed-based clustering, are also displayed. It should be noted that the statistics for noise regions are not provided on purpose, due to the fact that available ground-truth information displayed in Figure 6b is not absolute. In all cases, parameter v was set automatically using the multi-level Otsu method, and parameter S was set to 0.01 empirically. As shown by Table 3, the use of appropriate structuring element sizes in the proposed method produced segmentation results which were superior to those found by ISODATA and Soille’s watershed-based clustering algorithm. In particular, the best results were obtained when a disk-shaped SE $B = B_{15}^{(\text{disk})}$ was used. This is mainly due to the relation between the SE and the spatial properties of regions of interest in the scene. Specifically, the usage of $B_{15}^{(\text{disk})}$ resulted in a number of correct detections (11) which was the highest one in experiments, while the scores for all error metrics were minimized. Interestingly, no under-segmentation or missed instances were obtained in this case, while only one case of over-segmentation was observed. This case comprised the four lettuce romaine regions contained in the Salinas A subscene, which are at different weeks since planting (4, 5, 6 and 7 weeks, respectively), and covering the soil in different proportions. These four ground-truth regions were always detected as a single, over-segmented region, which is a reasonable segmentation

Figure 6. (a) Spectral band at 488 nm of an AVIRIS hyperspectral image comprising several agricultural fields in Salinas Valley, California, and a sub-scene of the dataset (Salinas A), outlined by a white rectangle; (b) land-cover ground truth regions

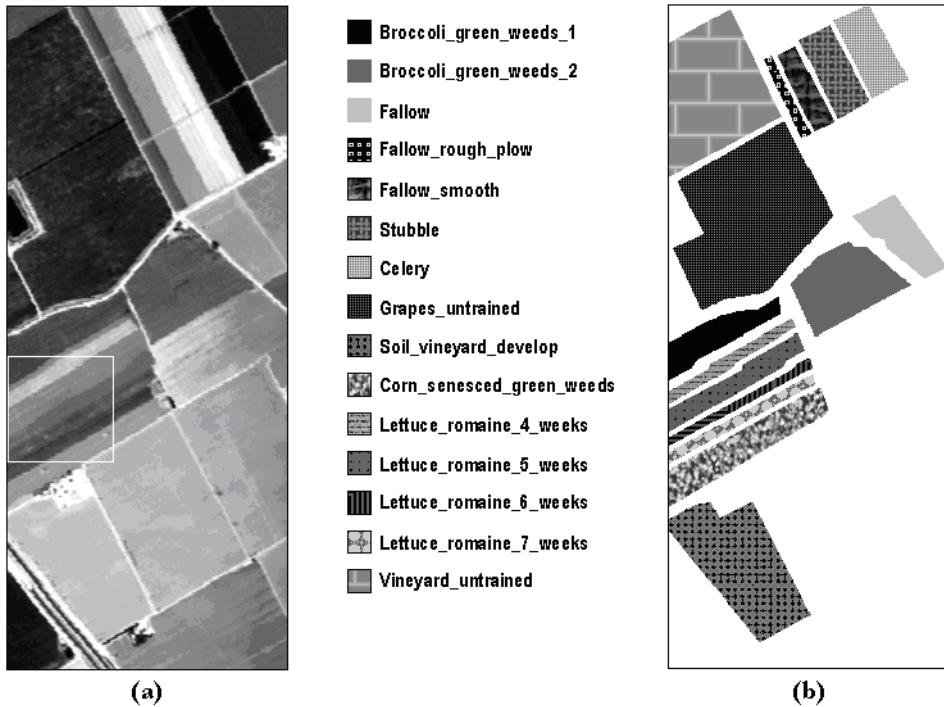


Table 3. Number of correct detections (C), over-segmentations (O), under-segmentations (U) and missed (M) regions obtained after applying the proposed multichannel watershed algorithm, Soille's watershed-based clustering, and ISODATA to the AVIRIS Salinas scene in Fig. 6(a) using different tolerance (T) values.

Method		$T = 80\%$				$T = 90\%$				$T = 95\%$			
		C	O	U	M	C	O	U	M	C	O	U	M
Multichannel watershed Algorithm	$B_3^{(\text{disk})}$	2	2	5	6	2	2	2	9	1	3	2	9
	$B_7^{(\text{disk})}$	6	2	3	5	5	2	3	6	5	2	2	7
	$B_{11}^{(\text{disk})}$	10	0	0	6	10	0	1	5	9	0	1	6
	$B_{15}^{(\text{disk})}$	11	1	0	0	11	1	0	0	11	1	0	0
Watershed-based clustering		7	1	2	5	6	1	1	6	5	1	1	8
ISODATA		3	2	2	8	2	1	2	10	2	0	1	12

result given the slight differences in the spectral characteristics of the four lettuce fields. Overall, the results shown in Table 3 reveal that the proposed algorithm can achieve very accurate segmentation results in a complex analysis scenario given by agricultural classes with very similar spectral features.

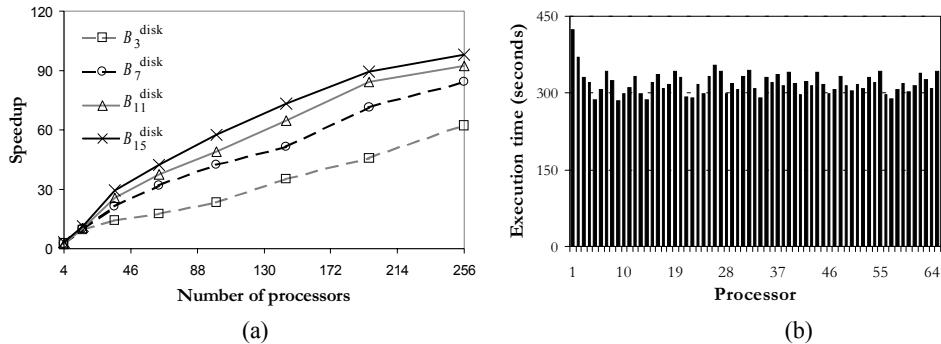
It should be noted that the proposed algorithm required several hours of computation in the same computing environment described in MRI experiments, which created the need for a parallel implementation. In order to investigate the efficiency of our parallel multi-channel watershed implementation, we coded the algorithm using the C++ programming language with calls to a message passing interface (MPI). The parallel code was tested on a high-performance Beowulf cluster (Brightwell et al., 2000) available at NASA's Goddard Space Flight Center in Maryland. The system used in experiments, called Thunderhead, is composed of 256 dual 2.4 Ghz Intel Xeon nodes with 1 Gigabyte of local memory and 80 Gigabytes of main memory.

Table 4 shows execution times in seconds of the parallel algorithm with the AVIRIS scene for several combinations of SE sizes and number of processors. Processing times were considerably reduced as the number of processors was increased. For instance, for the case of using $B_{15}^{(\text{disk})}$, which resulted in the most accurate segmentation results as shown in Table 4, 36 processors were required to produce a segmentation result in less than ten minutes. If the application under study can tolerate less accurate segmentations, such as those obtained using $B_3^{(\text{disk})}$, then the number of required processors to produce the output in about five minutes was only 16. In order to further analyze the scalability of the parallel code, Figure 7a plots the speed-up factors achieved by the parallel algorithm over a single-processor run of the algorithm as a function of the number of processors used in the parallel computation. The achieved speed-up factors were better

Table 4. Execution time in seconds, T(K), and speed-up, S_K, achieved by the proposed multi-channel watershed algorithm with the AVIRIS Salinas scene in Figure 6a for several combinations of structuring element sizes and number of processors (K).

K	$B_3^{(\text{disk})}$		$B_7^{(\text{disk})}$		$B_{11}^{(\text{disk})}$		$B_{15}^{(\text{disk})}$	
	T(K)	S _K	T(K)	S _K	T(K)	S _K	T(K)	S _K
1	3145	1.00	7898	1.00	11756	1.00	16234	1.00
4	1195	2.63	2668	2.96	5090	2.31	5026	3.23
16	329	9.56	772	10.23	1112	10.58	1445	11.23
36	220	14.34	372	21.22	459	25.56	551	29.45
64	180	17.43	245	32.11	313	37.45	424	39.34
100	134	23.45	186	42.45	238	49.21	282	57.45
144	89	35.10	154	51.25	182	64.56	221	73.28
196	68	45.67	110	71.23	139	84.32	181	89.34
256	49	62.45	93	84.39	127	92.34	165	98.23

Figure 7. (a) Speed-up factors achieved by the parallel algorithm as a function of the number of processors; (b) execution times in seconds achieved for each of the processors using $B_{15}^{(\text{disk})}$ as structuring element and 64 processors



for large SEs, a fact that reveals that the proposed parallel implementation is more effective as the volume of computations increases.

Finally, in order to investigate load balance, Figure 7b shows the execution times of the parallel algorithm for each of the processors on Thunderhead for a case study where 64 processors were used in the parallel computation; one processor (master or root processor) presents a slightly higher computational load as compared to the other processors. This comes as no surprise because the root processor is in charge of data partitioning, and also combines the partial results provided by every processor. It can be seen, however, that load balance is much better among the rest of the processors. Summarizing, we can conclude that the proposed multi-channel watershed segmentation algorithm, implemented on a commodity cluster of PCs, achieved good results in terms of segmentation accuracy, speed-up, scalability and load balance in the context of a high-dimensional image analysis application, dominated by large data volumes and complex patterns of communication and calculation.

CONCLUSIONS

This chapter has developed an approach to generalize the concepts of mathematical morphology to multi-channel image data. A new vector organization scheme was described, and fundamental morphological vector operations were defined by extension. Theoretical definitions of extended morphological operations were then used in the formal definition of a multi-channel watershed-based segmentation algorithm, which naturally combines the spatial and spectral/temporal information present in multi-channel images in simultaneous fashion. While such an integrated approach holds great promise in several applications, it also creates new processing challenges. For that purpose, this chapter also developed a parallel implementation which allows processing of large images quickly enough for practical use. A quantitative segmentation evaluation

comparison with regard to standard techniques, using both MRI brain images and remotely sensed hyperspectral data collected by the NASA's Jet Propulsion Laboratory AVIRIS imaging spectrometer, revealed that the proposed parallel algorithm can produce highly accurate segmentation results in reasonable computation times, even when the computational requirements introduced by multi-channel imagery are extremely large.

ACKNOWLEDGMENTS

The research in this chapter was supported by the European Commission through the project entitled "Performance analysis of endmember extraction and hyperspectral analysis algorithms" (contract no. HPRI-1999-00057). The authors would like to thank Professor Chein-I Chang for providing the MRI data and Dr. J. Anthony Gualtieri for providing the hyperspectral data. A. Plaza would also like to acknowledge support received from the Spanish Ministry of Education and Science (Fellowship PR2003-0360), which allowed him to conduct research as postdoctoral scientist at NASA's Goddard Space Flight Center and University of Maryland, Baltimore County.

REFERENCES

- Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 641-647.
- Beucher, S. (1994). Watershed, hierarchical segmentation and waterfall algorithm. In E. Dougherty (Ed.), *Mathematical morphology and its applications to image processing*. Boston: Kluwer.
- Brightwell, R., Fisk, L. A., Greenberg, D. S., Hudson, T., Levenhagen, M., Maccabe, et al. (2000). Massively parallel computing using commodity components. *Parallel Computing*, 26, 243-266.
- Chang, C.-I (2003). *Hyperspectral imaging: Techniques for spectral detection and classification*. New York: Kluwer.
- Fan, J., Yau, D. K. Y., Elmargamid, A. K., & Aref, W. G. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10, 1454-1466.
- Green, R. O., et al. (1998). Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, 65, 227-248.
- Haralick, R., & Shapiro, L. (1985). Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29, 100-132.
- Hoover, A., et al. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, 673-689.
- Le Moigne, J., & Tilton, J. C. (1995). Refining image segmentation by integration of edge and region data. *IEEE Transactions on Geoscience and Remote Sensing*, 33, 605-615.
- Malpica, N., Ortúñoz, J. E., & Santos, A. (2003). A multi-channel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters*, 24, 1545-1554.

- Mehnert, A., & Jackway, P. (1997). An improved seeded region growing algorithm. *Pattern Recognition Letters*, 18, 1065-1071.
- Moga, A. N., & Gabbouj, M. (1997). Parallel image component labeling with watershed transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 441-450.
- Moga, A. N., & Gabbouj, M. (1998). Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing*, 51, 27-45.
- Montoya, M. G., Gil, C., & García, I. (2003). The load unbalancing problem for region growing image segmentation algorithms. *Journal of Parallel and Distributed Computing*, 63, 387-395.
- Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2002). Spatial/spectral endmember extraction by multidimensional morphological operations. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9), 2025-2041.
- Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2004). A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles. *Pattern Recognition*, 37, 1097-1116.
- Rajapakse, J., Giedd, J., & Rapaport, J. (1997). Statistical approach to segmentation of single-channel cerebral MR images. *IEEE Transactions on Medical Imaging*, 16, 176-186.
- Richards, J., & Jia, X. (1999). *Remote sensing digital image analysis* (3rd ed). Berlin: Springer.
- Seinstra, F. J., Koelma, D., & Geusebroek, J. M. (2002). A software architecture for user transparent parallel image processing. *Parallel Computing*, 28, 967-993.
- Shafarenko, L., Petrou, M., & Kittler, J. Automatic watershed segmentation of randomly textured color images. *IEEE Transactions on Image Processing*, 6, 1530-1544.
- Soille, P. (1996). Morphological partitioning of multispectral images. *Journal of Electronic Imaging*, 5, 252-265.
- Soille, P. (2003). *Morphological image analysis, principles and applications* (2nd ed.). Berlin: Springer.
- Tilton, J. C. (1999). A recursive PVM implementation of an image segmentation algorithm with performance results comparing the HIVE and the Cray T3E. In *Proceedings of the 7th Symposium on the Frontiers of Massively Parallel Computation*, Annapolis, MD.
- Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2, 176-201.
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 583-598.
- Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335-1346.

Section V: Special Segmentation Applications

Chapter XIV

Fuzzy Clustering-Based Approaches in Automatic Lip Segmentation from Color Images

Shilin Wang, Shanghai Jiaotong University, China

Wing Hong Lau, City University of Hong Kong, China

Alan Wee-Chung Liew, Chinese University of Hong Kong, China

Shu Hung Leung, City University of Hong Kong, China

ABSTRACT

Recently, lip image analysis has received much attention because the visual information extracted has been shown to provide significant improvement for speech recognition and speaker authentication, especially in noisy environments. Lip image segmentation plays an important role in lip image analysis. This chapter will describe different lip image segmentation techniques, with emphasis on segmenting color lip images. In addition to providing a review of different approaches, we will describe in detail the state-of-the-art classification-based techniques recently proposed by our group for color lip segmentation: “Spatial fuzzy c-mean clustering” (SFCM) and “fuzzy c-means with shape function” (FCMS). These methods integrate the color information along with different kinds of spatial information into a fuzzy clustering structure and demonstrate superiority in segmenting color lip images with natural low contrast in comparison with many traditional image segmentation techniques.

INTRODUCTION

It is well known that the dynamics of the lip shape contain much information related to both the speaker and the utterance. The increasing interest in extracting such lip gestures from lip image sequence stimulates a variety of applications for which the extracted visual information can improve the performance of the respective systems. One major type of application is audio-visual speech recognition (Kaynak et al., 2004; Nakamura, 2002). The famous McGurk effect (McGurk & MacDonald, 1976) demonstrates the bimodal nature of speech understanding, and also shows a promising way to improve the performance of automatic speech recognition (ASR) systems by integrating visual information. Speaker identity authentication is another type of applications exploiting the visual information extracted from lip images. Recent research shows that lip dynamics is an important cue for personal identity verification (Kanak et al., 2003; Mok et al., 2004) and it can also enhance the performance of voice-based and face-based methods (Wark et al., 2000; Li et al., 2003; Yemez et al., 2003).

Facing the growing demands of extracting visual information from lip movements, various techniques have been proposed in the past decades. These techniques can be divided into two major categories: image-based, that is, model-less approaches (Bregler & Konig, 1994) and model-based approaches (Liew et al., 2002; Wang et al., 2004). Information retention is the key advantage of the image-based approaches, however, feature selection and extraction and linguistic information extraction are nontrivial tasks. They are also of high dimensionality in general. On the other hand, the lip features extracted by a model-based approach are usually related to the physical lip characteristics and of much lower dimensionality; they are also invariant to translation, rotation, scaling and illumination. Hence, the model-based approaches have been widely studied in recent years and various lip models have been proposed.

The most critical problem for model-based approaches is how to extract the lip region from the entire lip image, which is called the lip region segmentation problem. The accuracy and robustness of the lip segmentation process is of key importance for subsequent processing. Recently, segmenting color lip images has become more popular than segmenting gray-scale images due to the availability of low-cost hardware and increasing computing power. Color provides additional information that is not available in gray-scale images and thus the robustness of a lip segmentation algorithm can be enhanced. However, the large variations exhibited in different images caused by different speakers, utterances, lighting conditions or makeup create difficulties for this task. Low chromatic and luminance contrast between the lip and facial skin for an unadorned face is another major problem for lip region detection.

In this chapter, we describe how to automatically segment the lip region from color lip images. The chapter is organized as follows. We first provide a brief review of various kinds of lip segmentation techniques reported in recent literature. Major merits and drawbacks of these methods are also explicated. Then we describe in detail state-of-the-art fuzzy clustering-based techniques recently proposed by our group for color lip segmentation: The “spatial fuzzy c-mean clustering” (SFCM) method (Liew et al., 2003) and the “fuzzy c-means with shape function” (FCMS) method (Leung et al., 2004). From the experimental results, we show that our methods are able to provide accurate and robust lip region segmentation result even for lip images with undesirable color contrast.

BACKGROUND AND RELATED WORK

Non Clustering-Based Lip Image Segmentation Approaches

Lip segmentation is a process to identify the lip pixels from the entire lip image. Various techniques have been proposed to achieve this goal. These techniques can be categorized into three major approaches and their strengths and shortcomings are discussed in the following.

Color Space Analysis

The color space analysis approaches (Baig et al., 1999; Eveno et al., 2001) are usually applied to lip images where the lip and skin color difference is prominent. By transforming the RGB components onto the YUV color space, the corresponding V component image can be used for segmentation (Baig et al., 1999). The advantage of utilizing V component is that the difference between teeth, tongue and the dark portion of the interior mouth is very small, whereas the contrast between the lip and interior mouth is quite large. However, this approach will have problems for closed-mouth images since the difference in V component between the lip and skin pixels is not large enough for differentiation.

In order to enlarge the lip and skin color difference, the RGB color space can be transformed to the chromatic curve map (Eveno et al., 2001). The lip region is then obtained by properly thresholding the chromatic curve map in order to minimize the detection error. The reduced processing time is the key advantage of this kind of approach (Baig et al., 1999; Eveno et al., 2001), however they are sensitive to color contrast and noise. Nevertheless, methods solely depending on color space analysis will result in large segmentation error if the color distribution of the lip region overlaps with that of the background region.

Boundary Detection

The main idea of the boundary detection approaches is to segment the lip region by detecting the lip background boundary. In Caplier (2001), the lip contour was fitted toward the maximum values of the spatiotemporal gradients. This approach assumes that the lip region has darker pixels, which are associated with the spatial gradient, and the lip motion induces a temporal variation of luminance, which is associated with the temporal gradient. However, segmenting lip images using edge direction has been shown to be more reliable than using edge magnitude under natural lighting conditions (Gordan et al., 2001). Lip segmentation based on gradient information for both magnitude and direction of gray-level images has also been performed. The lip region was then manually separated into several non-overlapping subregions and the lip boundary pixels were detected in each subregion based on the magnitude and direction information in the edge map.

These boundary-detection based approaches have proven to be effective for speakers with lipstick or reflective markers. The assumptions of having consistent and prominent luminance changes along the lip boundary have always been inherently imposed in this approach, which may not be satisfied for lip images with poor contrast.

Markov Random Field (MRF)-Based Techniques

The Markov random field (MRF) technique has been widely used in image segmentation recently. It exploits local neighborhood information to enhance the robustness against “pepper” noise. Lievin and Luthon have proposed an MRF-based lip segmentation method for which the RGB image is first transformed to HI (hue and intensity) color space and the sequence dependent parameters are evaluated (Lievin & Luthon, 1998). Then an MRF-based approach is employed to segment the lip region using red hue predominant region and motion in a spatiotemporal neighborhood.

The spatial edge and hue color information have been combined in a MRF-based method (Zhang & Mersereau, 2000). A modified Canny edge detection algorithm, which accentuates the importance of the horizontal edges, has been applied to derive the hue edge map. The two streams of information (hue and edge) are then combined within the MRF framework to provide lip segmentation results. From the segmentation results shown (Lievin & Luthon, 1998; Zhang & Mersereau, 2000), patches outside and holes inside the lip region will generally be found. This phenomenon is caused by the aggregation of a large number of pixels with similar color, and exploiting local spatial information does not help improve this situation.

Fuzzy Clustering-Based Lip Image Segmentation Approaches

Recent research shows that fuzzy clustering techniques are powerful tools for image segmentation. The traditional fuzzy c-means (FCM) clustering algorithm was proposed in Bezdek (1981). FCM attempts to assign a probability value to each pixel in order to minimize the fuzzy entropy. Since it is an unsupervised learning method for which neither prior assumptions about the underlying feature distribution nor training is required, FCM is capable of handling lip and skin color variations caused by makeup. However, since the traditional FCM only consider the color information, the segmentation performance will be degraded when the color contrast is poor or noise is present. In order to handle these problems, incorporating the spatial information into the objective function has been proposed to improve the robustness, for example, Qian and Zhao (1997). The rationale behind their approaches is that pixels in the homogeneous region are likely belong to the same cluster while those along the edge are likely belong to a different one. Nevertheless, the performance of these approaches greatly depends on the accuracy of the edge detector. Pham introduces a “smooth” term in the objective function to penalize neighbouring pixels belonging to different clusters (Pham, 2002). However, the term also smooths the edge region and hence some important structure, that is, the boundaries of the object, in the image may be discarded.

Recently, we have proposed two novel fuzzy clustering-based segmentation techniques, namely, the “spatial fuzzy c-mean clustering” (SFCM) method (Liew et al., 2003) and the “fuzzy c-means with shape function” (FCMS) method (Leung et al., 2004). SFCM takes the color information of the neighbourhood pixels into account so that the segmentation error caused by noise and ambiguity in the image can be suppressed. In FCMS, the spatial distance of a pixel toward the lip center is considered and thus pixels with similar color but located in varying distances can be differentiated. The additional spatial information of SFCM and FCMS can be seamlessly incorporated into a metric

measure of an objective function as well as the updated formulae in the optimization process. Details of these two algorithms will be described in the following sections.

Fuzzy Clustering-Based Techniques in Lip Images Segmentation

Generally speaking, fuzzy clustering is a generalized version of the conventional hard (crisp) clustering method. It allows data to be classified into several clusters; to some extent, while in the hard clustering each datum can only be constrained to one cluster. The fuzzy c-means (FCM) clustering algorithm is the original form of almost all fuzzy clustering techniques (Bezdek, 1981) and it has been widely used in many applications involving image segmentation.

The Mathematical Framework of FCM

In image segmentation, FCM aims to assign a membership value for each pixel based on the distance measure to the cluster centroids in the feature space in order to minimize a fuzzy entropy measure. Let $\mathbf{X} = \{\mathbf{x}_{1,1}, \dots, \mathbf{x}_{r,s}, \dots, \mathbf{x}_{N,M}\}$ be the set of feature vectors associated with an image I of size $N \times M$, where $\mathbf{x}_{r,s} \in \mathbf{R}^q$ is a q -dimensional color vector at the pixel location (r, s) . Let $d_{i,r,s}$ denote the Euclidean distance between the feature vector $\mathbf{x}_{r,s}$ and the centroid \mathbf{v}_i of the i^{th} cluster, that is:

$$d_{i,r,s}^2 = \|\mathbf{x}_{r,s} - \mathbf{v}_i\|^2 \quad (1)$$

Then FCM is formulated as the minimization of an objective function J_{FCM} given below:

$$J_{FCM} = \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m d_{i,r,s}^2 \quad (2)$$

subject to:

$$\sum_{i=0}^{C-1} u_{i,r,s} = 1, \quad \forall (r, s) \in I \quad (3)$$

where the $N \times M \times C$ matrix $\mathbf{U} \in M_{fc}$ is a fuzzy c -partition of X , $\mathbf{V} = \{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{C-1}\} \in R^{cq}$ with $\mathbf{v}_i \in R^q$ representing the set of fuzzy cluster centroids, $m \in (1, \infty)$ defining the fuzziness of the clustering and $u_{i,r,s}$ denoting the membership value of the $(r, s)^{\text{th}}$ pixel in fuzzy cluster C_i . The optimum solution of the fuzzy clustering is to minimize J_{FCM} , that is:

$$\{\mathbf{U}^*, \mathbf{V}^*\} = \min_{(\mathbf{U}, \mathbf{V})} \{J_{FCM}(\mathbf{U}, \mathbf{V})\} \quad (4)$$

Picard iteration can be used to solve for the optimum point (U^*, V^*) , since the optimum solution is the stationary point of J_{FCM} . The derivation of the parameter updating formulae in each iteration is described in the following.

First, $J_{FCM}(U, V)$ is minimized with respect to U with all color centroids fixed. Let $\Lambda = \{\lambda_{r,s}\}$ be the set of multipliers, the Lagrangian $\Phi(U, \Lambda)$ can be obtained as follows:

$$\Phi(U, \Lambda) = \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m d_{i,r,s}^2 + \sum_{r=1}^N \sum_{s=1}^M \lambda_{r,s} \left(1 - \sum_{i=0}^{C-1} u_{i,r,s} \right) \quad (5)$$

Taking the partial derivative of $\Phi(U, \Lambda)$ with respect to $u_{i,r,s}$ and setting it to zero, we have the solution for non-singular case with $d_{i,r,s}^2 \neq 0$ given as:

$$u_{i,r,s} = \frac{(d_{i,r,s}^2)^{1/(m-1)}}{\sum_{j=0}^{C-1} (d_{j,r,s}^2)^{1/(m-1)}}; \quad (6)$$

and the solution for singular case with $d_{i,r,s}^2 = 0$ is $\begin{cases} u_{i,r,s} = 1 \\ u_{k,r,s} = 0 \quad \text{for } k \neq i \end{cases}$.

Then, J_{FCM} is minimized with respect to V with the membership distribution U fixed. Taking partial derivative of J_{FCM} with respect to v_i , we have:

$$v_i = \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \mathbf{x}_{r,s} / \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \quad (7)$$

The optimization procedure of FCM is performed by iterating the updating equations (6) and (7) until the solution is converged.

For the conventional FCM algorithm, the classification process depends solely on the feature space distribution, and each pixel is treated independently. However, for data with an inherent spatial ordering, such as image data, this kind of segmentation method will be greatly affected by noise and ambiguity. Hence, the correlation between neighborhood pixels should be exploited to overcome these difficulties. The property of spatial continuity is an important characteristic of images and the class information of all pixels is somehow associated with their neighbors. Considering this kind of information, along with the feature distribution, the performance of FCM in image segmentation can be further improved.

Spatial Fuzzy C-Means (SFCM) Clustering Algorithm

In this section, we will introduce our recently proposed spatial fuzzy c-means (SFCM) clustering algorithm for image segmentation. This algorithm makes use of the spatial continuity information to enhance the segmentation performance against image noise and ambiguity. The spatial information is seamlessly incorporated into the formulation of the fuzzy entropy rather than being used in the post-processing or in a heuristic way (Tolias & Panas, 1998).

Let's consider a 3×3 image window. If the 3×3 patch belongs to the same class, then the center pixel should be smoothed by its neighboring pixels so that eventually all pixels in the window have high and similar membership values in one of the clusters. Now, consider the feature vector $\mathbf{x}_{r,s}$ and its topological neighbor $\mathbf{x}_{r-1,s-1}$. Let ∂ be the l_2 distance between them, that is, $\partial_{\{(r,s),(r-1,s-1)\}} = \|\mathbf{x}_{r,s} - \mathbf{x}_{r-1,s-1}\|$. Let $d_{i,r,s}$ be the l_2 distance between $\mathbf{x}_{r,s}$ and the cluster centroid \mathbf{v}_i . If $\partial_{\{(r,s),(r-1,s-1)\}}$ is small, that is, similar in feature, we would like $d_{i,r,s}$ to be strongly influenced by $d_{i,r-1,s-1}$. Otherwise, $d_{i,r,s}$ should be largely independent of $d_{i,r-1,s-1}$. Taking the 8-neighborhoods into account, we define a dissimilarity measure $DS_{i,r,s}$ that measures the dissimilarity between $\mathbf{x}_{r,s}$ and \mathbf{v}_r :

$$DS_{i,r,s} = \frac{1}{8} \sum_{l_1=-l_2=1}^1 \sum_{l_2=-l_1=1}^1 [d_{i,r,s}^2 \lambda_{l_1,l_2}^{r,s} + d_{i,r+l_1,s+l_2}^2 (1 - \lambda_{l_1,l_2}^{r,s})] , \quad (l_1, l_2) \neq (0,0) \quad (8)$$

where $\lambda_{i,j}^{r,s} = \lambda(\partial_{\{(r,s),(r+i,s+j)\}})$ is the weighting factor controlling the degree of influence of the neighboring pixels $(r+i, s+j)$ on the center pixel (r,s) :

$$\lambda(\partial) = \frac{1}{1 + e^{-(\partial - \mu)/\sigma}} \quad (9)$$

and μ, σ specifies the displacement of λ from 0, and the steepness of the sigmoid curve, respectively.

Compared with the $d_{i,r,s}^2$ of FCM, the modified dissimilarity measure $DS_{i,r,s}$ in effect smooths the cluster assignment of the center pixel by the cluster assignment of the adjacent pixels. It should be noted that this is not a simple noise filtering process. When the centre pixel is along the edge boundary, its feature value will be very different from that of its neighbors, reflecting that they are unlikely to belong to the same class. In this case, $DS_{i,r,s} \approx d_{i,r,s}^2$, that is, the center pixel is not affected by its neighborhood. When the window is on a step boundary, the center pixel is only affected by the neighboring pixels in the same class, i.e., on the same step level. When the center pixel is on a smooth region and is affected by all its neighbors, the degree of influence of each neighbor on the center pixel is determined by the similarity between the neighbor's and center pixel's features. Hence, $DS_{i,r,s}$ enables local spatial interactions between neighboring pixels that is adaptive to image content. $DS_{i,r,s}$ can easily be modified to allow larger regions of influence by using a larger window. Weighting can also be applied to the neighboring pixels, such that distant pixels become less influential.

The introduction of $DS_{i,r,s}$ successfully incorporates the feature space information and local spatial interactions into the fuzzy clustering structure. The modified objective function of SFCM can then be written as:

$$J_{SFCM} = \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m DS_{i,r,s} \quad (10)$$

subject to:

$$\sum_{i=0}^{C-1} u_{i,r,s} = 1, \quad \forall (r,s) \in I \quad (11)$$

Note that the notations used in SFCM are consistent with that of FCM. The minimization of J_{SFCM} can also be solved by Picard iteration. The parameter updating equation for the membership distribution and cluster centroids can be formulated as:

$$\text{For } DS_{i,r,s} \neq 0, \quad u_{i,r,s} = \frac{(DS_{i,r,s})^{-1/(m-1)}}{\sum_{j=0}^{C-1} (DS_{j,r,s})^{-1/(m-1)}} \quad (12)$$

$$\text{and for the singular case } DS_{i,r,s} = 0, \quad \begin{cases} u_{i,r,s} = 1 \\ u_{k,r,s} = 0 \quad \text{for } k \neq i \end{cases}$$

$$\mathbf{v}_i = \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \hat{\mathbf{x}}_{r,s} \Bigg/ \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \quad (13)$$

$$\hat{\mathbf{x}}_{r,s} = \frac{1}{8} \sum_{l_1=-1}^1 \sum_{l_2=-1}^1 [\lambda_{l_1,l_2}^{r,s} \mathbf{x}_{r,s} + (1 - \lambda_{l_1,l_2}^{r,s}) \mathbf{x}_{r+l_1,s+l_2}], \quad (l_1, l_2) \neq (0,0) \quad (14)$$

where $\hat{\mathbf{x}}_{r,s}$ is the locally smoothed vector for pixel (r,s) . Iterating (12) to (14) form a Picard iteration and it will be terminated when the l_∞ difference between two consecutive iterations of the fuzzy c-partition matrix U falls below a small threshold (Liew et al., 2000).

The weighting functions $\lambda(\partial)$ in the dissimilarity measure (8) can be precomputed, as they remain unchanged in each iteration. The parameter μ in (9) can be viewed as the average “randomness” of the homogeneous region. It takes into account the presence of noise in the homogeneous region. Let us denote ∂_{av} as the average ∂ in a 3×3 window centered at pixel (r,s) :

$$\partial_{av}(r,s) = \frac{1}{8} \sum_{l_1=-1}^1 \sum_{l_2=-1}^1 \partial_{\{(r,s),(r+l_1,s+l_2)\}}, \quad (l_1, l_2) \neq (0,0) \quad (15)$$

Assuming that for a real image, most 3×3 windows fall on the homogeneous region, then μ can be set to be the average of $\partial_{av}(r,s)$ over all (r,s) , that is, $\mu = \sum_{r=1}^N \sum_{s=1}^M \partial_{av}(r,s)$. The steepness parameter σ in (9) controls the influence of neighboring pixels on the center pixel. When σ is of a large value, λ is less sensitive to the change of feature difference ∂ and a larger value of ∂ is required to suppress the influence of neighboring pixels, that is, larger ∂ is needed before λ reaches one. We estimate σ as follows:

1. Compute $\partial_{av}(\underline{x})$ for the entire image and set ∂_t to be the 95 percentile of $\partial_{av}(\underline{x})$.
2. Then, we let $\lambda(\partial_t)=0.8$ and solve for σ using (9).

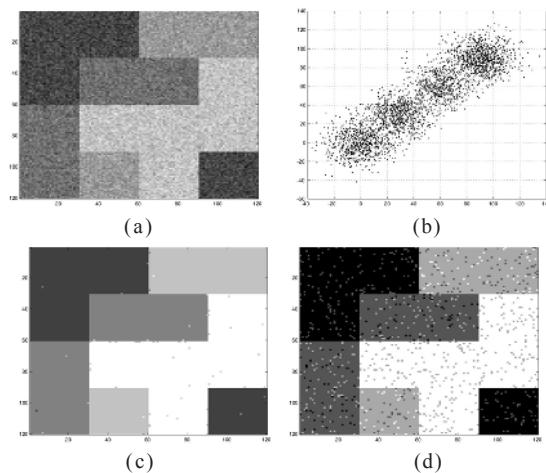
Figure 1(a) shows a 120×120 synthetic image with grey levels of 0, 30, 60 and 90. The image was then contaminated by additive zero mean Gaussian noise with SNR=13.6dB. Two such images are then used to form a 2-D test image, that is, each pixel in the image is described by a two-element vector. The correct solution of the clustering process is a clean four level image with centroids (0,0), (30,30), (60,60) and (90,90).

Figure 1(b) shows a scatterplot of the test image with every five pixels plotted and four classes can clearly be seen. Figure 1(c) shows the clustering result after hard-thresholding using the SFCM algorithm. The algorithm took ten iterations to terminate with four clusters and only 58 pixels, that is, 0.4% of the entire image, are misclassified. The cluster centroids obtained are (-0.40, -0.53), (29.59, 29.40), (61.95, 61.65) and (90.42, 90.36) and are very close to the ground truth solution. Figure 1 (d) shows the clustering result after hard-thresholding using the conventional FCM algorithm. The algorithm took 22 iterations to terminate for four clusters. 934 pixels are misclassified, i.e., 6.49% of the image, and the final cluster centroids obtained are (-1.99, -2.13), (28.86, 28.60), (61.32, 61.21) and (92.21, 91.95). It can clearly be seen that the clustering result of our SFCM is superior to that of the conventional FCM algorithm.

Color Lip Image Segmentation Using SFCM

The lip image database we used for analysis is in 24-bit RGB format. As Euclidean distance is adopted in the fuzzy clustering algorithms, it is desirable to work in a uniform color space, where the distance between any two points in the color space is proportional

Figure 1. (a) Four-level test image corrupted by additive Gaussian noise; (b) scatter plot of (a); segmentation result by (c) SFCM, (d) conventional FCM



to the perceived color difference. CIELAB and CIELUV are reported to be approximately uniform and that fulfills the above requirement (CIE 1986). Details of the transformation from RGB to CIELAB and CIELUV are:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (16)$$

$$L^* = \begin{cases} 116(Y')^{1/3} - 16 & \text{if } Y' > 0.008856 \\ 903.3Y' & \text{otherwise} \end{cases} \quad (17)$$

$$a^* = 500(K_1^{1/3} - K_2^{1/3}) \quad (18)$$

$$b^* = 200(K_2^{1/3} - K_3^{1/3}) \quad (19)$$

where

$$K_i = \begin{cases} \Phi_i & \text{if } \Phi_i > 0.008856 \\ 7.787\Phi_i + 16/116 & \text{otherwise} \end{cases} \quad (20)$$

for $i=1,2,3$ and $\Phi_1=X'=X/X_0$, $\Phi_2=Y'=Y/Y_0$, $\Phi_3=Z'=Z/Z_0$. And

$$u^* = 13L^*(u' - u_0) \quad (21)$$

$$v^* = 13L^*(v' - v_0) \quad (22)$$

where $u'=u=4X/(X+15Y+3Z)$, $v'=1.5v$ with $v=6Y/(X+15Y+3Z)$.

The X_0, Y_0, Z_0, u_0 and v_0 are the values of X, Y, Z, u and v for the reference white, respectively. The reference white in the transformation is defined as $\{R=G=B=225\}$. The color vector $\{L^*, a^*, b^*, u^*, v^*\}$ is adopted to represent the color information. The main reason for using both $\{L^*, a^*, b^*\}$ and $\{L^*, u^*, v^*\}$ as the color feature is to increase the segmentation robustness for a wide range of lip colors and skin colors.

Before the clustering process takes place, two preprocessing steps, that is, equalization of gradual intensity variation and teeth masking, must be carried out. The intensity equalization is used to compensate for the undesirable effects caused by uneven illumination and the procedure is carried out as follows:

1. Calculate the intensity variation along the columns on the upper and lower borders of the image using a 5×3 window. The upper and lower borders' intensity variations along columns are denoted by $u(j)$ and $l(j)$, respectively, where j ranges from 1 to M .
2. Compute the mean value m_{lu} by averaging $u(j)$ and $l(j)$.

3. Calculate the equalized luminance for each pixel along the j^{th} column by:

$$L'(i, j) = L(i, j) + \frac{l(j) - u(j)}{N-1} (i-1) + m_{lu} - l(j), \quad (23)$$

where N is the number of rows in the image.

The luminance component of the color feature is replaced by its equalized value and the color feature vector used in the clustering becomes $\{L^*, a^*, b^*, u^*, v^*\}$.

The presence of teeth pixels in the lip image is of concern for the fuzzy clustering based algorithms. Their presence will disturb the membership distribution by biasing the cluster centroids. It has been observed that the teeth region has a lower a^* and u^* value than that of the rest of the lip image. By experimenting with different lip images, best segmentation results can generally be obtained by setting the teeth thresholds as follows:

$$t_a = \begin{cases} \mu_a - \sigma_a & \text{if } (\mu_a - \sigma_a) < 9 \\ 9 & \text{otherwise} \end{cases} \quad (24)$$

$$t_u = \begin{cases} \mu_u - \sigma_u & \text{if } (\mu_u - \sigma_u) < 29 \\ 29 & \text{otherwise} \end{cases} \quad (25)$$

where μ_a, σ_a and μ_u, σ_u are the mean and standard deviation of a^* and u^* , respectively. Possible teeth pixels, that is, $a^* \leq t_a$ or $u^* \leq t_u$, are masked for subsequent clustering processes.

After preprocessing, the parameters $m=2$ and $c=2$ are used for the SFCM to segment the lip region. Mean feature vectors for the lip and skin classes obtained from hand labeled training images are used as initial cluster centroids. In order for the teeth region to be included in the lip region, the lip membership values of teeth region are set to 0.75. With the membership derived by SFCM, the following post-processing steps are adopted to enhance the segmentation result.

Morphological filtering: Grayscale morphological closing and opening with an 8-neighborhood structuring element is used to smooth the membership map and eliminate small erroneous blobs and holes.

Symmetry processing: Large spurious protrusions occasionally found around the upper or lower lip boundary cannot be eliminated by the morphological filtering operation. Taking advantage of the symmetry of the left and right side of the lip can eliminate these protrusions. The x -coordinate of the left (right) lip corner, $x_l(x_r)$, is found by scanning from the left (right) and detecting the first occurrence of a group of five pixels with membership value > 0.5 , arranged column-wise, and located approximately around the center row. Then the x -coordinate of the lip center, x_c , can be found. Next, the row-wise integral projection from the left lip corner to the lip centre given in (26) is computed for every row y .

$$\zeta_l(y) = \sum_{x=x_l}^{x_r} z(x, y) \quad (26)$$

where $z(x, y) = \begin{cases} 1 & , \text{ if } m(x, y) > 0.5 \\ 0 & , \text{ otherwise} \end{cases}$ and $m(x, y)$ denotes the lip membership. The row-wise integral projections of the right, $\zeta_r(y)$, are obtained similarly. By scanning downward from the top, the upper y -coordinate of the lip region is determined if the two following conditions are satisfied:

$$\zeta_l(y) > 0 \cap \zeta_r(y) > 0 \text{ and } (\zeta_l(y) < 3 * \zeta_r(y)) \cap (\zeta_r(y) < 3 * \zeta_l(y)) \quad (27)$$

The lower y -coordinate of the lip region is detected likewise. Lip pixels that are above the upper y -coordinate or below the lower y -coordinate are set to be non-lip.

Luminance processing: Since lip pixels have lower luminance value than skin pixels, the lip membership map can be enhanced by augmenting the membership value of pixels which are of low luminance value. We first estimate the luminance statistics of the skin pixels by computing the mean μ_{skin} and standard deviation σ_{skin} of the pixels in a strip of region around the image border. Next, for pixels having a luminance value $v < t_{skin}$, where $t_{skin} = \mu_{skin} - 3.5\sigma_{skin}$, the difference $d = t_{skin} - v$ is computed. For pixels with $v > t_{skin}$, d is set to zero. Then, set $d_{max} = \min(\sigma_{skin}, \max(d))$. Finally, for any pixel with a membership value $u > 0.45$ or $d > d_{max}$, u is augmented by a value equal to $d/(2d_{max})$ if $d \leq d_{max}$, or 0.5 if $d > d_{max}$. The modified membership value is then clipped at the maximum membership value in the unmodified membership map.

Shape Processing: Prior knowledge about the lip shape can be used to further reduce misclassification of pixels. A best-fit ellipse can be fitted onto the lip membership map to suppress any remaining spurious protrusions. The parameters of the best fit ellipse, that is, the center of mass, (x_m, y_m) , the inclination q about the center of mass, the semimajor axis x_d and the semiminor axis y_d , are computed from the lip membership $m(x, y)$ (Jain, 1989):

$$x_m = \frac{\sum_{x=1}^M \sum_{y=1}^N x * m(x, y)}{\sum_{x=1}^M \sum_{y=1}^N m(x, y)} \quad (28)$$

$$y_m = \frac{\sum_{x=1}^M \sum_{y=1}^N y * m(x, y)}{\sum_{x=1}^M \sum_{y=1}^N m(x, y)} \quad (29)$$

$$\theta = \frac{1}{2} \tan^{-1} \left\{ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right\} \quad (30)$$

$$x_a = \left(\frac{4}{\pi} \right)^{1/4} \left[\frac{(I_y)^3}{I_x} \right]^{1/8} \quad (31)$$

$$y_a = \left(\frac{4}{\pi} \right)^{1/4} \left[\frac{(I_x)^3}{I_y} \right]^{1/8} \quad (32)$$

with,

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - x_m)^p (y - y_m)^q m(x, y) \quad (33)$$

$$I_x = \sum_{x=1}^M \sum_{y=1}^N ((y - y_m) \cos \theta - (x - x_m) \sin \theta)^2 m(x, y) \quad (34)$$

$$I_y = \sum_{x=1}^M \sum_{y=1}^N ((y - y_m) \sin \theta + (x - x_m) \cos \theta)^2 m(x, y) \quad (35)$$

where the lip image is of size $N \times M$ and only potential lip pixels with membership value $m(x, y) \geq 0.5$ will be involved in the ellipse computation. After obtaining the best-fit ellipse, any potential lip pixels outside of the best-fit ellipse are flagged to be non-lip pixels and the lip membership map is smoothed by a Gaussian filter. Finally, a value of 0.45 is used for hard-thresholding the lip membership map and the single largest connected patch is considered as the lip region.

Figure 2 shows an example and the lip region can be clearly identified from the lip membership map produced by the SFCM algorithm, despite two spurious blobs above

Figure 2. (a) RGB lip image (shown in gray), (b) lip membership map, (c) after morphological filtering, (d) after symmetry, luminance and shape processing, (e) after Gaussian smoothing, (f) final segmentation

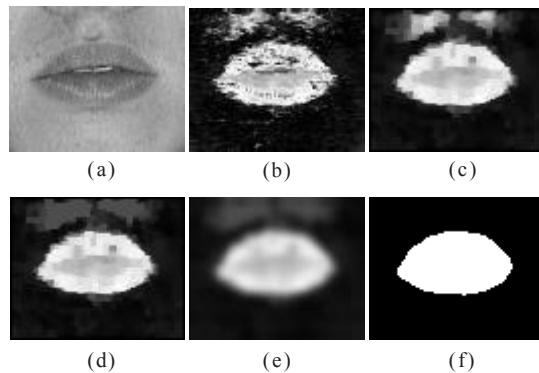
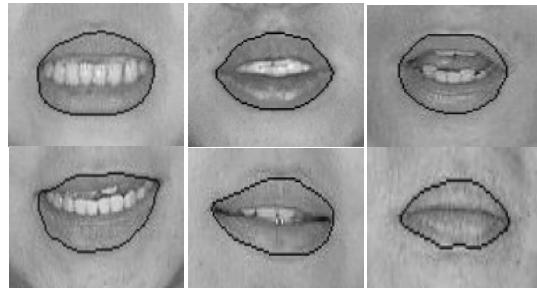


Figure 3. More segmentation results using SFCM

the lip region, as shown in Figure 2b. After morphological filtering, the membership map is smoothed and small holes are filled up as shown in Figure 2c. Figure 2d shows the membership map after symmetry, luminance and shape processing and it can be seen that the two spurious blobs have been eliminated. Figure 2e shows the membership map after Gaussian smoothing and Figure 2f shows the final segmentation result. More segmentation results using SFCM are shown in Figure 3.

In the next section, we will introduce another fuzzy clustering-based technique which is particularly designed for segmenting the lip region without the need of using these post-processing steps. A new kind of spatial information, which exploits high level knowledge of the lip shape, is incorporated for differentiating pixels with similar color but located in different regions.

Fuzzy C-Means with Shape Function (FCMS) Clustering Algorithm

The basic idea of using spatial information in FCMS is quite straightforward. A pixel whose location is far away from the lip center is unlikely to be a lip pixel even if its color component is similar to that of the lip pixels. For a pixel located close to the lip center, there is a high probability of being inside the lip region no matter what color it is. In other words, the distance between a particular pixel and the lip center can be used to provide important differential information. Since the shape of the outer lip contour resembles an ellipse, the spatial information about a pixel is naturally related to its elliptic distance to the lip center. However, one key issue in formulating the spatial information is that the lip center location and the approximated lip size are unknown. We will show in the following that these spatial parameters can also be updated and refined iteratively.

Let $\mathbf{p} = \{x_c, y_c, w, h, \theta\}$ denotes the set of parameters that describes the elliptic function, where (x_c, y_c) is the center of mass of the ellipse, w and h are respectively the semi-major axis and the semiminor axis and θ is the inclination angle about the ellipse center. The shape function is incorporated into the traditional FCM to form a new fuzzy measure. The joint dissimilarity measure $DF_{i,r,s}$ is defined as:

$$DF_{i,r,s} = d_{i,r,s}^2 + \alpha f(i, r, s, \mathbf{p}) \quad (36)$$

where

$$f(i, r, s, \mathbf{p}) = \left\{ \frac{((r - x_c) \cos \theta + (s - y_c) \sin \theta)^2}{w^2} + \frac{((s - y_c) \cos \theta - (r - x_c) \sin \theta)^2}{h^2} \right\}^{p_i} \quad (37)$$

The weighting parameter α in the dissimilarity measure defined in (36) is to adjust the weight of the physical distance against the color feature. The exponent p_i , defined in the elliptic function in (37), ensures a small function value for pixels within the i^{th} cluster and a large value for pixels outside the cluster. For lip segmentation problems, only two clusters are considered in the FCMS, one represents the lip region ($i=0$) and the other is the non-lip region ($i=1$). The elliptic function for each cluster has identical $\mathbf{p}=\{x_c, y_c, w, h, \theta\}$ but with different p_i .

The objective function of FCMS is given by:

$$J_{FCMS}(\mathbf{U}, \mathbf{V}, \mathbf{p}) = \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m DF_{i,r,s} \quad (38)$$

subject to

$$\sum_{i=0}^{C-1} u_{i,r,s} = 1, \quad \forall (r, s) \in I \quad (39)$$

Similar to FCM and SFCM, iterative method is adopted for deriving the optimal parameters of the membership distribution \mathbf{U} , color centroids \mathbf{V} and the spatial parameters \mathbf{p} . The derivation of the parameter updating formulae in each iteration remains unchanged: Let $\varphi: M_{fc} \rightarrow R$, $\varphi(\mathbf{U}) = J_{FCMS}(\mathbf{U}, \mathbf{V}, \mathbf{p})$ with $\mathbf{V} \in R^{cq}$ and $\mathbf{p} \in R^5$. Taking the partial derivative of $\varphi(\mathbf{U})$ with respect to \mathbf{U} and subject to the constraint (38), the updated membership value $u_{i,r,s}^+$ can be obtained by setting the derivative to zero and it is given by:

$$u_{i,r,s}^+ = \left[\sum_{j=0}^{C-1} \left(\frac{DF_{i,r,s}}{DF_{j,r,s}} \right)^{1/(m-1)} \right]^{-1} \quad (40)$$

Let $\psi: R^{cq} \rightarrow R$, $\psi(\mathbf{V}) = J_{FCMS}(\mathbf{U}, \mathbf{V}, \mathbf{p})$ with $\mathbf{U} \in M_{fc}$ and $\mathbf{p} \in R^5$ remains unchanged, and J_{FCMS} is expressed as:

$$\begin{aligned} J_{FCMS}(\mathbf{U}, \mathbf{V}, \mathbf{p}) &= \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m (d_{i,r,s}^2 + \alpha f(i, r, s, \mathbf{p})) \\ &= \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m d_{i,r,s}^2 + \alpha \sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} u_{i,r,s}^m f(i, r, s, \mathbf{p}) \\ &= J_{m1}(\mathbf{U}, \mathbf{V}) + \alpha J_{m2}(\mathbf{U}, \mathbf{p}) \end{aligned} \quad (41)$$

The partial derivative of $\psi(V)$ with respect to V is given by:

$$\frac{d\psi}{dV} = \frac{\partial J_{FCMS}}{\partial V} = \frac{\partial J_{m1}}{\partial V} + \alpha \frac{\partial J_{m2}}{\partial V} \quad (42)$$

Since J_{m2} is a constant when $U \in M_{fc}$ and $p \in R^5$ are fixed, the second term on the right hand side of (42) vanishes and the derivative $\frac{d\psi}{dV}$ is identical to that of the FCM. Following the derivation in Bezdek (1981), the updated centroid can be computed as follows:

$$\mathbf{v}_i^+ = \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \mathbf{x}_{r,s} \left/ \sum_{r=1}^N \sum_{s=1}^M u_{i,r,s}^m \right. \quad (43)$$

Finally, the partial derivative of $J_m(U, V, p)$ with respect to p is given by:

$$\frac{\partial J_{FCMS}(U, V, p)}{\partial p} = \frac{\partial J_{m1}}{\partial p} + \alpha \frac{\partial J_{m2}}{\partial p} \quad (44)$$

The first term on the right hand side of (44) vanishes, since J_{m1} is a function of the color features and is independent of the spatial parameter set p . The following equation is obtained by setting the partial derivative in (44) to zero:

$$\sum_{r=1}^N \sum_{s=1}^M \sum_{i=0}^{C-1} p_i u_{i,r,s}^m dist(r, s)^{p_i-1} \frac{\partial dist(r, s)}{\partial p} = 0 \quad (45)$$

where

$$dist(r, s) = \frac{((r - x_c) \cos \theta + (s - y_c) \sin \theta)^2}{w^2} + \frac{((s - y_c) \cos \theta - (r - x_c) \sin \theta)^2}{h^2} \quad (46)$$

$$\frac{\partial dist(r, s)}{\partial w} = -\frac{2t^2}{w^3} \quad , \quad \frac{\partial dist(r, s)}{\partial h} = -\frac{2g^2}{h^3} \quad (47)$$

$$\frac{\partial dist(r, s)}{\partial x_c} = -\frac{2t \cos \theta}{w^2} + \frac{2g \sin \theta}{h^2} \quad , \quad \frac{\partial dist(r, s)}{\partial y_c} = -\frac{2t \sin \theta}{w^2} - \frac{2g \cos \theta}{h^2} \quad (48)$$

$$\frac{\partial dist(r, s)}{\partial \theta} = -\frac{2t[(r - x_c) \sin \theta - (s - y_c) \cos \theta]}{w^2} - \frac{2g[(s - y_c) \sin \theta + (r - x_c) \cos \theta]}{h^2} \quad (49)$$

$$t = (r - x_c) \cos \theta + (s - y_c) \sin \theta \text{ and } g = (s - y_c) \cos \theta - (r - x_c) \sin \theta. \quad (50)$$

Since the formulation of $\frac{\partial \text{dist}(r,s)}{\partial p}$ is quite complicated, solving p^+ directly from (45) is rather complex. Instead, the Conjugate Gradient (CG) method (Flannery et al., 1988) is adopted to solve p^+ numerically for its fast convergence. Finally, the optimal solution (U^*, V^*, p^*) is derived by iterating equations (40) and (43) together with p^+ obtained via the CG method.

Color Lip Image Segmentation Using FCMS

Similar to SFCM, $\{L^*, a^*, b^*, u^*, v^*\}$ is used to represent the color feature of all pixels in the image and the two preprocessing steps, the intensity equalization and teeth masking, are also applied prior to clustering. After that, the FCMS algorithm is performed. A threshold value of 0.5 is used to hard-threshold the membership map in order to obtain the final segmentation. About 5,000 lip images collected from more than twenty speakers in our laboratory have been used to test the performance of the FCMS algorithm. In the experiments, the parameters of FCMS are set as follows: $C=2, m=2, p_0=5, p_1=-3$ and $\alpha=10$.

Convergence Time Analysis

It is observed from the experimental results that the initial values of w and h do not significantly affect the converged membership and segmentation results. Nevertheless, they will affect the convergence time of the algorithm. The effects of the initial values of w and h on the average convergence time for the 5,000 images have been investigated. The number of iterations required for the convergence is listed in Table 1. It is observed that the FCMS algorithm with starting conditions of $w=50$ and $h=30$ uses the least number of iterations to converge. The FCMS requires average of about 6.2 iterations to converge while the FCM algorithm needs 8.52 iterations. It is also worthwhile to note that the number of iterations required by the FCMS is less sensitive to the initial value of the color centroids as compared with the FCM.

Table 1. Average number of iterations with different initial w and h for FCMS and FCM to converge

Iterations	w=10	w=20	w=30	w=40	w=50	w=60	w=70
$h=10$	10.40	10.00	9.99	9.70	9.63	9.59	9.57
$h=20$	9.72	9.04	8.48	8.14	7.92	7.67	7.46
$h=30$	9.18	7.92	7.09	6.48	6.20	6.32	6.53
$h=40$	8.58	7.39	6.58	6.35	6.52	7.02	7.68
$h=50$	8.43	7.39	7.15	7.36	7.92	8.64	9.35
$h=60$	7.74	7.74	7.96	8.52	9.30	10.10	10.80
$h=70$	8.52	8.14	8.59	9.42	10.20	11.00	11.70
	FCM		8.52				

Computational Complexity Analysis

Incorporating the shape function into the fuzzy cluster analysis has inevitably increased the computational complexity. In order to compare the complexity involved in both FCMS and FCM, the number of additions, multiplications and table lookups required for the updating equations in each iteration has been counted. The lookup table is to provide the elliptic function value in (37) multiplied by α . With m equal to 2, the computational complexity per iteration for a lip image the size $N \times M$ is summarized in Table 2.

Compared with addition and table lookup, the multiplication operation predominates in the complexities of FCMS and FCM. The complexity of FCMS in terms of multiplications is $41 NM$ whereas the complexity of FCM is $23 NM$. Hence FCMS is about 1.78 times the complexity of FCM in each iteration.

It is shown that the FCMS and FCM require an average of 6.2 iterations and 8.52 iterations, respectively, to converge. The average computational times of the FCMS and FCM are thus in the order of $254.2 NM$ and $196 NM$, respectively. Hence, the average computational time of the FCMS is about 1.3 times that of the FCM.

Table 2. Computational complexity analysis for (a) FCMS and (b) FCM with $m=2$

FCMS	Number of additions	Number of multiplications	Number of table look-ups
Dissimilarity measure calculation	$19NM$	$18 NM$	$2 NM$
Membership calculation	$2 NM$	NM	
Color centroids calculation	$12 NM$	$12 NM$	
Parameters for best-fit ellipse calculation	$3 NM$	$10 NM$	
Overall	$36 NM$	$41 NM$	$2 NM$

(a)

FCM	Number of additions	Number of multiplications
Dissimilarity measure calculation	$10 NM$	$10 NM$
Membership calculation	$2 NM$	NM
Color centroids calculation	$12 NM$	$12 NM$
Overall	$24 NM$	$23 NM$

(b)

Segmentation Results

Three lip images with different lip shapes and oral openings are used to illustrate the segmentation performance of the FCMS. In particular, the third lip image has weak color contrast between the lip and skin. In order to illustrate the merits of the FCMS, its segmentation results are compared with those obtained by color transformation algorithm (CT) (Eveno et al., 2001), traditional FCM (Bezdek, 1981), Lievin and Luthon's method (LL) (Lievin et al., 1998) and Zhang and Mercereau's method (ZM) (Zhang et al., 2000). The membership distribution and segmentation results of the six methods for image 1, image 2 and image 3 are shown in Figure 4, Figure 5 and Figure 6, respectively. It should

Figure 4. (a) Lip image 1, (b)-(f): segmentation results by LL, ZM, CT, FCM and FCMS, (g)-(i): membership distribution of CT, FCM and FCMS

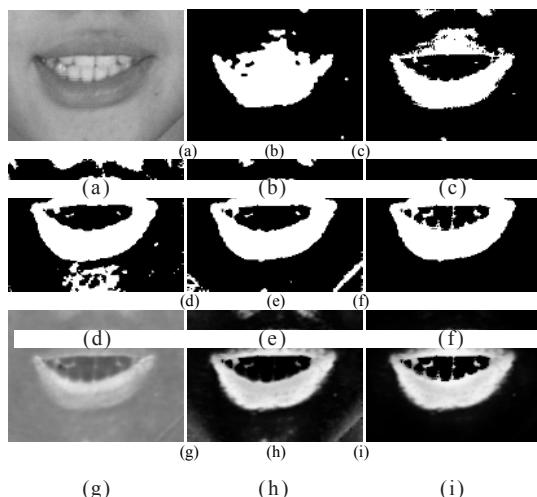


Figure 5. (a) Lip image 2, (b)-(f): segmentation results by LL, ZM, CT, FCM and FCMS, (g)-(i): membership distribution of CT, FCM and FCMS

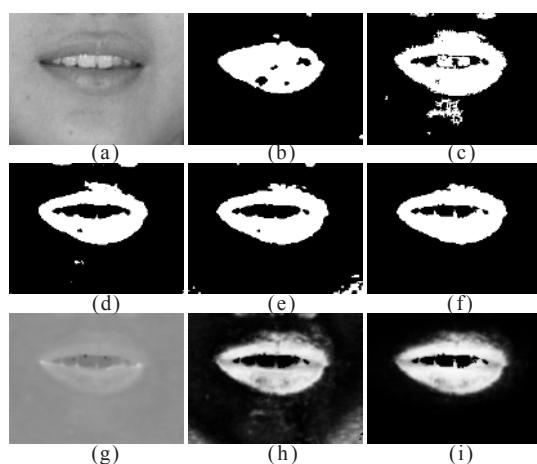
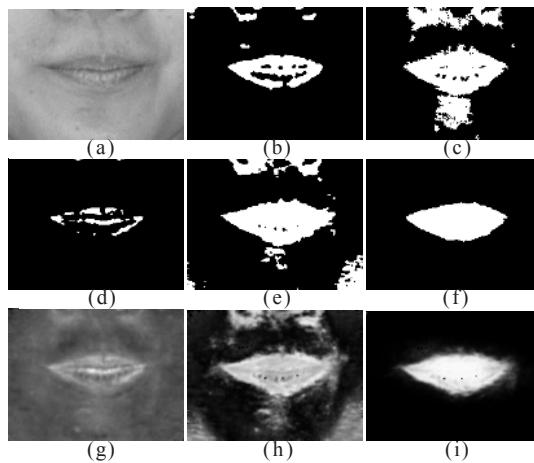


Figure 6. (a) Lip image 3, (b)-(f): segmentation results by LL, ZM, CT, FCM and FCMS, (g)-(i): membership distribution of CT, FCM and FCMS



be noted that the curve value $\kappa(x,y)$ of each pixel (x,y) is considered as equivalent to the membership value for the color transformation method.

It is clearly shown from these results that FCMS has strengthened the membership of both the lip and non-lip regions and thus yields a better boundary. For the lip images with reasonably good color contrast, CT, FCM and FCMS basically contain the membership reflecting the lip shape. On the other hand, for images having poor contrast against the background, such as image 3 in Figure 6, FCMS can produce much better membership in both the skin and lip regions than other methods.

The segmentation results from the CT, FCM, FCMS LL and ZM are respectively shown in Figures 6b-6f. These results also demonstrate the superior performance of the new FCMS among the five methods investigated. With the incorporation of the elliptic shape function, the background produced by FCMS is considerably cleaner in comparison with other methods and the boundary is also much smoother. Unlike other methods with patches scattered in the images and jagged boundary, FCMS generates a segmentation well matched to the original image.

To objectively quantify the performance of various techniques, a quantitative technique is applied to evaluate the quality of the segmentation results. The boundary of the lip region for the three lip images is drawn manually and a segmentation error (SE) measure (Lee et al., 1990) is defined as:

$$SE = P(O) \cdot P(B|O) + P(B) \cdot P(O|B) \quad (51)$$

where $P(B|O)$ is the probability of classifying an object as background and $P(O|B)$ is the probability of classifying background as object. $P(O)$ and $P(B)$ stand for the *a priori* probabilities of the object and the background of the image, respectively.

Table 3. Comparison of $P\{B|O\}$, $P\{O|B\}$ and SE among CT, FCM, FCMS, LL and ZM for the three lip images shown in Figure 4, Figure 5 and Figure 6

	SE(%)		
	Image 1 (Fig.4)	Image 2 (Fig.5)	Image 3 (Fig.6)
CT	9.56	2.53	8.18
FCM	5.13	4.15	8.90
FCMS	3.04	2.27	1.40
LL	7.60	4.94	11.36
ZM	4.58	3.61	4.56

Table 4. Comparison of average SE among CT, FCM, FCMS, LL and ZM for a further 27 lip images (not shown)

	CT	FCM	FCMS	LL	ZM
Average SE (%)	4.38	4.13	2.72	7.45	5.04

The SE for each of the five methods is given in Table 3. It is observed that the error percentages of FCMS for image 1 and image 2 are the smallest among all the methods investigated. In particular, the SE of FCMS for image 3, which has relatively poor color contrast, is substantially smaller than that of all other methods.

Another 27 lip images of different shapes from different speakers with fairly good color contrast have been selected from the database for detailed analysis. The average segmentation errors of the five methods for these 30 lip images are summarized in Table 4 and FCMS is again proven to have the smallest SE.

Analysis of Weighting Parameter and Exponents

The choice of the weighting parameter a and the exponents $\{p_0, p_1\}$ affects the balance between the spatial and color dissimilarity measure. In order to determine an appropriate setting, eight lip images with different lip shapes have been selected to evaluate the SE and convergence time (in terms of number of iterations) for different sets of $\{\alpha, p_0, p_1\}$ and the results are tabulated in Tables 5 and 6. It is observed from these results that the error rate is large when the values of $\{p_0, p_1\}$ are small because in this case the distance function is not effective to help differentiate the lip and skin regions. Using large p_i values can achieve smaller errors at the expense of taking longer time to converge. It is also observed that a very large value of a will cause the dissimilarity measure D to be too sensitive to the change in physical distance. The number of iterations is thus increased and a large segmentation error is also produced. However, if a is too small, there will be little difference between the FCMS and FCM since the dissimilarity measure D is mainly dependent upon the color features. Hence, the parameters $\{\alpha, p_0, p_1\}$ have been chosen as $\{10, 5, -3\}$ for fewer iterations required to converge and the reasonably small segmentation error produced.

Table 5. Average segmentation error (SE) of FCMS with different parameter sets $\{\alpha, p_0, p_1\}$

SE(%)	$p_0=1$ $p_1=-1$	$p_0=1$ $p_1=-2$	$p_0=2$ $p_1=-1$	$p_0=2$ $p_1=-2$	$p_0=3$ $p_1=-1$	$p_0=3$ $p_1=-3$	$p_0=5$ $p_1=-3$	$p_0=10$ $p_1=-7$
$\alpha=1$	7.01	6.95	6.80	6.71	6.26	6.88	4.04	3.55
$\alpha=2$	6.87	7.31	6.45	6.80	5.86	7.32	3.82	3.48
$\alpha=5$	6.54	8.09	6.09	7.63	5.26	7.43	3.64	3.38
$\alpha=10$	6.89	10.53	6.40	8.98	4.62	8.28	3.48	3.35
$\alpha=20$	11.31	19.40	8.94	18.24	4.74	10.99	3.29	3.63
$\alpha=50$	28.84	36.97	29.99	30.39	26.14	14.32	3.11	6.80
$\alpha=100$	53.46	57.32	49.77	47.09	40.95	15.57	3.29	15.84

Table 6. Convergence time (average number of iterations) of FCMS with different parameter sets $\{\alpha, p_0, p_1\}$

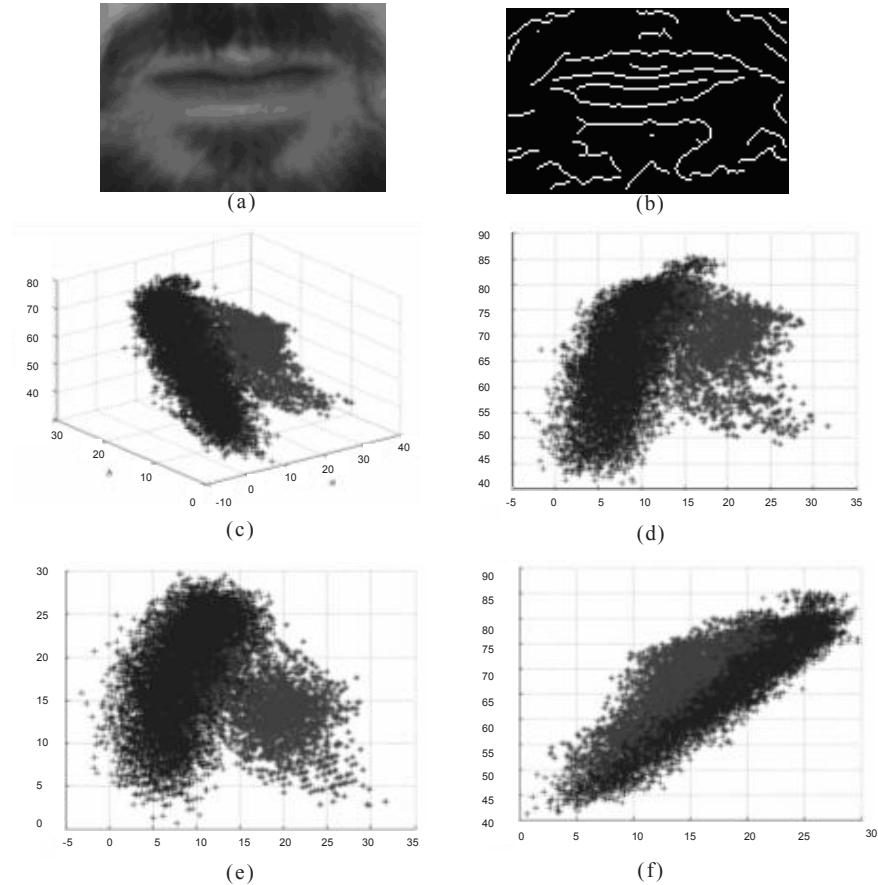
Convergence Time	$p_0=1$ $p_1=-1$	$p_0=1$ $p_1=-2$	$p_0=2$ $p_1=-1$	$p_0=2$ $p_1=-2$	$p_0=3$ $p_1=-1$	$p_0=3$ $p_1=-3$	$p_0=5$ $p_1=-3$	$p_0=10$ $p_1=-7$
$\alpha=1$	8.000	7.875	9.375	8.750	8.125	7.250	7.125	7.000
$\alpha=2$	6.875	6.625	8.250	7.625	8.000	7.000	7.125	8.000
$\alpha=5$	6.375	6.375	8.000	7.500	7.500	6.875	6.875	8.750
$\alpha=10$	6.250	7.000	7.250	7.375	6.875	7.375	7.125	10.500
$\alpha=20$	7.375	7.625	7.000	8.000	6.750	8.625	7.750	13.000
$\alpha=50$	9.000	9.875	7.750	9.250	9.875	10.375	9.625	18.000
$\alpha=100$	10.250	11.875	10.125	11.000	12.875	11.500	11.750	22.250

FUTURE WORKS

From the analysis and experimental results described above, our newly-developed lip segmentation algorithms, SFCM and FCMS, can well handle the segmentation problems caused by poor color contract. However, when the background region becomes complex, i.e., it does not only contain the skin area but various kinds of beards or mustaches as well, all the current techniques (including SFCM and FCMS) will fail. Figure 7 shows an example of this type of lip image and its corresponding color distribution of the lip and non-lip pixels in CIE-1976 CIELAB color space, where * and + represent the lip and background (or non-lip) pixels, respectively. The luminance edge map is also shown in Figure 7.

Some major observations can be drawn from these figures. First, the color distribution of the lip and background pixels overlaps in the color space and this will cause unsolvable problems for those segmentation methods solely based on color information. Moreover, the mislabeled pixels due to the color similarity will lead to undesirable disturbances to both the membership map and color centroids for the FCM based algorithms. Secondly, the presence of beards causes many luminance edges in the

Figure 7. (a) An example of lip image with complex background, (b) the luminance edge map of (a), (c) color distribution of (a) in CIELAB color space, (d)-(f): color distribution projection on the L-a, b-a, L-b plane, respectively (represents the lip pixel and + represents the background pixels).*



background region and using edge information to infer the lip boundary becomes extremely difficult and unreliable. As a result, methods that solely or partially depend on the edge information will not deliver satisfactory performance. Finally, the presence of beards causes a rather complex and inhomogeneous background, the traditional two-class partitioning algorithms are utterly inappropriate for these kinds of lip images. Due to these three major difficulties, all the lip segmentation algorithms, including the SFCM and FCMS reported in the literatures, cannot provide satisfactory segmentation results. One possible solution to solve these difficulties is to extend the FCMS algorithm to handle more than one background cluster. The color similarity and overlap problem can be well handled by incorporating the “global” spatial information in the FCMS and the

Figure 8. Examples of segmentation results obtained by the OOMB for lip images with different degrees of thickness and shape of beards and mustaches. The white region represents the segmented lip region.



complex background will then be modeled more adequately by increasing the number of background clusters. By integrating this idea into the FCMS algorithm to form a new framework, we have recently developed a new “one-object, multiple-background” (OOMB) clustering method to deal with this problem (Wang et al., 2004). Figure 8 shows examples of segmentation results obtained by the OOMB approach. It can be seen that the lip region has been accurately segmented for the first two images, but not for the third. It is generally observed that the dark regions in the lip corner always present difficulties to the OOMB algorithm and further work is still required to improve its performance.

CONCLUSIONS

Segmenting the lip region from color lip images is a key issue for lip image analysis and processing. Many techniques have been proposed in recent literatures to solve this segmentation problem, however, the problems of poor color contrast and noise hinder the performance of these methods. In this chapter, we have presented two fuzzy clustering based algorithms, namely spatial fuzzy c-mean (SFCM) clustering and fuzzy c-means with shape function (FCMS) clustering, to deal with the problems of image ambiguity and low color contrast. In our SFCM approach, local spatial context from neighborhood pixels is considered and thus the segmentation results will be more robust against noise and classification ambiguity. In our FCMS approach, an elliptic distance towards the lip center is considered, which helps differentiate pixels of similar color but located in different regions. Both approaches use fuzzy clustering structure to combine the spatial information with color information. The differences between the two approaches lie in: (1) the spatial information used in SFCM is in a “local” sense while that of FCMS is in a “global” sense; and (2) FCMS performs better in lip image segmentation since high level knowledge about the lip shape is used, while SFCM is more general since no specific structure about the object being segmented is assumed. Our approaches, as compared with other existing lip segmentation techniques, consistently produce superior results in the experiments. However, for lip images with beards/mustaches, these two approaches are unable to deliver satisfactory results. Further work is still required to solve this problem and the OOMB approach is possibly a direction to follow.

REFERENCES

- Baig, A. R., Seguier, R., & Vaucher, G. (1999). Image sequence analysis using a spatio-temporal coding for automatic lipreading. In *Proceedings of the International Conference on Image Analysis and Processing*, Venice, Italy (pp. 544-549).
- Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press.
- Bregler, C., & Konig, Y. (1994). Eigenlips for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, & Signal Processing*, Adelaide (Vol. 2, pp. 669-672).
- Caplier, A. (2001). Lip detection and tracking. In *Proceedings of the 11th International Conference on Image Analysis and Processing*, Palermo, Italy (pp. 8-13).
- CIE (1986). *Colorimetry*, 15(2). Bureau Central de la CIE, Vienna, Austria: CIE Publications.
- Eveno, N., Caplier, A., & Coulon, P. Y. (2001). New color transformation for lips segmentation. In *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France (pp. 3-8).
- Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1988). *Numerical recipes in C: The art of scientific computing*. New York: Cambridge University Press.
- Gordan, M., Kotropoulos, C., & Pitas, I. (2001). Pseudoautomatic lip contour detection based on edge direction patterns. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, Pula, Croatia (pp. 138-143).
- Jain, A. K. (1989). *Fundamentals of digital image processing*. NJ: Prentice-Hall International, Inc.
- Kanak, A., Erzin, E., Yemez, Y., & Tekalp, A. M. (2003). Joint audio-video processing for biometric speaker identification. In *Proceedings of IEEE International Conference on Acoustics, Speech, & Signal Processing*, Hong Kong, China (Vol. 2, pp. 377-380).
- Kaynak, M. N., Zhi, Q., Cheok, A. D., Sengupta, K., Jian, Z., & Chung, K. C. (2004). Analysis of lip geometric features for audio-visual speech recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 34(4), 564-570.
- Lee, S. U., Chung, S. Y., & Park, R. H. (1990). A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics and Image Processing*, 52, 171-190.
- Leung, S. H., Wang, S. L., & Lau, W. H. (2004). Lip image segmentation using fuzzy clustering incorporating an elliptic shape function. *IEEE Transactions on Image Processing*, 13, 51-62.
- Li, Y., Narayanan, S., & Kuo, C-C. J. (2003). Audiovisual-based adaptive speaker identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, & Signal Processing* (Vol. 5, pp. 812-815). Hong Kong, China.
- Lievin, M., & Luthon, F. (1998). Lip features automatic extraction. In *Proceedings of the IEEE International Conference on Image Processing*, Chicago (Vol. 3, pp. 168-172).
- Liew, A. W. C., Leung, S. H., & Lau, W. H. (2000). Fuzzy image clustering incorporating spatial continuity. *IEEE Proceedings of Vision, Image and Signal Processing*, 147, 185-192.

- Liew, A. W. C., Leung, S. H., & Lau, W. H. (2002). Lip contour extraction from color images using a deformable model. *Pattern Recognition*, 35, 2949-2962.
- Liew, A. W. C., Leung, S. H., & Lau, W. H. (2003). Segmentation of color lip images by spatial fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 11, 542-549.
- McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264, 746-748.
- Mok, L. L., Lau, W. H., Leung, S. H., Wang, S. L., & Yan, H. (2004). Lip features selection with application to person authentication. In *Proceedings of the IEEE International Conference on Acoustics, Speech, & Signal Processing*, Montreal, Canada (Vol. 3, pp. 397-400).
- Nakamura, S. (2002). Statistical multimodal integration for audio-visual speech processing. *IEEE Transactions on Neural Networks*, 3(4), 854-866.
- Pham, D. L. (2002). Fuzzy clustering with spatial constraints. In *Proceedings of IEEE International Conference on Image Processing*, New York (Vol. 2, pp. 65-68).
- Qian, Y., & Zhao, R. (1997). Image segmentation based on combination of the global and local information. In *Proceedings of the IEEE International Conference on Image Processing*, Santa Barbara, CA (Vol. 1, pp. 204-207).
- Tolias, Y. A., & Panas, S. M. (1998). Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions. *IEEE Transactions on System, Man and Cybernetics, Part A*, 28, 359-369.
- Wang, S. L., Lau, W. H., & Leung, S. H. (2004). Automatic lip contour extraction from color images. *Pattern Recognition*, 37, 2375-2387.
- Wang, S. L., Lau, W. H., Leung, S. H., & Liew, A. W. C. (2004). Lip segmentation with the presence of beards. In *Proceedings of the IEEE International Conference on Acoustics, Speech, & Signal Processing*, Montreal, Canada (Vol. 3, pp. 529-532).
- Wark, T., Sridharan, S., & Chandran, V. (2000). The use of temporal speech and lip information for multi-modal speaker identification via multi-stream HMMs. In *Proceedings of International Conference on Acoustics, Speech, & Signal Processing*, Istanbul, Turkey (Vol. 6, pp. 2389 - 2392).
- Yemez, Y., Kanak, A., Erzin, E., & Tekalp, A. M. (2003). Multimodal speaker identification with audio-video processing. In *Proceedings of the IEEE International Conference on Image Processing*, Istanbul, Turkey (Vol. 3, pp. 5-8).
- Zhang, X., & Mersereau, R. M. (2000). Lip feature extraction towards an automatic speechreading system. In *Proceedings of IEEE International Conference on Image Processing*, Vancouver, Canada (Vol. 3, pp. 226-229).

Chapter XV

Mathematical Morphology-Based Automatic Restoration and Segmentation for Degraded Machine-Printed Character Images

Shigueo Nomura, Kyoto University, Japan

Keiji Yamanaka, Federal University of Uberlândia, Brazil

Osamu Katai, Kyoto University, Japan

Hiroshi Kawakami, Kyoto University, Japan

Takayuki Shiose, Kyoto University, Japan

ABSTRACT

This chapter presents a morphological approach (AutoS) for automatic segmentation with feature vector extraction of seriously degraded machine-printed character images. This approach consists of four modules. The first detects and segments natural pitch characters based on the vertical projection of their binary images. The second uses an algorithm based on the vertical projection and statistical analysis of coordinates to detect fragments in broken characters and merge them before the eventual segmentation.

The third employs a morphological thickening algorithm on the binary image to locate the separating boundaries of overlapping characters. Finally, the fourth executes a morphological thinning algorithm and a segmentation cost calculation to determine the most appropriate coordinate at the image for dividing touching characters. By the automatic running of a suitable segmentation module for each problem, the AutoS approach has been robust, flexible and effective in obtaining useful and accurate feature vectors concerning degraded machine-printed character images.

INTRODUCTION

Mainly in the case of degraded images, character image segmentation is fundamental to such user interfaces as automatic reading systems based on optical character recognition (OCR), which perform on individual characters. Segmentation consists of an essential step for avoiding incorrect character recognition due to inappropriate input data, such as degraded images. Though problems like broken and touching characters are responsible for the majority of errors in automatic reading of machine-printed text (Casey & Nagy, 1982), segmentation of degraded character images is all too often ignored in the research community. The processing required for information capturing and extraction is still in its infancy (Marinai, Gori, & Soda, 2005).

In particular, the automation of character image segmentation and feature vector extraction is an essential stage for processing degraded images acquired from real-time imaging systems and used in our experiments.

Since most OCR systems already work with binary images, our proposed approach falls within the scope of these images.

RELATED WORKS AND TOPIC RELEVANCE

Essentially, existing techniques (Jung, Shin, & Srihari, 1999; Lu, 1995; Tan, Huang, Yu, & Xu, 2002; Taxt, Flynn, & Jain, 1989) are based on heuristics, whereby the text contains horizontal lines and the characters are proportionally sized and uniformly well-separated. In the case of degraded characters, recognition-based segmentation algorithms (Arica & Yarman-Vural, 1998; Lee & Kim, 1999; Nomura, Michishita, Uchida, & Suzuki, 2003), which require a recognizer to validate the segmentation process, are used. Under this concept, if the character is recognized then the segmentation is accepted, otherwise the segmentation is re-applied. These recognition-based segmentation algorithms are not only time consuming, but their outputs are heavily dependent on the character recognition process. In other words, existing techniques have not yet been able to simultaneously and appropriately treat such fragmenting, touching and overlapping character problems. Furthermore, they are not able to accurately extract feature vectors from real-world degraded images. On the other hand, automatic reading systems such as Anagnostis (Bourbakis, 1998) scan the text pages in order to extract characters in the form of binary strings.

Thus, approaches for providing those image restoration and segmentation abilities to the automatic reading systems still consist of a relevant topic in the field of segment processing.

APPROACH'S CONTRIBUTION AND EVIDENCE OF IT

Mathematical morphology operations in conjunction with heuristics that determine the potential segmentation points constitute the main idea for the adaptive approach's contribution in this work. Experimental degraded images were acquired from real scenes of moving vehicles taken by cameras installed on the roads in Uberlândia city, Brazil (Nomura, Yamanaka, & Katai, 2002). These images were captured under dubious light and focusing conditions, and their corresponding binary images include serious problems such as broken, overlapping or touching characters. Moreover, the degraded images present size and inclination problems in the characters. Thus, an adaptive approach is required to automatically solve all the problems mentioned above in real time.

Morphological segmentation techniques are considered promising because they rely on powerful mathematical morphology operations that are very attractive for accurately dealing with object-oriented criteria such as size and shape.

Also, character position identification and preservation (in degraded binary images) during segmentation followed by the feature vector extraction process constitute another contribution of the proposed approach. To date, the simple segmentation of characters has been insufficient for recognizing and retrieving codes (Nomura, 2002); that is, the position of each character (feature vector) in the code has become too relevant for dealing with machine-printed document images of a fixed number of characters. Position detection makes up for some unrecognizable character problems caused by poor input image quality.

MATHEMATICAL MORPHOLOGY- BASED THEORETICAL FOUNDATION

In this work, we apply a morphological approach to support the character (shape) image segmentation and feature vector extraction of degraded machine-printed character images. Morphological operators have been widely used in image analysis and processing (Serra, 1982; Serra, 1988).

The language of mathematical morphology (MM) is a set theory that represents objects in an image (Gonzalez & Woods, 1993; Serra, 1982). It is known that MM is a versatile and powerful image analysis technique for extracting useful image components with respect to representation and description of regions, such as boundaries, size, area, shape or connectivity in segmentation-oriented image processing (Gao, Siu, & Hou, 2001).

Furthermore, MM is preferable to other techniques, such as artificial neural networks, when the task is to provide accurate segmentation in real time for automatic reading systems. The reasons are: (1) it does not require complex and heavy mathematical calculations, (2) it does not need the training stage or training parameters, and (3) it provides more reliable results.

Normally, most existing morphological approaches employ a top-down technique (Park & Ra, 1999; Salembier & Pardas, 1994). In this case, if two shapes have ambiguous boundaries due to low contrast, they may be considered to belong to the same shape in

the sense of a homogeneity or similarity measure. Thus, sometimes the top-down technique tends to imprecisely segment the shapes. This tendency may cause a serious problem in perceptual applications such as realistic automatic recognition of degraded images, where semantic object shapes should be defined accurately.

In this work, morphological image processing is used to achieve the following tasks:

- Thickening operation for detection of separating boundaries for overlapping characters.
- Pruning operation for cleaning up parasitic branches in the thickened image.
- Thinning operation for constructing character skeletons of touching characters.
- Detection of crossing points at the skeletons of characters using structuring elements.

Then, MM operators (Gonzalez & Woods, 1993) for dilation, erosion and “hit-or-miss” transform of objects are used for implementing thickening, thinning and pruning algorithms. Also, crossing-point detection supported by segmentation cost calculation, which is similar to splitting cost in handwritten words (Kimura, Tsuruoka, Miyake, & Shridhar, 1994), has been developed to increase the accuracy of the segmentation process.

Thickening Algorithm

A morphological thickening operation consists in adding background pixels with a specific configuration to the set of object pixels (Soille, 1999). In this work, the thickening operation-based algorithm is employed to determine boundaries between overlapping characters in binary versions of degraded machine-printed character images.

The thickening (Soille, 1999) of a binary image X by a structuring element B is denoted by $X \otimes B$ and defined as the union of X and the hit-or-miss transform of X by B that is denoted by $X * B$: $X \otimes B = X \cup (X * B)$.

Thinning Algorithm

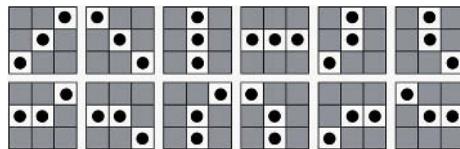
The thinning algorithm is the morphological dual of the thickening algorithm. Thus, a morphological thinning operation consists in removing the object pixels having a given configuration (Soille, 1999). A suitable thinning algorithm should preserve the connectivity and remove pixels from a object-character in a binary image until the width of the skeleton to unity (Elnagar & Alhajj, 2003; Jang & Chin, 1992). The thinning algorithm is considered sufficiently convenient and it is used to reveal crossing points of touching characters in this work.

The thinning (Soille, 1999) of a binary image X by a structuring element B is denoted by $X \times B$ and defined as the set difference between X and the hit-or-miss transform of X by B : $X \times B = X - (X * B)$.

Pruning Algorithm

A morphological pruning operation-based algorithm is an essential complement to the thickening algorithm, and is used to clean up parasitic components from skeletons.

Figure 1. Structuring elements for detecting an interconnecting pixel (Nomura, Yamanaka, Katai, Kawakami, & Shiose, 2005) in the pruning operation; the component “•” represents the pixel of the 8-connected object



The solution is based on suppressing a parasitic branch by successively eliminating its endpoint (Gonzalez & Woods, 1993).

Particularly in this work, there are situations in which a branch with endpoint must not be suppressed. For this reason, a particular pruning algorithm is employed to obtain the necessary “image without spurs” as reference data for segmentation (Nomura, 2002).

First of all, the 12 structuring elements in Figure 1 have been adopted to detect an interconnecting pixel (Nomura, 2002). The interconnecting pixel must belong to the 8-connected object, that is, it must not be an endpoint.

Then, the following definition of a relevant pixel is used to eliminate a parasitic branch. A pixel is considered relevant if it is in accordance with one of the two following conditions, that is, it must not be removed.

1. The pixel touches the boundary of the image.
2. It is an interconnecting pixel.

All eight neighbors of each non-relevant pixel (non-zero) are analyzed, after which it is possible to maintain the relevant pixels so that only parasitic branches are eliminated.

PROPOSED AUTOMATIC SEGMENTATION APPROACH

Figure 2 shows the architecture of the proposed automatic segmentation (AutoS) approach. The AutoS approach is composed by the following two parts:

Part 1: Preprocessing

Construct Equidistant Reference (ER) Lines

In this part, equidistant reference (ER) lines are determined for supporting the posterior segmentation process. The ER lines in Figure 3 are vertical ones to delimit a zone that will contain one character. They are equidistant because the zones for all the characters have equal width as verified in Figure 3. In terms of quantity, for $n \in \mathbb{N}$ characters or zones, we will have $n+1$ ER lines.

Figure 2. Architecture of the automatic segmentation (AutoS) approach

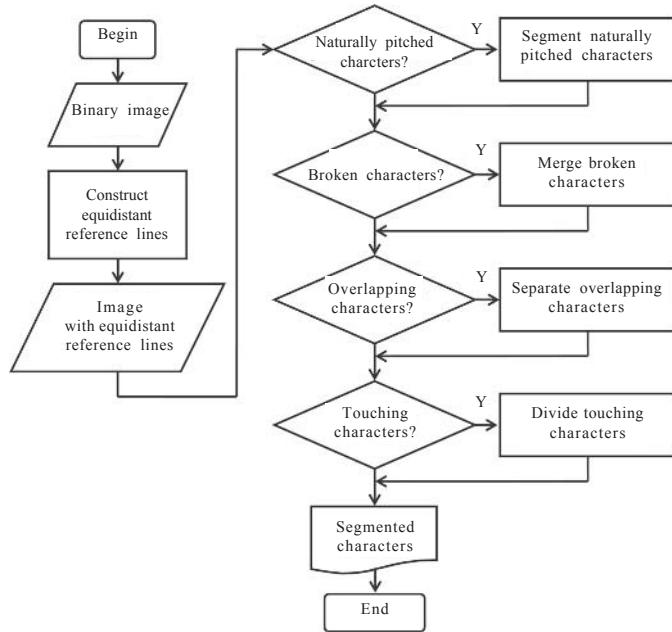
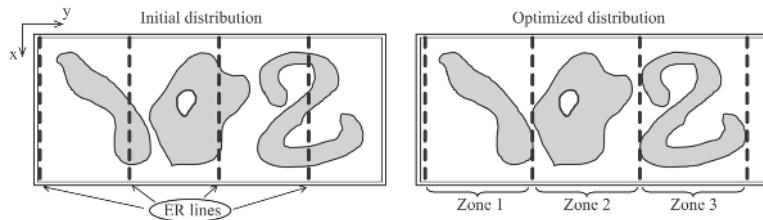


Figure 3. Initial distribution and optimized distribution of four ER lines for three zones (objects) according to the minimum segmentation cost criterion



The main condition for determining these ER lines is the best distribution of them along the image. The optimized distribution means providing the appropriate zone width to accommodate the characters (one in each zone), that is, the best positions for the ER lines. To determine this optimized distribution, we based it on the criterion of minimum segmentation cost for each distribution of ER lines.

Mathematical equations for determining the ER lines are the following:

$$w_z = \frac{wI}{nz} \quad (1)$$

where wz is the maximum width of a zone delimited by two consecutive ER lines; wI denotes the width of an image with all the zones to be delimited and nz represents the number of zones to be delimited.

$$mj = wI - nz * w \quad (2)$$

where: mj is the maximum index of column for 1st line belonging to a set of ($nz+1$) ER lines and w represents the width of a zone delimited by two consecutive ER lines.

$$sc_j = \sum_{i=1}^R f(i, j), \quad (3)$$

where: sc is the segmentation cost, that is, it gives the number of pixels when an eventual ER line hits an object (character) in the image; i denotes the index for representing x coordinates; j is the index for representing y coordinates of the image where the ER line is currently positioned; R represents the total number of rows in the image and $f(i, j)$ is the value of a pixel with i and j coordinates. Since the image is binary, the value is zero for background pixel, and one for character pixel.

Basically, we have two main loops in the algorithm for determining ER lines as shown in Figure 4. The outside loop is to increase the width of each zone and to save the corresponding width that satisfies the criterion of minimum cost segmentation. The inside loop is to shift the ER lines and to save the corresponding y coordinates of ER ones that satisfy the given cost criterion.

Figure 4. Algorithm for constructing ER lines

<p><i>Variables</i></p> <p><i>BI: the binary image with all the characters</i> <i>iw: the initial width of a zone delimited by two consecutive ER lines</i> <i>wz: the maximum width of the zone above</i> <i>w: the width of the zone above</i> <i>j: the position corresponding to 1st line in a set of ER lines</i> <i>mj: the maximum index of column for 1st line in a set of ER lines</i> <i>sc: the segmentation cost corresponding to an ER line</i></p> <p><i>Read BI</i></p> <p><i>Estimate iw</i></p> <p><i>Calculate wz by equation (1)</i></p> <p><i>For w = iw to wz</i></p> <p style="padding-left: 20px;"><i>Calculate mj by equation (2)</i></p> <p style="padding-left: 20px;"><i>For j = 1 to mj</i></p> <p style="padding-left: 40px;"><i>Get the columns for ER lines according to w</i></p> <p style="padding-left: 40px;"><i>Calculate sc by equation (3) for all the ER lines</i></p> <p style="padding-left: 40px;"><i>Save sc, j, w</i></p> <p><i>Get j, w for minimum sc</i></p>

Part 2: Segmentation Process

This part contains four modules for processing natural pitch, broken, overlapping and touching characters. Figure 5 shows the convention used in this work. We assume that a binary image has white objects in black background, and it has its rows represented by x coordinates and columns represented by y coordinates.

Segment Naturally Pitched (NP) Characters

We call a character on NP character when its space occupied in the image is clearly determined by segmentation points at a vertical projection. The vertical projection in Figure 5 is a histogram determined by counting the number of character pixels in each vertical scan at a particular column of the image (Lu, 1995). Thus, a peak in the histogram occurs for each vertical stroke of the character, and a zero value occurs between characters. In Figure 5, “1” and “2” indicate zero values between characters. This information from the histogram should be sufficient for segmenting well separated characters or NP ones.

Figure 5. Coordinate convention, a binary image and its corresponding vertical projection

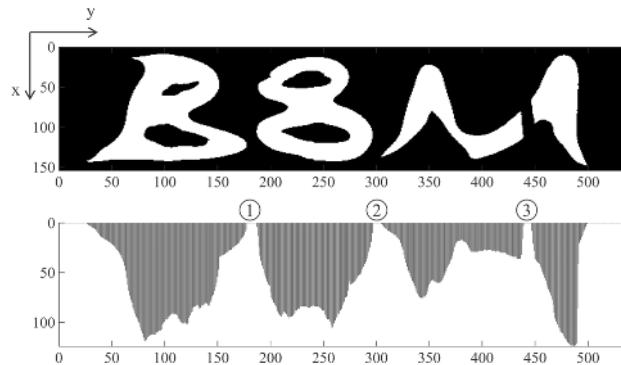
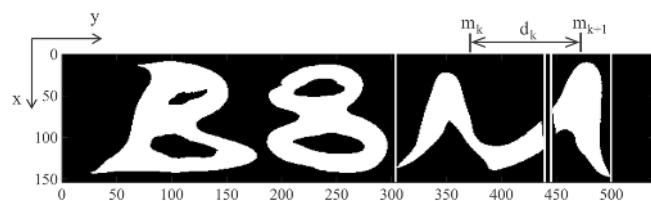


Figure 6. Example of the distance between two midpoints



Merge Broken Characters

Broken characters cause false zero values (they are not between characters) in the vertical projection. A false zero value is indicated by “3” in Figure 5 and it leads to incorrect segmentation of the character based on this vertical projection.

Lu, Haist, Harmon, and Trenkle (1992) proposed an algorithm for segmenting broken characters or fragments based on the estimated character width (Lu, Haist, Harmon, & Trenkle, 1992). In our case, however, this kind of estimation is not convenient because the width depends on the character type. For example, the digit “1”, or the letter “I” is narrower than other characters. Instead of estimating the fragment (segment) width, therefore, we calculated the distance between two midpoints of the fragments and its standard deviation of y coordinates belonging to the fragments. Since midpoints of machine-printed characters have uniform distribution, we can have the robustness of the proposed method for making a decision whether the analyzed segment is a fragment or not. Also, information about the quantity of segments in a word is necessary to make that decision about the analyzed segment. Based on these statistical values, the algorithm decides whether the fragments must be merged as presented in Figure 7.

Mathematical equations for merging the fragments are the following:

$$m_i = \frac{f_{i1} + f_{i2}}{2}, \quad (4)$$

where: m_i is the mean value or the midpoint of the i^{th} fragment; i denotes the index for indicating the i^{th} fragment in the image; f_{i1} represents the first y coordinate of the i^{th} fragment and f_{i2} is the last y coordinate of the i^{th} fragment.

$$d_k = m_{k+1} - m_k, \quad (5)$$

where: d_k is the k^{th} distance between two midpoints of consecutive fragments like in the image of Figure 6; k represents the index for indicating the k^{th} distance; m_k is the midpoint of the first fragment and m_{k+1} denotes the midpoint of the second fragment.

$$\mu_d = \frac{\sum_{k=1}^{F-1} d_k}{F-1}, \quad (6)$$

where: μ_d is the average distance of all distances and F represents the total number of fragments in the image.

$$\sigma_d = \sqrt{\frac{\sum_{k=1}^{F-1} (d_k - \mu_d)^2}{F-1}} \quad (7)$$

where: σ_d denotes the standard deviation between the k^{th} distance and the average of distances.

Figure 7. Algorithm for merging broken characters

```

Variables
    BI: the binary image with all the characters (including the broken ones)
    SI: the standard image without broken characters
    qs: the quantity of segments (characters) in SI
    F: the total number of segments including fragments of characters
    i: the index for fragments in the image
    mi: the midpoint of the ith fragment
    k: the index for distances
    dk: the kth distance between two midpoints
    μd: the average distance
    σk: the standard deviation
    ms: the maximum standard deviation among all the distances
    Vf: the vector with fragments concerning ms

Read BI, SI
Get F from BI
Get qs from SI
While F>qs
    For all the current fragments
        Compute mi by equation (4)
    For all the current midpoints
        Compute dk by equation (5)
    For all the current distances
        Compute μd by equation (6)
        Compute σk by equation (7)
    Get Vf corresponding to ms
    Merge the fragments in Vf
    Decrement F

```

Separate Overlapping Characters

In this case, the vertical projection is not able to provide a zero value between characters because of the overlapping problem. It is, therefore, not possible to find natural pitch characters using the vertical projection information. To solve this problem, our idea is to apply the morphological thickening operation to the binary image, and to determine the necessary boundaries to separate the overlapping characters. The closest ER line is used as reference to determine the final boundary for separating overlapping characters as presented in Figure 8.

Mathematical equations for separating the overlapping characters are the following:

$$mc = \frac{\sum_{i=1}^N c_i}{N}, \quad (8)$$

Figure 8. Algorithm for separating overlapping characters

```

Variables
BI: the binary image with all the characters
TI: the thickened version of BI
V: the vector with coordinates of ER lines in BI
mc: the average value corresponding to a separating line
dm: the distance measurement
mdm: the minimum value of dm

Read BI, TI
Get V from BI
For each candidate to separating line in TI
    Compute mc by equation (8)
    For each coordinate from V
        Compute dm by equation (9)
    Calculate mdm
    If mdm < tolerance (adopted 2 in this work)
        Save the coordinates of this separating line

```

where: mc represents the average value of y coordinates corresponding to an eventual separating line; c_i is the column (y coordinate) corresponding to i^{th} pixel of the separating line and N denotes the total number of pixels for the separating line.

$$dm_e = \frac{\|c_e - mc\|}{2}, \quad (9)$$

where: dm_e denotes the distance measurement between the e^{th} ER line and the average value mc corresponding to the eventual separating line and c_e is the column corresponding to e^{th} ER line.

Divide Touching Characters

Because of touching characters, the vertical projection cannot detect an exact column as division line for these characters. In such a case, the proposed idea for detecting touching characters and posterior division consists of the following two stages as presented in Figure 10.

First Stage: Detecting all the crossing points in the document

These crossing points work to indicate the exact y coordinate for the division line of touching characters. First of all, skeletons of characters in the document must be constructed to locate the crossing points (Oliveira, Lethelier, Bortolozzi, & Sabourin, 2000). We used the morphological thinning operation presented in the thinning algorithm

Figure 9. Structuring elements for locating crossing points (Nomura et al., 2005) on the skeletons of touching characters. The component “•” represents the pixel of the skeleton.

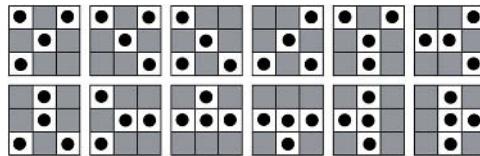


Figure 10. Algorithm for dividing touching characters

<p>Variables</p> <p><i>BI: the binary image with all the characters TN: the thinned version of BI TI: the thickened version of BI S: the matrix with structuring elements presented in Figure 9 Vc: the vector with crossing points (division lines) by TN Ve: the vector with ER lines by BI ds: the distance measurement between the nearest ER line and crossing point j: the coordinate corresponding to the current division line cd: the cost of segmentation for j coordinate cdn: the cost of segmentation for the next coordinate</i></p> <p><i>Read BI, TN, TI, S Get Vc from TN Get Ve from BI For each crossing point (division line) from Vc Get the nearest ER line from Ve Compute ds by equation (10) If $ds > \text{tolerance}$ Scrape the crossing point Else Initialize j Repeat Compute cd by equation (11) Compute c_{dn} by equation (11) Increment j Until $cd < c_{dn}$ Save j as the coordinate of division line</i></p>
--

section to obtain the skeletons of characters. The obtained skeletons are input data for detecting the crossing points. We used the set of 12 structuring elements in Figure 9 to locate possible crossing points on the skeletons.

Second Stage: Filtering the crossing points for dividing touching characters

In this stage, the crossing points, which can cause mis-division of touching characters, are scrapped. To evaluate the scrapping of these inconvenient crossing points, two criteria must be considered. The first is reference data from the ER lines. The y coordinate corresponding to a crossing point is compared with the coordinate corresponding to the nearest ER line in terms of distance measurement similarly calculated to detect the separating line of overlapping characters. The second is the cost of possible division lines within a considered range of y coordinates in the binary image. This cost is analogously calculated by equation (11) as the segmentation cost of a division line on the image. The most appropriate y coordinate for the division line is selected as one that provides the minimum segmentation cost.

Mathematical equations for dividing the touching characters are as follows:

$$ds_i = \frac{\|e_i - cp\|}{2}, \quad (10)$$

where: ds_i denotes the distance measurement between the i^{th} ER line and the column corresponding to the crossing point; e_i is the column corresponding to i^{th} ER line and cp is the column corresponding to the crossing point.

$$cd_j = \sum_{i=1}^R g(i,j), \quad (11)$$

where: cd is the segmentation cost of a division line on the image; i denotes the index for representing rows of the image; j is the index for representing the column at the image where the division line is currently positioned; R represents the total number of rows in the image and $g(i,j)$ is the value of a pixel with coordinates (i,j) . Since the image is binary, the value is zero for background pixel, and one for character pixel.

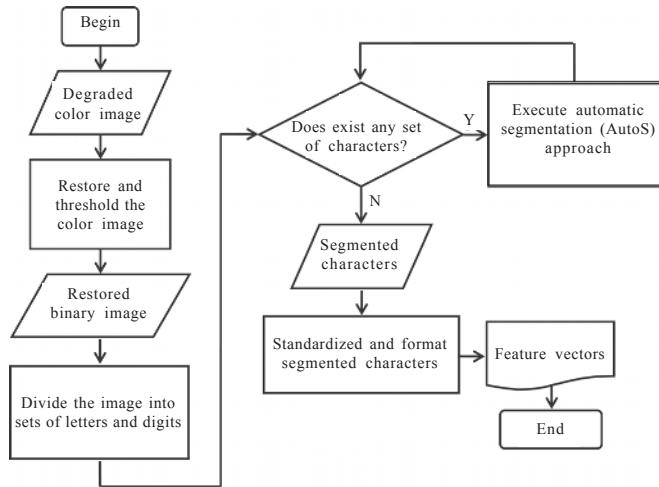
EXPERIMENTAL PROCEDURE

The following geometric information about the license plate images applies to the operations of this experimental procedure:

- A set of letters occupies the first region (almost half of a plate) and a set of digits occupies the second region on each plate.
- All the plates have three letters and four digits, whose sets constitute plate codes.

An overview of our proposed experimental procedure is verified in Figure 11. The procedure operations are described as follows:

Figure 11. Diagram of the proposed experimental procedure using AutoS



Restore and Threshold the Color Image

A color image is automatically converted into a restored binary image by using the adaptive methods (Nomura et al., 2002; Nomura, Yamanaka, Katai, & Kawakami, 2004) proposed in our previous work. These adaptive methods perform the automatic restoration of degraded images as follows:

- The adaptive lightning method (Nomura et al., 2004) detects the luminance intensity of degraded images, divides these images into regions according to the distribution of luminance intensities and adjusts the contrast of each region by adaptively increasing the intensity of luminance in order to obtain the best level of contrast.
- The shadow lighting method (Nomura et al., 2002) is composed of two parts. The first consists of adaptively locating shadowy regions of critical shadows by applying mathematical morphology techniques. The second consists of lightening these shadowy regions with the critical shadows that cause noise in binary images. In this lightening process, pixels of the shadowy regions are replaced with new higher values in order to separate from relevant objects (characters).

Figure 12 shows a sample of original degraded images in grayscale version, in noisy binary version and in restored version by the above adaptive methods.

Divide an Image into Sets of Letters and Digits

According to the above geometric information, a binary image is divided into sets of letters and digits. The exact determination of the line or y coordinate for such division is based on a vertical projection of the binary image.

Figure 12. Grayscale version of original degraded images (first row), binary version of these degraded images (second row), and restored binary images (third row)

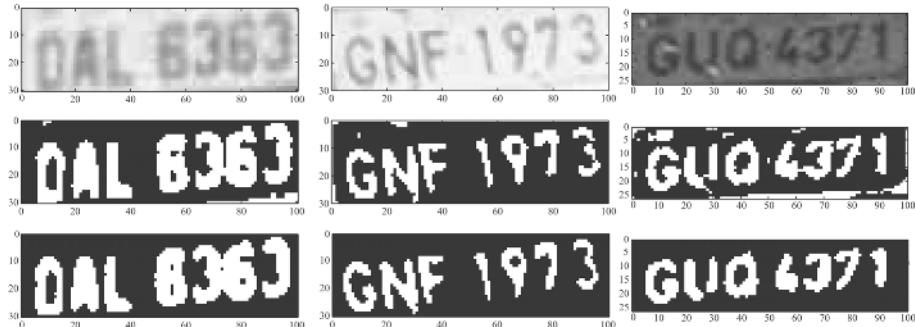


Figure 13. Binary image with broken characters, and its vertical projection

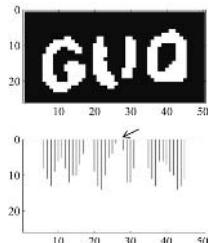


Figure 14. Segmentation results after merging the fragments of the broken character



Execute Automatic Segmentation (AutoS) Approach

The proposed approach (AutoS) to automatically segment degraded machine-printed characters is tested by implementing the architecture presented in Figure 2.

Segmentation of Naturally Pitched Characters

The first characters “G”, “G” and “8”, respectively, in Figures 13, 15 and 17 are segmented by using their pitch information from vertical projection.

Merging Process of Broken Characters

Figure 13 shows a binary image with broken characters (fragments) and its respective vertical projection. In this figure we can verify that vertical projection provides a zero value (indicated by the arrow) at y axis that does not correspond to a segmentation line.

The decision to merge fragments is based on the following information:

- Quantity of characters in each analyzed set.
- Statistical analysis presented in Merge Broken Characters section corresponding to y coordinates of eventual fragments.

Figure 14 shows the results after merging the fragments and segmenting the binary image in Figure 13 with broken characters.

Separation of Overlapping Characters

The last two characters “N” and “F” in the binary image (first image) of Figure 15 (overlapping problem is indicated by the arrow) cannot be separated by only using corresponding vertical projection (second image). To determine a possible line of separation between overlapped characters, we applied morphological thickening (third image) followed by pruning (fourth image) to the binary image in Figure 15. The exact boundary is obtained after performing statistical comparison (presented in Separate Overlapping Characters section) between this current line and the closest ER line previously constructed.

Figure 16 shows the separation results of overlapping characters.

Figure 15. Binary image with overlapping characters, its vertical projection, its thickened version and the final version without spurs

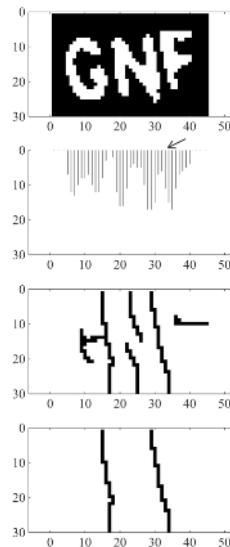


Figure 16. Segmentation results including separation of overlapping character



Figure 17. Binary image with touching characters, its vertical projection and the thinned version with skeleton of characters

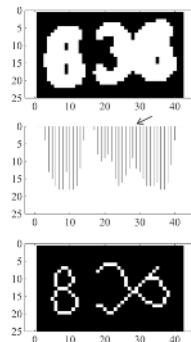


Figure 18. Segmentation results by applying the division of touching characters



Division of Touching Characters

In the binary image (first image) of Figure 17, we verify the touching characters problem that obstructs the detection of zero value (indicated by the arrow) in its vertical projection (second image).

The skeleton (third image) in Figure 17 clearly denotes the crossing point where the division line must be traced after the filtering process presented in Divide Touching Characters section. The results of touching character division are shown in Figure 18.

Standardize and Format Segmented Characters

Standardization consists of adjusting the size of each character by considering height and width proportionality. In this work, adequate size was determined as 20 rows by 15 columns of pixels. Standard segmented characters can be verified in Figures 14, 16 and 18. The formatting process converts two-dimensional representation of standard segmented characters into one-dimensional representation of feature vectors that will have 300 components. Each component corresponds to a pixel of the segmented character image. The component “0” represents the background of the image, and the component “1” represents the object (character). This operation is a simple preparation that addresses future trends of using such feature vectors as input data for automatic reading systems based on artificial neural networks (Fausset, 1994; Haykin, 1999). For this purpose, the components of feature vectors can be binary (“0” or “1”) or bipolar (“-1” or “1”) depending on the representation of target vector used for training the neural nets.

EXPERIMENTAL RESULTS

The robustness, flexibility and effectiveness of the proposed approach were evaluated by processing a set of 8,323 machine-printed character images obtained from license plate degraded images. The criterion used to evaluate whether characters are correctly segmented is based on the geometric information described in Experimental Procedure section. If the contents of the feature vector match the corresponding characters by respecting the position in the plate code, then the segmentation is considered valid. The feature vectors can be restored into characters by automatically converting each component of these vectors into the pixel of the character images. By visual inspection, the segmentation of the restored characters is evaluated according to the geometric information of license plates.

In order to show the strength and efficiency of the proposed approach, its segmentation results are compared with the results by classical method of segmentation based on vertical projection as follows:

Using Input Image with Broken Characters

The first row of Figure 19 presents the image with broken characters. The second row of Figure 19 presents the set of the segmented characters by applying the segmentation method based on vertical projection. We can verify that the second character of the image has two fragments that were segmented without an appropriate merging process. Finally, the third row of Figure 19 presents the correct segmented characters by the proposed approach because of automatic merging process of these fragments.

Using Input Image with Overlapping Characters

An image with overlapping characters is presented at the first row of Figure 20. The second row of this figure presents the set of segmented characters corresponding to the segmentation by the method based on vertical projection. We verify that the second and third characters were unnecessarily merged because of overlapping problem. This problem was solved by the proposed approach which separated the overlapping characters as the results presented at the third row of Figure 20.

Figure 19. Comparison results using image with broken characters

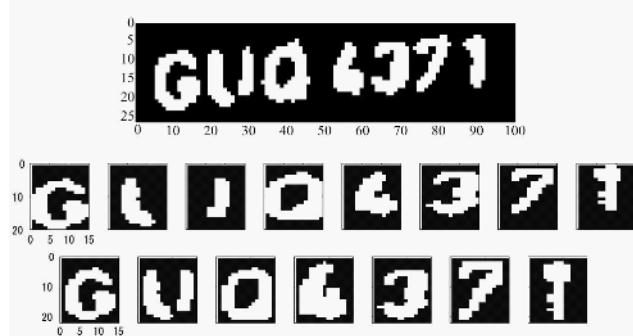


Figure 20. Comparison results using image with overlapping characters*Figure 21. Comparison results using image with touching characters*

Using Input Image with Touching Characters

The image on top of Figure 21 contains touching characters. The feature vectors of the second row of Figure 21 show that the segmentation based on vertical projection was not able to divide the fifth and sixth characters with the touching problem. On the contrary, the third row of Figure 21 shows that the proposed approach was able to solve the problem of touching characters in this input image.

Furthermore, we can verify that the segmented characters extracted by the proposed approach present uniformity in size and format. This uniformity also contributes to improving the performance of posterior recognition systems (Nomura, 2002).

In addition, the segmentation results were verified by a recognition system; however, recognition does not affect the decision-making of the segmentation process.

Table 1. Experimental results

Total of evaluated characters	Total of correctly segmented characters
8,323	7,035

Table 1 shows that a set of 7,035 character images from the test set was correctly segmented and the feature vectors extracted to be used by the posterior automatic reading system.

CONCLUSIONS

We presented an automatic approach for segmenting characters (AutoS) and extracting feature vectors from degraded images acquired by unsophisticated imaging systems installed at roads in Uberlândia city, Brazil, taking advantage of mathematical morphology's potential for solving problems of touching and overlapping characters is the main contribution of the AutoS approach. In this work, an experimental system employing the AutoS provided successful results in performing the automatic image restoration and segmentation of natural pitch, broken, overlapping or touching characters on the set of 7,035 degraded machine-printed character images processed during the experiments. Furthermore, the proposed experimental system performed well even when the characters had inclination problems or size variations. The system was also able to insert each extracted character in a standard matrix size (20x15 in the experiments) by an appropriate scaling process in addition to keeping the position of each character in the document during the entire process. A brief pattern recognition experiment based on artificial neural networks techniques confirmed that a high recognition rate could be obtained by using the feature vectors extracted by the system, based on the AutoS proposed approach. In summary, the results lead to the conclusion that AutoS as an automatic morphological approach is a promising alternative for accurately segmenting characters and extracting feature vectors from degraded machine-printed character images.

REFERENCES

- Arica, N., & Yarman-Vural, F. (1998). A new scheme for off-line handwritten connected digit recognition. In *Proceedings of the International Conference on Pattern Recognition* (Vol. 1, pp. 127-129).
- Bourbakis, N. G. (1998). ANAGNOSTIS—An automatic text reading system. *Microprocessing and Microprogramming*, 103-113.
- Casey, R. G., & Nagy, G. (1982). Recursive segmentation and classification of composite patterns. In *Proceedings of the 6th International Conference on Pattern Recognition* (pp. 1023-1026).

- Elnagar, A., & Alhajj, R. (2003). Segmentation of connected handwritten numeral strings. *Pattern Recognition*, 36, 625-634.
- Fausset, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. Englewood Cliffs, NJ: Prentice-Hall.
- Gao, H., Siu, W., & Hou, C. (2001). Improved techniques for automatic image segmentation. *IEEE Transactions on Circuits & Systems for Video Technology*, 11(12), 1273-1280.
- Gonzalez, R. C., & Woods, R. E. (1993). *Digital image processing*. Reading, MA: Addison-Wesley.
- Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.
- Jang, B., & Chin, R. (1992). One-pass parallel thinning: Analysis, properties, and quantitative evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(11), 1129-1140.
- Jung, M. C., Shin, Y. C., & Srihari, S. N. (1999). Machine printed character segmentation method using side profiles. In *IEEE SMC Conference Proceedings*, 6(10), 863-867.
- Kimura, F., Tsuruoka, S., Miyake, Y., & Shridhar, M. (1994). A lexicon directed algorithm for recognition of unconstrained handwritten words. *IEICE Transactions on Information and Systems*, E77-D, 7, 785-793.
- Lee, S., & Kim, S. (1999). Integrated segmentation and recognition of handwritten numerals with cascade neural network. *IEEE Transactions Systems Man & Cybernetics*, 29(2), 285-290.
- Lu, Y. (1995). Machine printed character segmentation—An overview. *Pattern Recognition*, 28(1), 67-80.
- Lu, Y., Haist, B., Harmon, L., Trenkle, J., & Vogt, R. (1992). An accurate and efficient system for segmenting machine-printed text. In *U.S. Postal Service 5th Advanced Technology Conference*, Washington, DC, A93-A105.
- Marinai, S., Gori, M., & Soda, G. (2005). Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1), 23-35.
- Nomura, A., Michishita, K., Uchida, S., & Suzuki, M. (2003). Detection and segmentation of touching characters in mathematical expressions. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh (pp. 126-130).
- Nomura, S. (2002). *New methods for image binarization and character segmentation applied to an automatic recognition system of number plates*. Unpublished master thesis, Faculty of Electrical Engineering, Federal University of Uberlândia, Uberlândia, Minas Gerais, Brasil.
- Nomura, S., Yamanaka, K., & Katai, O. (2002). New adaptive methods applied to printed word image binarization. In *Proceedings of the 4th IASTED International Conference Signal and Image Processing*, Kauai (pp. 288-293).
- Nomura, S., Yamanaka, K., Katai, O., & Kawakami, H. (2004). A new method for degraded color image binarization based on adaptive lightning on grayscale versions. *IEICE Trans. on Information and Systems*, E87-D, 4, 1012-1020.

- Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2005). A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, 38(11), 1961-1975.
- Oliveira, L. S., Lethelier, E., Bortolozzi, F., & Sabourin, R. (2000). A new approach to segment handwritten digits. In *Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition* (pp. 577-582).
- Park, H. S., & Ra, J. B. (1999). Efficient image segmentation preserving semantic object shapes. *IEICE Trans. Fundamentals*, E82-A, 6, 879-886.
- Salembier, P., & Pardas, M. (1994). Hierarchical morphological segmentation for image sequence coding. *IEEE Trans. Image Processing*, 3(5), 639-651.
- Serra, J. (1982). *Image analysis and mathematical morphology* (Vol. 1). London: Academic Press.
- Serra, J. (1988). *Image analysis and mathematical morphology: Theoretical advances* (Vol. 2). London: Academic Press.
- Soille, P. (1999). *Morphological image analysis: Principles and applications*. Berlin: Springer-Verlag.
- Tan, C. L., Huang, W., Yu, Z., & Xu, Y. (2002). Image document text retrieval without OCR. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(6), 838-844.
- Taxt, T., Flynn, P. J., & Jain, A. K. (1989). Segmentation of document images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 11(12), 1322-1329.

Chapter XVI

Segmentation in Food Images

Domingo Mery, Pontificia Universidad Católica de Chile, Chile

Franco Pedreschi, Universidad de Santiago de Chile, Chile

ABSTRACT

In this chapter, a robust algorithm for segmenting food imagery from a background is presented using colour images. The proposed method has three steps: (i) computation of a high contrast grey value image from an optimal linear combination of the RGB colour components; (ii) estimation of a global threshold using a statistical approach; and (iii) a morphological operation in order to fill the possible holes presented in the segmented binary image. Although the suggested threshold separates the food image from the background very well, the user can modify it in order to achieve better results. The algorithm was implemented in Matlab and tested on 45 images taken under very different conditions. The segmentation performance was assessed by computing the area A_z under the Receiver Operation Characteristic (ROC) curve. The achieved performance was $A_z = 0.9982$.

INTRODUCTION

Computer vision is a technology for acquiring and analysing an image of a real scene by computers to obtain information or to control processes (Brosnan & Sun, 2003). Computer vision has been used in the food industry for quality evaluation; detection of defects and identification, grading and sorting of fruits and vegetables, meat and fish, bakery products and prepared goods, among others (Gunasekaram & Ding, 1994; Gerrard, et al., 1996; Luzuriaga et al., 1997; Leemans et al., 1998; Sun, 2000; Shanin & Symons, 2001; Shanin & Symons, 2003). In particular, computer vision has been used to objectively

measure the colour of fried potatoes by means of grey-level values (Scanlon et al., 1994). A computer-based video system was developed to quantify the colour of potato chips in the L*a*b* colour space, which correlated well with the perception of the human eye (Segnini et al., 1999). The video image analysis technique had some obvious advantages over a conventional colorimeter, namely, the possibility of analyzing the whole surface of the chips and quantifying characteristics such as brown spots and other defects.

Generally, the automatic computer vision process, as shown in Figure 1, consists of five steps (Castleman, 1996; Pedreschi et al., 2004). The general methodology that is applied to analyse foods is briefly described below.

Image Acquisition

A digital image of the object is captured and stored in the computer. When acquiring images, it is important to consider the effect of illumination intensity and the specimen's orientation relative to the illumination source because the grey level of the pixels is determined not only by the physical features of the surface but also by these two parameters (Peleg, 1993; Chantler, 1995). Typically, a colour digital camera provides three digital images, namely, red (R), green (G) and blue (B).

Image Pre-Processing

Digital images must be preprocessed to improve their quality before they are analysed. Using digital filtering, the noise of the image can be removed and the contrast can be enhanced. In addition, in this step the colour image is converted to a greyscale image, called the intensity image (I).

Image Segmentation

The intensity image is used to identify disjointed regions of the image with the purpose of separating the part of interest from the background. This segmented image (S) is a binary image consisting only of black and white pixels, where '0' (black) and '1' (white) mean background and object, respectively. In our case, the region of interest within the image corresponds to the area where the food is located.

Measurement

Segmentation detects regions of interest inside the image or structural features of the object. Subsequently, feature extraction is concentrated principally around the measurement of geometric properties (perimeter, form factors, Fourier descriptors, invariant moments and so forth) and on the intensity and colour characteristics of regions (mean value, gradient, second derivative, texture features and so forth). The geometric features are computed from the segmented image (S), the intensity features are extracted from the intensity and the colour features from the RGB images. It is important to know ahead which features provide relevant information for the classification to be accomplished. For this reason, a feature selection must be performed in a training phase (Pedreschi et al., 2004).

Interpretation

The extracted features are interpreted using some knowledge about the analysed object. Usually, the interpretation classifies each segmented region to one of the predefined classes, which represent all possible types of regions expected in the image. A classifier is designed following a supervised training, and simple classifiers may be implemented by comparing measured features with threshold values. Nonetheless, it is also possible to use more sophisticated classification techniques such as those that carry out statistical and geometric analyses of the vector space of the features or those that use neural networks or fuzzy logic (Castleman, 1996; Jain et al., 2000).

Segmentation is an essential step in computer vision based on image analysis of foods as subsequently extracted data are highly dependent on the accuracy of this operation. In general, the automated segmentation is one of the most difficult tasks in the image analysis (Gonzalez & Wintz, 1991), because a false segmentation will cause degradation of the measurement process and therefore the interpretation may fail. Food image segmentation is still an unsolved problem because of its complex and underconstrained attributes. Figure 2 shows an example, where the input is a colour image

Figure 1. Schematic representation for computer vision used in food industry

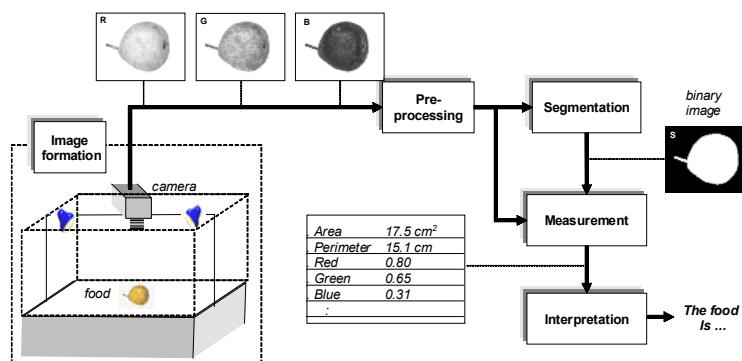
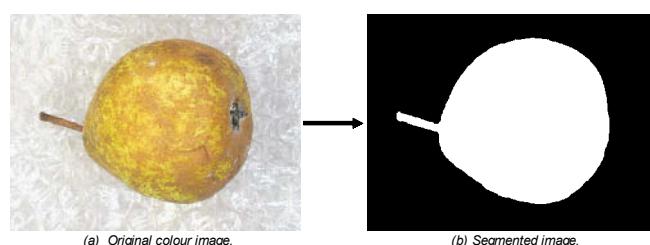


Figure 2. Segmentation process for a pear image



of a pear (with background) and the output is an image with only two colours: white for the pear and black for the background. Recently, Sun and Du (2004) developed an algorithm for segmenting complex images of many types of foods including pizza, apple, pork and potato. In this approach, the food image itself is segmented into different subregions, for example, a pizza is partitioned into cheese, ham, tomato sauce, and so forth.

In this chapter, a robust algorithm that we developed for segmenting food images from their backgrounds using colour images is presented (Mery & Pedreschi, 2005). The approach has been used to segment a large variety of foods with different backgrounds yielding very good results. The rest of the chapter is organised as follows: Section 2 describes the algorithm. Section 3 gives the Matlab code for the algorithm. Section 4 shows some experimental results obtained using the proposed algorithm. Finally, Section 5 gives a summary.

SEGMENTATION ALGORITHM

The proposed method has basically three steps: (i) computation of a high contrast grey value image from an optimal linear combination of the RGB colour components; (ii) estimation of a global threshold using a statistical approach; and (iii) a morphological operation in order to fill the possible holes presented in the segmented binary image. They will be explained in this section.

Computation of a High Contrast Monochrome Image

After the colour image acquisition, an RGB image is obtained. The RGB image is stored in three matrices, called **R**, **G** and **B**, respectively, which contain the intensity values of the red, green and blue components of the image. The corresponding intensity values for a (x, y) pixel are denoted in this paper as $R(x, y)$, $G(x, y)$ and $B(x, y)$, for $x=1, \dots, M$ and $y=1, \dots, N$, where M and N are respectively the size of the files and columns of the digital image.

There are several colour space transformations (see, for example, Hunt (1991)). They attempt to obtain a better representation of the colour. Many of these transformations have the linear form:

$$I(x, y) = k_r R(x, y) + k_g G(x, y) + k_b B(x, y) \quad (1)$$

where (k_r, k_g, k_b) are the weights that ponder the RGB components, and $I(x, y)$ is the transformed grey value of the (x, y) pixel. For instance, the chrominance value Q in the YIQ space is computed with $k_r=0.212$, $k_g=-0.523$ and $k_b=0.311$ (Gonzalez & Wintz, 1991), and a grey value image is obtained with $k_r=0.2989$, $k_g=0.5870$ and $k_b=0.1140$, where the hue and saturation information is eliminated while retaining the luminance (MathWorks, 2003).

In our approach, we use a normalised monochrome image computed as:

$$J(x, y) = \frac{I(x, y) - I_{\min}}{I_{\max} - I_{\min}} \quad (2)$$

where I_{\min} and I_{\max} are the minimum and maximum values of \mathbf{I} . Thus, the normalisation ensures that the grey values of \mathbf{J} are between 0 and 1.

An appropriate representation of the image should have a high contrast, that is, a low homogeneity. Since an image with low homogeneity will have a high variance, we seek the best combination (k_r, k_g, k_b) in (1) that maximises the variance of \mathbf{J} :

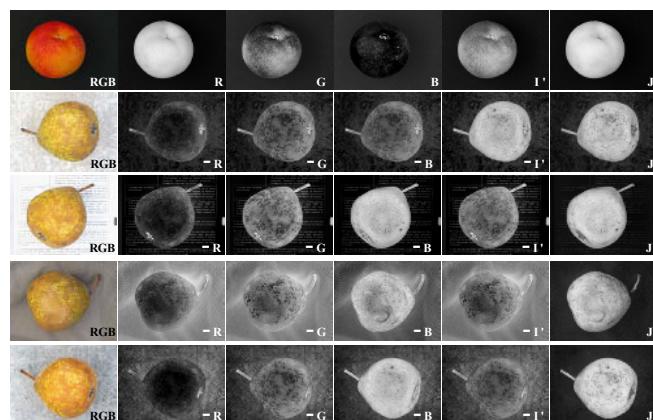
$$\sigma_J^2(k_r, k_g, k_b) \rightarrow \max \quad (3)$$

Since only the ratios $k_r : k_g : k_b$ are significant for the normalization, we can set $k_b = 1$ without loss of generality. The optimal high contrast image can be found using an exhaustive method that evaluates (3) for several combinations (k_r, k_g) (with $k_b = 1$) and takes the combination that maximises σ_J^2 . In the exhaustive search, we can use the following values for $k_r, k_g = k_0, k_0 + \Delta k, \dots, k_1$, with $k_0 = -1$, $k_1 = 1$ and $\Delta k = 0.1$. However, a better solution can be obtained using a numerical gradient method. In this case, we start with an initial guess $(k_r, k_g)_0$ and update this value using the iteration:

$$(k_r, k_g)_{i+1} = (k_r, k_g)_i + (\Delta_r, \Delta_g)_i \quad (4)$$

where $(\Delta_r, \Delta_g)_i$ is computed using the gradient of (3) evaluated at $(k_r, k_g)_i$. The iteration is interrupted, once no considerable modification of $(k_r, k_g)_{i+1}$ is achieved upon adding $(\Delta_r, \Delta_g)_i$. This multidimensional unconstrained nonlinear maximisation is included in the Toolbox for Optimisation of Matlab (MathWorks, 2000) (see, for example, function fminsearch that can be used to minimise $-\sigma_J^2$). Section 3 shows a Matlab program, called rgb2hcm, that computes the high contrast monochrome image from an RGB image. Several examples are illustrated in Figure 3, where the colour components ($\mathbf{R}, \mathbf{G}, \mathbf{B}$) of the colour image are transformed into a new optimal high contrast monochrome image \mathbf{J} . In addition,

Figure 3. Comparison between high contrast image \mathbf{J} and $\mathbf{R}, \mathbf{G}, \mathbf{B}$ colour components and greyscale image \mathbf{I}' in RGB images with different backgrounds



we show a typical greyscale image \mathbf{I}' converted from the RGB image using $k_r = 0.2989$, $k_g = 0.5870$ and $k_b = 0.1140$ (in some cases -**R**, -**G**, -**B** and -**I** are shown in order to preserve the dark background). In these examples, the Matlab command `imshow(X,[])` was employed to display image X using the whole scale, that is, the minimum value in X is displayed as black, and the maximum value as white. The greyscale image \mathbf{I}' was computed using the Matlab command `rgb2gray`. In all these examples, we observe the ability of our transformation to obtain a high contrast monochrome image. Since the high variability of the background is attenuated, the foreground can be clearly identified.

Global Threshold Estimation

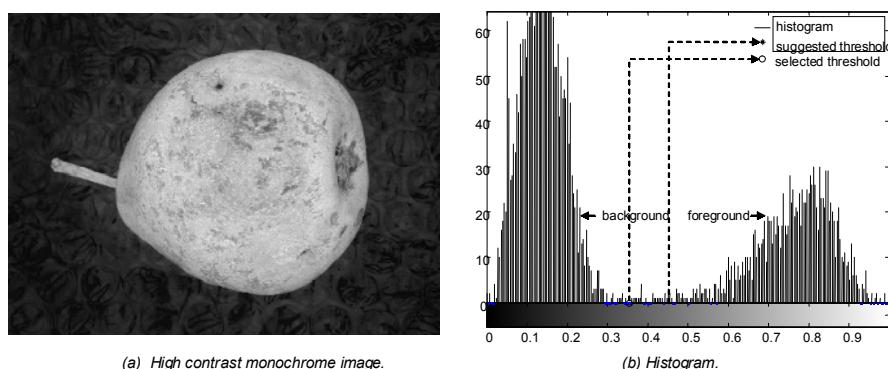
The obtained image \mathbf{J} has a bimodal histogram as shown in Figure 4, where the left distribution corresponds to the background and the right to the food image. In this high contrast image, a first separation between foreground and background can be performed estimating a global threshold t . Thus, we define a binary image:

$$K(x, y) = \begin{cases} 1 & \text{if } J(x, y) > t \\ 0 & \text{else} \end{cases} \quad (5)$$

where '1' means foreground and '0' background, defining two classes of pixels in the image. The problem is to determine a best threshold t that separates the two modes of the histogram from each other. A good separation of the classes is obtained by ensuring (i) a small variation of the grey values in each class, and (ii) a large variation of the grey values in the image (Haralick & Shapiro, 1992). The first criterion is obtained by minimising a weighted sum of the within-class variances (called *intraclass* variance $\sigma_w^2(t)$):

$$\sigma_w^2(t) = p_b(t)\sigma_b^2(t) + p_f(t)\sigma_f^2(t) \quad (6)$$

Figure 4. High contrast image and corresponding histogram



where the indices b and f denote, respectively, background and foreground classes, and p and σ^2 are respectively the probability and the variance for the indicated class. These values can be computed from the histogram.

The second criterion is obtained by maximising the between-class variance (called *interclass variance* $\sigma_B^2(t)$):

$$\sigma_B^2(t) = p_b(t)(\mu_b(t) - \mu)^2 + p_f(t)(\mu_f(t) - \mu)^2 \quad (7)$$

where μ_b , μ_f and μ indicate the mean value of the background, foreground and the whole image, respectively.

The best threshold t can be estimated by a sequential search through all possible values of t that minimise $\sigma_w^2(t)$ (or maximise $\sigma_B^2(t)$). Both criteria, however, lead to the same result because the sum $\sigma_w^2(t) + \sigma_B^2(t)$ is a constant and corresponds to the variance of the whole image (Haralick & Shapiro, 1992). Matlab computes the global image threshold by minimising the intraclass variance $\sigma_w^2(t)$. The threshold can be obtained with the function graythresh (MathWorks, 2003) (see Section 3 for details). An example is shown in Figure 5.

Morphological Operation

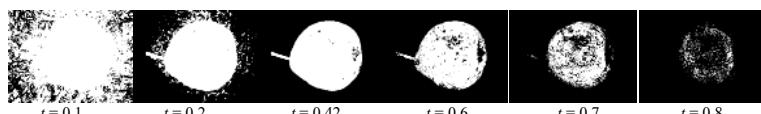
We observe in Figure 5 that the segmentation suffers from inaccuracy because there are many dark (bright) regions belonging to the foreground (background) that are below (above) the chosen threshold and therefore misclassified. For this reason, an addition morphological processing must be achieved.

The morphological operation is performed in three steps as shown in Figure 6: (i) remove small objects, (ii) close the binary image and (iii) fill the holes.

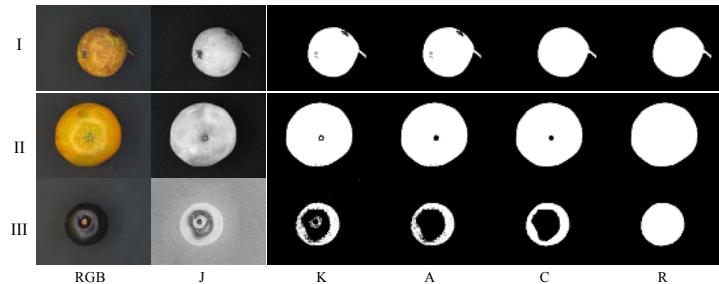
In the first step, we remove all connected regions that have fewer than n pixels (see image A in Figure 6) from binary image **K** obtained in the previous section. This operation is necessary to eliminate those isolated pixels of the background that have a grey value greater than the selected threshold. This situation may occur when there are shiny surfaces in the background that produce specular reflections. Empirically, we set $n = NM/100$, where $N \times M$ is the number of pixels in the image.

The second step *closes* the image, that is, the image is *dilated* and then *eroded*. The dilation is the process that incorporates into the foreground any background pixels that touch it. On the other hand, erosion is the process that eliminates all the boundary pixels of the foreground. The closing process (dilation followed by erosion) fills small holes and

Figure 5. Separation between foreground and background of Figure 4a for different thresholds. The value $t=0.42$ was suggested by the outlined algorithm (see histogram in Figure 4b).



*Figure 6. Morphological operations: **R****G****B**: colour image, **J**: high contrast monochrome image, **K**: binary image after thresholding, **A**: after removing of small objects, **C**: after closing process, and **R**: after filling holes; (I) pear, (II) mandarin and (III) plum*



thins holes in the foreground, connecting nearby regions, and smooths the boundaries of the foreground without changing the area significantly (Castleman, 1996) (see image **C** in Figure 6). This operation is very useful in foods that have spots in the boundary (see, for example, the pear in Figure 6).

Finally, the last operation fills the holes in the closed image (see image **R** in Figure 6). We use this operation to incorporate all pixels ‘0’ that are inside of the region into the foreground (see, for example, the mandarin and the plum in Figure 6).

The whole algorithm, with the three steps described in this section, is summarised in Figure 7.

MATLAB PROGRAMS

In this section we describe briefly the Matlab programs implemented. The main program is called SegFood (see Figure 8). We can use this program with the following instructions:

```
I=imread(file_name);
[R,E,J]=SegFood(I);
```

In this example, the image saved in *file_name* is stored in matrix **I**. The program SegFood segments image **I** in **R**. An example is shown in Figure 2. In addition, the edges of this binary image are given in **E**, and the high contrast monochrome image is stored in **J**. In this case the parameter *p* is chosen automatically by the program according to the best value for *p* obtained in our experiments by processing 45 images, that is, *p* = -0.05. However, in cases where the histogram of the image **I** is not evidently bimodal the user can change the value of parameter *p* using the following instruction:

```
[R,E,J]=SegFood(I,p);
```

Figure 7. Proposed algorithm for segmentation in food images

```

0. Read image
X = read( colour image )
R = red_component(X)
G = green_component(X)
B = blue_component(X)

1. High contrast monochrome image
 $(\hat{k}_r, \hat{k}_g) = \arg \max_{k_r, k_g} (\sigma_f^2)$ 
where
 $\rightarrow \sigma_f^2 = \text{variance}((I(k_r, k_g) - I_{\min}) / (I_{\max} - I_{\min}))$ 
 $\rightarrow I(k_r, k_g) = k_r R + k_g G + B$ 
 $\hat{I} = \hat{k}_r R + \hat{k}_g G + B$ 
 $J = (\hat{I} - \hat{I}_{\min}) / (\hat{I}_{\max} - \hat{I}_{\min})$ 

2. Global threshold estimation
 $\hat{t} = \arg \min_t (\sigma_W^2)$ 
where
 $\rightarrow \sigma_W^2 = p_b(t) \sigma_b^2(t) + p_f(t) \sigma_f^2(t)$ 
 $\rightarrow \sigma_b^2(t) = \text{variance}(J > t)$ 
 $\rightarrow \sigma_f^2(t) = \text{variance}(J \leq t)$ 
 $\rightarrow p_f(t) = \text{probability}(J > t)$ 
 $\rightarrow p_b(t) = \text{probability}(J \leq t)$ 
 $K = (J > \hat{t})$ 

3. Morphological operation
A = remove_small_objects(K)
C = close(A)
R = fill_holes(C)

```

SegFood calls three functions: `rgb2hcm`, `graythresh` and `MorphoFood`. The first one, as shown in Figure 8, computes the high contrast monochrome image by minimising the variance of a normalised image (see explanation in previous section and examples in Figure 3). The variance is computed by function `StdMonochrome` shown in Figure 8. The second function belongs to Matlab Image Processing Toolbox, and calculates the threshold of a monochrome image according to Otsu's methods (see explanation in previous section and examples in Figure 5). Finally, the third function computes the morphological operations as explained in previous section. Examples are given in Figure 6.

EXPERIMENTAL RESULTS

In this section we present the results obtained by analysing the following foods (see Figure 9): mango, almond, Nestlé cereal cluster, corn flakes, cookies, mandarin, wheat, potato chip, raisin, pear, nectarine, plum, pepino, red apple, green apple, pear, avocado, banana, orange, tomato, passion fruit (granadilla) and peanut.

Images were captured using an image acquisition system for a digital colour camera similar to that developed by Papadakis, Abdul-Malek, Kamdem, and Yam (2000), namely:

Figure 8. Matlab code (see algorithm in Figure 7)

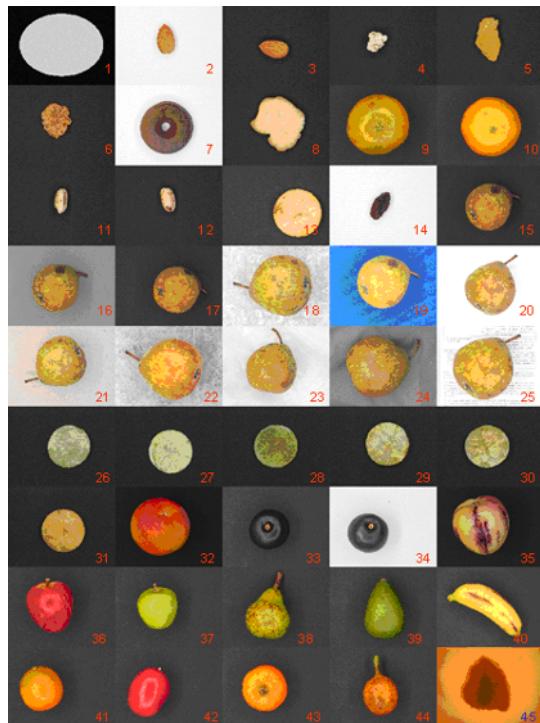
```
% File SegFood.m
function [R,E,J] = SegFood(I,p)
    J = rgb2hcm(double(I)/256);
    t = graythresh(J);
    if (~exist('p'))
        p = -0.05;
    end
    [R,E] = MorphoFood(J,t+p);

% File rgb2hcm.m
function J = rgb2hcm(RGB);
    if (size(RGB,3)==1)
        I = RGB;
    else
        RGB64 = imresize(RGB,[64 64]);
        k = fminsearch('StdMonochrome',[1 1],[ ],RGB64);
        I = k(1)*RGB(:,:,1) + k(2)*RGB(:,:,2) + RGB(:,:,3);
    end
    J = I - min(I(:));
    J = J/max(J(:));
    n = fix(size(J,1)/4);
    if (mean2(J(1:n,1:n)) > 0.3)
        J = 1 - J;
    end

% File StdMonochrome.m
function s = StdMonochrome(k,RGB)
    I = k(1)*RGB(:,:,1) + k(2)*RGB(:,:,2) + RGB(:,:,3);
    s = -std2(I)/(max(I(:))-min(I(:)));

% File MorphoFood.m
function [R,E] = MorphoFood(J,t);
    A = bwareaopen(J>t,fix(length(J(:))/100));
    C = imclose(A,strel('disk',7));
    R = bwfill(C,'holes',8);
    E = bwperim(R,4);
```

- Samples were illuminated by using four parallel fluorescent lamps (length of 60 cm) with a colour temperature of 6500°K (Philips, Natural Daylight, 18W) and a colour rendering index (Ra) near to 95%. The four lamps were situated 3.5 cm above the sample and at an angle of 45° from the food sample plane. This illumination system gave a uniform light intensity over the food plane.
- A Digital Colour Camera (DCC), Canon, Power Shot G3 (Japan) was located vertically at a distance of 22.5 cm from the sample. The angle between the camera lens axis and the lighting sources was around 45°. Sample illuminators and the DCC were inside a wood box whose internal walls were painted black to avoid the light

Figure 9. Images used in the experiments

and reflection from the room. The white balance of the camera was set using a standardised Kodak grey colour chart.

- Images were captured with the mentioned DCC at its maximum resolution (2272×1704 pixels) and connected to the USB port of a PC. Canon Remote Capture Software (version 2.6.0.15) was used for acquiring the images directly in the computer in TIFF format without compression.

The performance of our method is very sensitive to the variation of the threshold t . For this reason, we define a new threshold as $t_n = t + p$ (see Figure 4b where $p=-0.05$; in this case t and t_n are the suggested and selected thresholds, respectively). Parameter p can be given by the user (if no parameter p is given, the software assumes the default value $p=-0.05$). If $p>0$ (or $p<0$) the software will increase (or decrease) the area belonging to the background.

In order to assess the segmentation performance, the Receiver Operation Characteristic (ROC) (Duda et al., 2001) curve is analysed (see Figure 10), which is a plot of the “sensitivity” (S_n) against the “1-specificity” ($1-S_p$) defined as:

$$S_n = \frac{TP}{TP + FN}, \quad 1 - S_p = \frac{FP}{TN + FP} \quad (8)$$

where

- TP is the number of true positives (pixels of the foreground correctly classified);
- TN is the number of true negatives (pixels of the background correctly classified);
- FP is the number of false positives or false alarms (pixels of the background classified as foreground); and
- FN is the number of false negatives (pixels of the foreground classified as background).

Ideally, $S_n = 1$ and $1 - S_p = 0$, that is, all pixels belonging to the food are classified as foreground without flagging false alarms. The ROC curve permits assessment of the detection performance at various operating points of p , for example, $p = -0.3, \dots, 0.3$. The area under the ROC curve (A_z) is normally used as a measure of performance because it indicates how reliably the detection can be performed. A value of $A_z = 1$ gives perfect classification, whereas $A_z = 0.5$ corresponds to random guessing. The best value for p , denoted by \hat{p} , is chosen as the point on the ROC curve nearest to the ideal point (top left corner, that is, $S_n = 1$ and $1 - S_p = 0$). The coordinates of the nearest point are denoted by \hat{S}_n and $1 - \hat{S}_p$.

Figure 10. Analysis ROC: (a) class distribution, (b) confusion matrix, (c) ROC curve

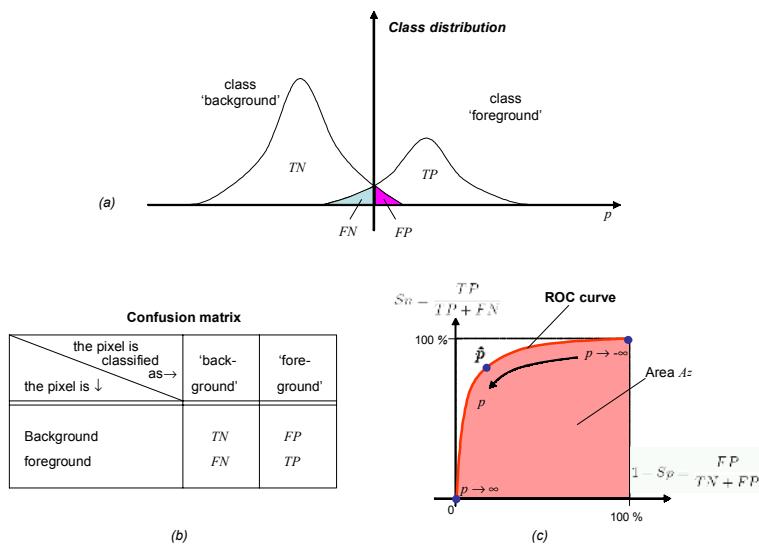


Table 1. Performance analysis on 45 images. A_z : area under the ROC curve, \hat{p} : threshold deviation, \hat{S}_n : sensitivity, $1 - \hat{S}_p$: 1-specificity, $\hat{N}M$: image size in pixels, d : image width

Description	Samples	A_z	\hat{p}	\hat{S}_n	$1 - \hat{S}_p$	N	M	d
almond	(2)	0.9983	-0.2500	0.9960	0.0003	426	568	6.2 cm
avocado	(1)	0.9853	-0.1250	0.9599	0.0000	426	568	10.1 cm
banana	(1)	0.9954	-0.3000	0.9909	0.0007	426	568	16.3 cm
chip	(7)	0.9989	-0.2607	0.9976	0.0008	390	520	6.2 cm
cluster	(1)	0.9986	-0.3000	0.9974	0.0005	426	568	6.2 cm
cookie	(2)	0.9966	-0.2875	0.9959	0.0045	426	568	6.2 cm
corn flake	(2)	0.9998	-0.1250	0.9994	0.0011	426	568	6.2 cm
green apple	(1)	0.9981	-0.2250	0.9918	0.0002	426	568	10.1 cm
mandarin	(3)	0.9998	-0.1583	0.9990	0.0004	426	568	6.2 cm
mango	(1)	0.9989	-0.3000	0.9978	0.0006	426	568	10.1 cm
puffed wheat	(2)	0.9999	-0.2750	0.9997	0.0005	426	568	6.2 cm
nectarine	(1)	1.0000	-0.2500	0.9995	0.0011	426	568	6.2 cm
orange	(1)	0.9979	-0.3000	0.9958	0.0004	426	568	10.1 cm
passion fruit	(1)	0.9951	-0.3000	0.9903	0.0006	426	568	10.1 cm
pear	(12)	0.9990	-0.1729	0.9947	0.0020	426	568	10.1 cm
peanut	(1)	0.9994	-0.1000	0.9936	0.0002	480	640	2.0 mm
pepino	(1)	0.9998	-0.2750	0.9988	0.0015	426	568	10.1 cm
plum	(2)	0.9988	-0.0500	0.9967	0.0039	426	568	10.1 cm
raisin	(1)	0.9999	-0.2500	0.9985	0.0008	426	568	6.2 cm
red apple	(1)	1.0000	-0.1750	0.9989	0.0010	426	568	10.1 cm
tomato	(1)	0.9992	-0.3000	0.9985	0.0005	426	568	10.1 cm
mean	--	0.9985	-0.2122	0.9956	0.0013	--	--	--
max	--	1.0000	0.1000	0.9999	0.0087	--	--	--
min	--	0.9853	-0.3000	0.9599	0.0000	--	--	--

In our experiments, 45 colour images of foods were analysed (Figure 9). For each image, an *ideal detection* was achieved using visual interpretation. Our methodology was to create an ideal binary image ("1" is foreground and "0" is background) according to the visual information with the software Microsoft Paint, using the biggest scale (zoom = 800%). The results obtained with our algorithm were then compared with the ideal binary image. Thus, the values for TP , TN , FP and FN were tabulated. The results obtained in each food image are summarised in Table 1, in which the number of food samples, the areas A_z and the optimal values \hat{p} , \hat{S}_n and $1 - \hat{S}_p$ are given. In order to reduce the table, an average is presented when more than one sample was evaluated. We observe that the mean sensitivity of all images was 0.9956, that is, on average 99.56% of all pixels of the foods were correctly detected. The mean quote of false alarms (pixels of the background detected as foreground) was 0.13%. In addition, Table 1 shows the dimensions in pixels of each image and the corresponding image width captured by the camera.

Analysing all images together with the same p for each image, we obtain $A_z = 0.992$. The best performance is achieved at $\hat{p} = -0.05$. In this case, $\hat{S}_n = 0.9831$ and $1 - \hat{S}_p = 0.0085$. For this reason, we chose the default value of p as -0.05.

SUMMARY

In this paper, a robust approach to segmenting food images is proposed. The approach has three steps: i) computation of a high contrast grey value image from an optimal linear combination of the RGB colour components; ii) estimation of a global threshold using a statistical approach; and iii) a morphological operation in order to fill the possible holes presented in the segmented binary image. After testing the implemented algorithm in Matlab on 45 images, the assessed segmentation performance computed from the area under the Receiver Operation Characteristic (ROC) curve was $A_z = 0.9982$.

ACKNOWLEDGMENT

Authors acknowledge financial support from FONDECYT-- Chile Project n. 1030411.

REFERENCES

- Brosnan, T., & Sun, D.-W. (2004). Improving quality inspection of food products by computer vision—A review. *Journal of Food Engineering*, 61(1), 3-16.
- Castleman, K. (1996). *Digital image processing*. Englewood Cliffs, NJ: Prentice Hall.
- Chantler M. (1995). Why illuminant direction is fundamental to texture analysis. *IEEE P Vis Image Sign*, 142, 199-206.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: John Wiley & Sons.
- Gerrard, D.E., Gao X., & Tan, J. (1996). Beef marbling and color score determination by image processing. *Journal of Food Science*, 61, 145-148.

- Gonzalez, R., & Wintz, O. (1991). *Digital image processing* (3rd ed.). MA: Addison-Wesley.
- Gunasekaram, S. (1996). Computer vision technology for food quality assurance. *Trends in Food Science and Technology*, 7(8), 245-246.
- Haralick, R., & Shapiro, L. (1992). *Computer and robot vision*. New York: Addison-Wesley.
- Hunt, R. (1991). *Measuring colour* (2nd ed.). New York: Ellis Horwood.
- Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis*, 22(1), 4-37.
- Leemans, V., Magein, H., & Destain, M. F. (1998). Defects segmentation on 'Golden Delicious' apples by using color machine vision. *Computer & Electronics in Agriculture*, 20, 117-130.
- Luzuriaga D., Balaban, M. O., & Yeralan, S. (1997). Analysis of visual quality attributes of white shrimp by machine vision. *Journal of Food Science*, 61(113-118), 130.
- MathWorks. (2000) *Optimization toolbox for use with MATLAB: Users guide*. The MathWorks Inc.
- MathWorks. (2003). *Image processing toolbox for use with MATLAB: Users guide*. The MathWorks Inc.
- Mery, D., & Pedreschi, F. (2005). Segmentation of colour food images using a robust algorithm. *Journal of Food Engineering*, 66(3), 353-360.
- Papadakis, S., Abdul-Malek, S., Kamdem, R., & Yam, K. (2000). A versatile and inexpensive technique for measuring color of foods. *Food Technology*, 54(12), 48-51.
- Pedreschi, F., Mery, D., Mendoza, F., & Aguilera, J. M. (2004). Classification of potato chips using pattern recognition. *Journal of Food Science*, 69(6), 264-270.
- Peleg, M. (1993). Fractals and foods. *Critical Reviews in Food Science and Nutrition*, 33, 149-165.
- Segnini, S., Dejmek, P., & Öste, R. (1999). A low cost video technique for colour measurement of potato chips. *Lebensm Wiss u- Technol*, 32, 216-222.
- Scanlon, M. G., Roller, R., Mazza, G., & Pritchard, M. K. (1994). Computerized video image analysis to quantify colour of potato chips. *Am Potato*, 71, 717-733.
- Shanin, M. A., & Symons, S. J. (2001). A machine vision system for grading lentils. *Canadian Biosystem Engineering*, 43(7.7-7.14).
- Shanin, M. A., & Symons, S. J. (2003). Lentil type identification using machine vision. *Canadian Biosystem Engineering*, 45(3.5-3.11).
- Sun, D. W. (2000) Inspecting pizza topping percentage and distribution by a computer vision method. *Journal of Food Engineering*, 44, 245-249.
- Sun, D. W., & Du, C.-J. (2004). Segmentation of complex food images by stick growing and merging algorithm. *Journal of Food Engineering*, 61(1), 17-266.

Chapter XVII

Segmentation via Thresholding Methodologies by Using Measure of Fuzziness towards Blind Navigation

Farrah Wong, Universiti Malaysia Sabah, Malaysia

R. Nagarajan, Northern Malaysia University College of Engineering, Malaysia

Sazali Yaacob, Northern Malaysia University College of Engineering, Malaysia

ABSTRACT

Blind navigation is specialized research directed towards the development of navigational aid for blind people to minimize assistance from sighted individuals during navigation. In this paper, two methodologies of segmentation are detailed and certain aspects of the methodologies are compared. Measure of fuzziness is applied in both the segmentation methodologies to find the threshold values. These methodologies are of an automated process resulting in the elimination of human circumvention. The segmentation methodologies have been found to work suitably for the purpose of blind navigation as shown by the results provided. The first methodology was developed for a single camera whereas the second was developed for a system of stereo cameras. A comparison in terms of results from both the methodologies is also discussed and finally, conclusions derived from the methodologies are presented.

INTRODUCTION

The unfortunate visually impaired may not have the privilege to see as normal sighted people, but the fact that these blind people can still “see” through their hearing sense is undeniable. Argument has arisen on the issue of whether the blind can “see” better through hearing—this ability is also termed “mental imagery” (Nigel, 2001). Research experiments conducted on blind and sighted individuals has deduced that blind subjects can map the auditory environment with equal or better accuracy than sighted subjects (Lessard et al., 1998; Zwiers et al., 2001). Hence, seeing through hearing is practical. The term “blind navigation” refers to a specialized research effort to help blind people navigate autonomously or with minimum assistance. The aid that is developed is termed Navigation Aid for Visually Impaired (NAVI). Single camera aid to the blind is made up of off-the-shelf equipment, namely, laptop, headgear with a digital video camera attached and headphones. Figure 1 shows the headgear system which is also known as audio-vision headgear. As for the stereo-based NAVI system (Figure 2), the hardware is comprised of two digital video cameras, a Single Board Processing System (SBPS) and a headphone. The other accessories in the NAVI system are a vest and batteries. The vest is used to contain the SBPS and batteries. The batteries, in turn, supply power to the SBPS, cameras and the headphone.

Segmentation is one of the procedures undertaken in the processing for both systems. Image segmentation is defined as the process of classifying an image into a set of disjoint regions whose characteristics, such as intensity, colour, texture, and so forth,

Figure 1. The headgear system, also known as the audio-vision headgear

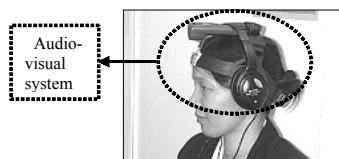


Figure 2. The stereovision-based NAVI system



are similar (Zingaretti & Carbonaro, 1999). The first methodology was developed for the single-camera system whereas the second was developed for the stereo cameras system. Measure of fuzziness is applied in both segmentation methodologies to find the threshold values.

MEASURE OF FUZZINESS

Fuzzy image processing (FIP) is a merging of two fields, namely fuzzy logic and image processing. According to Tizhoosh (1997), fuzzy image processing is the collection of all approaches that understand, represent and process images, their segments and features as fuzzy sets. One of the most common techniques of segmentation is known as thresholding (Ritter, 1996). As outlined in (Bezdek et al., 1999), there are four fuzzy approaches for image segmentation, namely segmentation via thresholding using measure of fuzziness, segmentation via clustering, supervised segmentation and rule-based segmentation. In this aid, segmentation via thresholding by using measure of fuzziness is applied. Measure of fuzziness (Zimmermann, 1996) is defined as an indication of the degree of fuzziness of a fuzzy set. The method of segmenting the image by using measure of fuzziness is referenced from Pal and Dutta (1986). Fuzzy segmentation by using fuzzy connectedness has been proposed in Carvalho et al. (1999), and a work on fuzzy clustering is reported in Setnes (2000). An analysis of fuzzy thresholding schemes for the purpose of segmentation can be referred to in Jawahar et al. (2000).

In this work, there are four measures of fuzziness applied to determine the threshold value required for implementing the segmentation procedure. The four measures of fuzziness are comprised of linear index of fuzziness, quadratic index of fuzziness, logarithmic entropy and exponential entropy (Ramar, 2000). According to Ramar (2000), the measure of fuzziness computed should give a minimum value at which its equivalent gray-level is selected as the threshold. This minimum measure of fuzziness is selected because it indicates that the picture is least fuzzy at that threshold value.

METHOD I: FUZZY-BASED IMAGE SEGMENTATION

There are three main steps in this procedure, namely Step 1: Threshold Computation; Step 2: Creation of the Quad tree Image and the “Split” Process; and Step 3: The “Merge” Process.

Step 1: Threshold Computation

The histogram of the image is created and two values, namely the peak value of the histogram, m_{max} , and its equivalent gray value, x_{max} , are obtained from the histogram. These parameters, m_{max} and x_{max} , are used to compute the “Estimated threshold.” A range in the grayscale is set to locate the correct threshold. For each region, a “Center of Gravity” value is calculated. The range is identified by taking the two adjacent “Center of Gravity” values to the “Estimated Threshold.” Each grayscale value in the threshold range is used to compute all four measures of fuzziness mentioned earlier. The compu-

Figure 3. Graph of the four measures of fuzziness in (b) for the picture of “Tiger” in (a)

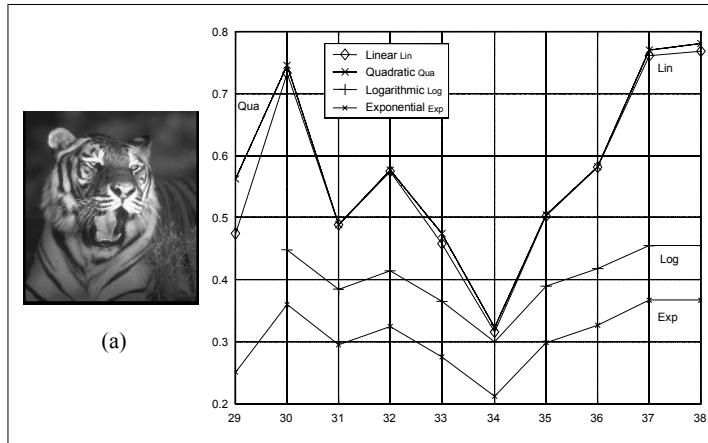
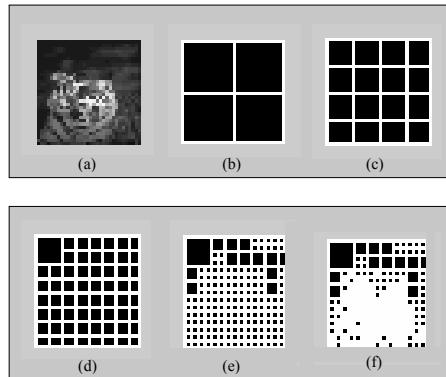


Figure 4. Simulation of the quadrant division of a 32-by-32-sized picture of “Tiger” in (a); Quadrant division of (b) 4 (c) 16 (d) 64 (e) 256 (f) 1024

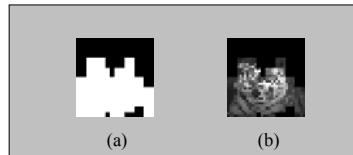


tation is summarized in a graph depicted in Figure 3b for the image of “tiger” shown in Figure 3a. For this example, the threshold value is at “34,” which occurs at the minimum of fuzziness.

Step 2: Creation of the Quad Tree Image and the “Split” Process

A quad tree image (Wong et al., 2001) is obtained by splitting an image recursively into quadrants. The creation of the quad tree image is done with the *qtdecomp* function in the MATLAB Image Processing Toolbox. This function requires a threshold value as

Figure 5. Final output (in black and white) of the FIP process shown in (a) and the whitened portion representation of the original image in (b)



obtained in Step 1 and an input image, which is the image used for computation in Step 1. This process is also called the “Splitting Process” because the image is split into different block sizes. Figure 4 shows the quad tree image creation in quadrant divisions of 4, 16, 64, 256 and 1024 for a 32-by-32-sized picture, as has been used in this work.

Step 3: The “Merge” Process

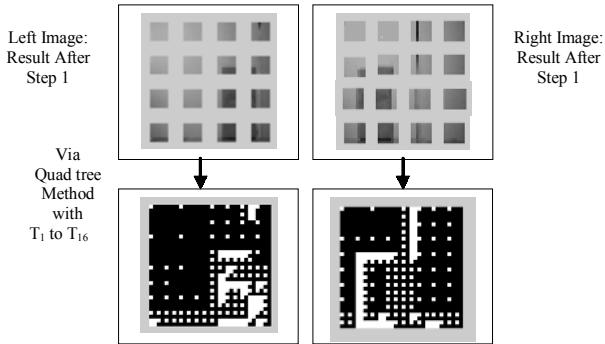
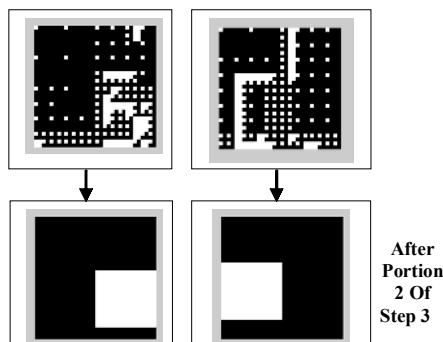
Blocks of 1-by-1-size are whitened whereas for the other block sizes they are blackened. The final quad tree image in Step 2 highlights the edges of the objects and thus, whitening these edges helps to identify objects in the image. The final output of this process is shown in Figure 5.

METHOD II: ADAPTIVE FUZZY-BASED IMAGE SEGMENTATION

There are three steps in this adaptive fuzzy-based segmentation methodology. After the stereo-pair images are captured, they go through simple preprocessing, that is, they are resized into 32-by-32-sized images and converted into grayscale. From here, they undergo the adaptive fuzzy-based segmentation. Basically, there are three major steps in the adaptive segmentation. The steps are: (1) initial processing, (2) sub-image segmentation and (3) region refining.

Step 1 in this adaptive fuzzy-based segmentation methodology is a similar process as in Step 1 of Method I, that is, to obtain a threshold value where measures of fuzziness were used. By using this threshold, T , obtained in Step 1, the initial processing on the grayscale image is undertaken. If $\max_b - \min_b > T$ (MathWorks, 1997) where, \max_b = maximum intensity value in the block and \min_b = minimum intensity value in the block, then, the grayscale image is divided into sub-images. The sub-images are obtained using the quad tree method (Ritter, 1996). The process of dividing sub-images is continued until $(2^y \times 2^y)$ sub images are reached. In this work, $y=3$ is considered and finally, the grayscale image is split into 16 numbers of Q8 sub-images. Q8 is an 8-by-8-sized quad tree sub-image. Then, the Q8 sub-images go to Step 2.

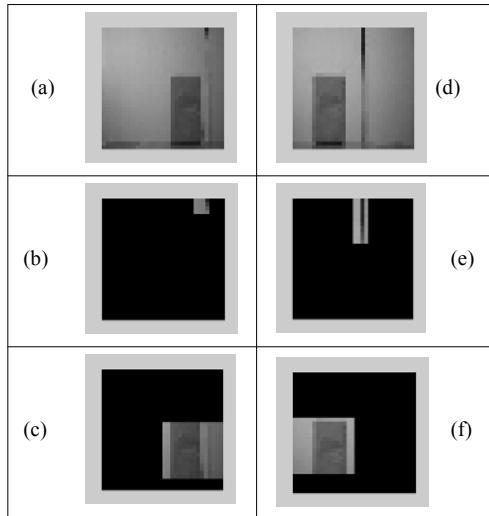
In Step 2, the sub-images are segmented using their respective sub-image threshold values, called sub-image fuzzy minimum thresholds, T_1 to T_{16} . The computations of these threshold values are similar to Step 1 except for the size of the image used. In Step 1, the

Figure 6. Results of the sub-images segmentation*Figure 7. The results of Step 3*

size of the image is Q32 and in this step, the size of the sub-image used is Q8. Q32 is a 32-by-32-sized image. These threshold values, T_1 to T_{16} , are obtained at the minimum of the quadratic index of fuzziness. The values are then applied onto the sub-images through the quad tree method. After Step 2, when all the segmented Q8 sub-images are put together, a Q32 image is obtained. The result due to Step 2 is shown in Figure 6. The quad tree method in Step 2 has created a segmented image that contains highlighted edges of objects from the foreground and also from the background.

In Step 3, these segmented sub-images will be refined so that it contains only object or objects in the foreground. The region-refining step is made up of two portions. Portion 1 is the process to identify the locality of the foreground object(s) and portion 2 consists of the growing process that increases the size of the foreground object(s) into its original size. Figure 7 shows the results of the segmentation after Step 3.

Figure 8. Comparison between the non-adaptive and adaptive segmentation



(a) and (d) original left and right images, (b) and (e) segmented left and right images—non-adaptive, (c) and (f) segmented left and right images—adaptive

A COMPARISON BETWEEN THE NON-ADAPTIVE AND ADAPTIVE METHODS

The performance of the adaptive fuzzy segmentation procedure (Method II) is now compared with that of the fuzzy-based segmentation procedure (non-adaptive: Method I). Figure 8 shows the result of the segmentation with the unwanted background blackened. The segmentation results for the non-adaptive method show that the method cannot segment images with a single global threshold despite showing a minimum value in the plot of measure of fuzziness. The graylevel that occurred at the minimum of fuzziness is selected as the threshold. Figure 8a and 8d are the left and right images of the segmented object.

CONCLUSIONS

This paper has detailed the application of one component of artificial intelligence (AI), fuzzy logic, within the field of image processing. A few examples shown in this paper ensure that the proposed method of segmentation is meaningful and applicable in the area of blind navigation. Fuzzy-based segmentation (non-adaptive) was not able to segment the sample images. One of the reasons is that the images are not generally of good quality. This has resulted in the inability to highlight edges as provided by the quad tree method.

On the other hand, the adaptive fuzzy-based segmentation method can tolerate even the lower quality images by having locally based threshold values that are less sensitive to the lower quality of image. This adaptive fuzzy-based segmentation method is useful in the stereo matching process. The matching procedure facilitates the disparity computation. The disparity is then used in obtaining the distance between the object and the cameras. The segmentation methods using fuzzy logic have been applied towards single image processing and also stereo image processing for the purpose of blind navigation. This approach of image processing is done toward helping blind people in their navigation by converting the AI segmented picture into coded sound.

ACKNOWLEDGMENTS

The authors express their gratitude to MOSTI, Malaysia for the funding under the IRPA Code No. 03-02-10-0043/EA0041 through Universiti Malaysia Sabah (UMS). Farrah Wong is grateful for the study leave and scholarship extended by UMS.

REFERENCES

- Bezdek, J.C., Keller, J., Raghu, K., & Pal, N.R. (1999). *Fuzzy models and algorithms for pattern recognition and image processing*. Boston: Kluwer Academic Publishers.
- Carvalho, B. M., Gau, C. J., Herman, G. T., & Kong, T. Y. (1999). Algorithm for fuzzy segmentation. In *Proceedings of International Conference on Advances in Pattern Recognition* (pp. 154-163). London: Springer
- Jawahar, C.V., Biswas P.K., & Ray, A.K. (2000). Analysis of fuzzy thresholding schemes. *Pattern Recognition*, 33, 1339-1349.
- Lessard, N., Pare, M., Lepore, F., & Lassonde, M. (1998). Early-blind human subjects localize sound sources better than sighted subjects. *Nature*, 395(6699), 278-280.
- MathWorks Inc. (1997). *Image processing toolbox user's guide*. Natick: MathWorks Inc.
- Nigel, T. (2001). Mental imagery. *The Stanford Encyclopedia of Philosophy*. Retrieved September 20, 2004, from <http://plato.stanford.edu/archives/win2001/entries/mental-imagery/>
- Pal, S.K., & Dutta, M.D. (1986). *Fuzzy mathematical approach to pattern recognition* (pp. 137-141). New Delhi: Wiley Eastern Ltd.
- Ramar, K. (2000). *A fuzzy neuro approach: Some studies and its applications in computer vision system*. PhD thesis. Manonmaniam Sundaranar University, India.
- Ritter, G. X. (1996). *Handbook of computer vision algorithms in image algebra* (pp. 125-141; 271-273). Boca Raton: CRC Press.
- Setnes, M. (2000). Supervised fuzzy clustering for rule extraction. *IEEE Transactions on Fuzzy Systems*, 8, 4, 416-424.
- Tizhoosh, H. R. (1997). *Fuzzy image processing: Introduction in theory and practice*. Springer-Verlag.
- Wong, F., Nagarajan, R., Sazali Y., Chekima, A., & Belkhamza, N-E. (2001). An image segmentation method using fuzzy-based threshold. In *Proceedings of 6th International Symposium on Signal Processing and its Applications* (ISSPA 2001), Kuala Lumpur (Vol. 1, pp. 144-147).

- Zimmermann, H.J. (1996). *Fuzzy set theory and its application* (2nd ed.). New Delhi: Allied Publishers Ltd., & Kluwer Academic Publishers.
- Zingaretti, P., & Carbonaro, A. (1999). On increasing the objectiveness of segmentation results. *Proceedings of International Conference on Advances in Pattern Recognition* (p. 103). London: Springer.
- Zwiers, M. P., Van Opstal, A. J., & Cruysberg, J. R. M. (2001). A spatial hearing deficit in early-blind humans. *The Journal of Neuroscience*, 21, 1-5.

Section VI: Segmentation Evaluation

Chapter XVIII

Unsupervised and Supervised Image Segmentation Evaluation

Christophe Rosenberger, Université d'Orléans, France

Sébastien Chabrier, Université d'Orléans, France

Hélène Laurent, Université d'Orléans, France

Bruno Emile, Université d'Orléans, France

ABSTRACT

Segmentation is a fundamental step in image analysis and remains a complex problem. Many segmentation methods have been proposed in the literature but it is difficult to compare their efficiency. In order to contribute to the solution of this problem, some evaluation criteria have been proposed for the last decade to quantify the quality of a segmentation result. Supervised evaluation criteria use some a priori knowledge such as a ground truth while unsupervised ones compute some statistics in the segmentation result according to the original image. The main objective of this chapter is to first review both types of evaluation criteria from the literature. Second, a comparative study is proposed in order to identify the efficiency of these criteria for different types of images. Finally, some possible applications are presented.

INTRODUCTION

As Confucius said, “*A picture is worth a thousand words.*” This quotation means that an image contains lots of information. The goal of image analysis is to automatically extract this information. Segmentation is an essential stage in image analysis since it

conditions the quality of the interpretation. This processing either consists in partitioning an image into several regions or in detecting their frontiers. The classical hypothesis is that a good segmentation result guarantees a correct interpretation. This hypothesis makes sense clearly when the gray-level of each pixel is related to the interpretation task. For example, if we consider satellite images, the localization of the different types of vegetation in the image can be achieved with a segmentation method. In this case, the relation between the segmentation and the interpretation is very close. However, much more complicated situations can be encountered. If we have an indoor image containing some objects we want to identify, a good segmentation result will determine the frontier of each object in the image. In this case, a region containing an object is not characterized by a gray-level homogeneity and the level of precision of a segmentation result affects the understanding of the image.

Many segmentation methods have been proposed in the literature in the last decades (Jain, Duin, & Mao, 2000). A major problem in segmentation is the diversity in the types of regions composing an image. Indeed, an image can be composed of uniform, textured or degraded regions. Few segmentation methods provide good results for each type of region. Moreover, the efficiency of a new segmentation method is usually illustrated by only a few segmentation results on benchmark images, such as the Lena image. The problem is that this visual evaluation is still subjective. Thus, the comparison of different segmentation methods is not an easy task. Some techniques have been proposed to facilitate the visual evaluation of a segmentation result by using a colored representation. Furthermore, different metrics have been proposed to quantify the quality of a segmentation result. In order to make an objective comparison of different segmentation methods or results, some evaluation criteria have already been defined and literature is available. Briefly stated, there are two main approaches.

On the one hand, there are supervised evaluation criteria. These criteria generally compute a global dissimilarity measure between the ground truth and the segmentation result. They need two components. The first one is a ground truth corresponding to the best and expected segmentation result. In the case of synthetic images, this ground truth is known. In other cases (natural images), an expert can manually define this ground truth (Martin, Fowlkes, Tal, & Malik, 2001). Even if these images are more realistic, one problem concerns the objectivity and variability of experts. The second component is the definition of a dissimilarity measure between the obtained segmentation result and the ground truth. In this case, the quality of a segmentation result depends on the correct classification rate of detected objects in the image (Huet & Philipp, 1998). This type of approach is based on local processing and is dedicated to a given application.

On the other hand, there are unsupervised evaluation criteria that enable the quantification quality of a segmentation result without any *a priori* knowledge (Zhang, 1996). The evaluation of a segmentation result makes sense at a given level of precision. The classical evaluation approach is based on the computation of statistical measures on the segmentation result, such as the gray-level standard deviation or the contrast of each region in the segmentation result. The problem is that most of these criteria are not adapted for texture segmentation results (Bartels & Fisher, 1995). This is a major problem as, in general, natural images contain textured areas.

These criteria can be used for different applications. The first application is the comparison of different segmentation results for a single image. We could compare the

behavior of different segmentation methods in order to choose the most appropriate for a given application. The second application is to facilitate the choice of the parameters of a segmentation method. Image segmentation generally needs the definition of some input parameters that are usually defined by the user. This sometimes arbitrary task can be automatic by determining the best parameters with the evaluation criterion. Another application is the possibility of defining new segmentation methods by optimizing evaluation criteria. Last, an evaluation criterion can be used to fuse several segmentation results of a single image or of the different bands in the multi-components case.

The aim of this chapter is to review the unsupervised and supervised evaluation criteria. Some criteria are compared in order to identify their efficiency for different kinds of images and we illustrate the interest of these evaluation criteria for different applications. Last, we discuss the emerging criteria for segmentation evaluation.

BACKGROUND

After having segmented a gray-level image by using different methods or a single one with different parameters, one has generally to determine which is the most satisfactory result. We first present in the next section a method that facilitates the visual evaluation of a segmentation result. If some evaluation criteria can guide the user in his or her decision, it will be easy to visually assess the results. The proposed method is based on the use of a colored representation. The classical presentation of different segmentation results in the gray-level domain often makes it difficult to compare their respective performances. The human eye is indeed incapable of distinguishing between similar gray-levels. The change to the color domain mainly allows us to overcome this human limitation. The second and third parts in this section are respectively dedicated to supervised and unsupervised metrics that can help the user to make his or her decision.

Visual Evaluation of a Segmentation Result

We address and illustrate here the difficulty of the visual comparison of different segmentation results relating to an image. The EDISON algorithm (Comaniciu & Meer, 2002) and a classical fuzzy C-means algorithm (FCM) (Krihnapuram & Keller, 1993) are used for this example.

One usual method to visualize a segmentation result is to attribute to each detected region the gray-level corresponding to the average value of all pixels composing the region in the original image. As shown in Figure 1, this method allows a good visual representation of a segmentation result. Nevertheless, when two adjacent regions have a similar average value, it can become difficult to evaluate the quality of the segmentation result. The human eye is indeed not able to distinguish between two regions with very close gray-levels. The different regions can then be seen to form a whole.

In order to overcome this problem, it is possible to use a colored image to display the segmentation result. Each region is represented with a random color chosen in a colormap uniformly spread among the RGB one. Let N_R be the number of regions of the image to be colored, the N_R color palette is first created. Each color is then randomly attributed to one region. This colored image allows, as shown in Figure 2, the borders of each region to be clearly distinguished.

Figure 1. Mean gray-level representations: (a) original image, (b) EDISON result, (c) FCM result



Figure 2. Mean gray-level and colored representations (EDISON result): (a) gray-level result, (b) colored result

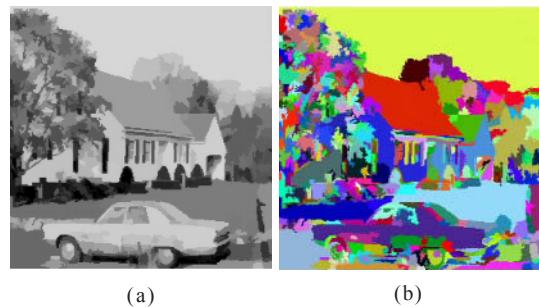


Figure 3. Random coloring (EDISON result): (a) first colored result, (b) second colored result

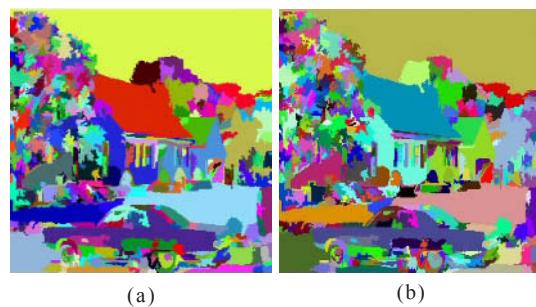
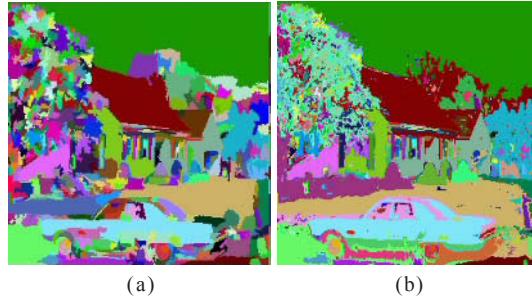


Figure 4. Matched coloring: (a) EDISON result, (b) FCM result



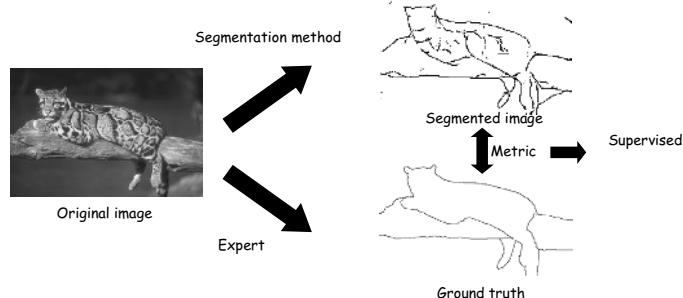
Nevertheless, the same segmentation result presented with two random palettes can appear very differently (see Figure 3). If the proposed procedure is independently applied to two segmentation results, it will be, in that case, difficult to compare them.

To solve this problem, we propose a color matching procedure to make the visual assessment easier. We first apply the above-mentioned coloring procedure to the segmentation result composed of the highest number of regions or classes. We then obtain the reference colormap and the colored representation: I_{ref} . For each segmentation result left I_k we search the region R of I_k having the highest intersection with a region X of I_{ref} and color this region R with the color of X . This color is then declared unusable and the process is repeated until no region of I_k has common pixels with any one left of I_{ref} . Finally, if some regions of I_k remain unpaired, they are randomly colored, taking one color among the I_{ref} unused ones. Figure 4 presents the colored visualizations corresponding to the segmentation results of Figure 1. It becomes, in that case, much easier to visually associate each region and to compare the quality of the different segmentation results.

Supervised Evaluation Criteria

The principle of this approach is to measure the dissimilarity between a segmentation result and a ground truth (ideal result). The different methods presented in this

Figure 5. Supervised evaluation of a segmentation result by computing a dissimilarity measure with a ground truth



section can be applied with synthetic or experts' ground truths. In the case of synthetic images, the ground truths are, of course, very reliable and have an extreme precision. For real applications, the expert ground truth is subjective and the confidence attached to this reference segmentation has to be known. Figure 5 illustrates the supervised evaluation procedure. This example is extracted from the ©Corel database. In this paragraph, we present different dissimilarity measures used within this context.

L_q and Divergence Distances

One of the first measures proposed to compare two images I_1 and I_2 , with a common domain X , is the root mean squared error (RMS):

$$RMS(I_1, I_2) = \left[\frac{1}{\text{card}(X)} \sum_{x \in X} (I_1(x) - I_2(x))^2 \right]^{\frac{1}{2}} \quad (1)$$

where $I_i(x)$ is the intensity of the pixel x in I_i . This measure can be extended to L_q distances:

$$L_q(I_1, I_2) = \left[\frac{1}{\text{card}(X)} \sum_{x \in X} (I_1(x) - I_2(x))^q \right]^{\frac{1}{q}} \quad (2)$$

with $q \geq 1$.

Different distances issued from probabilistic interpretation of images can complete the considered distance measures: the Kullback and Bhattacharyya (D_{Ku} and D_{Bh}) distances and the "Jensen-like" divergence measure (D_{Je}) based on Rényi entropies (Michel, Baraniuk, & Flandrin, 1994):

$$D_{Ku}(I_1, I_2) = \frac{1}{\text{card}(X)} \sum_{x \in X} (I_1(x) - I_2(x)) \log \left(\frac{I_1(x)}{I_2(x)} \right) \quad (3)$$

$$D_{Bh}(I_1, I_2) = -\log \left(\frac{1}{\text{card}(X)} \sum_{x \in X} \sqrt{I_1(x) \cdot I_2(x)} \right) \quad (4)$$

$$D_{Je}(I_1, I_2) = J_1 \left(\frac{I_1(x) + I_2(x)}{2}, I_1(x) \right) \quad (5)$$

with

$$J_1(I_1(x), I_2(x)) = H_\alpha(\sqrt{I_1(x) \cdot I_2(x)}) - \frac{H_\alpha(I_1(x)) + H_\alpha(I_2(x))}{2} \quad (6)$$

where H_α corresponds to the Rényi entropies parameterized by $\alpha > 0$.

If these measures permit a global comparison between two segmentation results, they sometimes express the present deformations in a very inaccurate way. For example,

they equally penalize a same intensity difference without taking into account the concerned gray-level domain. This does not correctly transcribe the human visual perception. On the other hand, the global position information does not intervene in the distance computation. Thus, if the same object appears in the two images with a simple translation, the distances will increase in an important way.

Hausdorff's Distance

This criterion (Beauchemin, Thomson, & Edwards, 1998) measures the distance between two pixel sets: $I_s = \{s_1, \dots, s_n\}$ and $I_t = \{t_1, \dots, t_m\}$:

$$H(I_t, I_s) = \text{Max}(h(I_t, I_s), h(I_s, I_t)) \quad (7)$$

where

$$h(I_t, I_s) = \text{Max}_{t_i \in I_t} \min_{s_i \in I_s} \|t_i - s_i\| \quad (8)$$

If $H(I_t, I_s) = d$, this means that all the pixels belonging to I_t are not further than d from some pixels of I_s . This measure is theoretically very interesting. It indeed gives a good similarity measure between the two images. On the other hand, this method is noise sensitive.

Baddeley's Distance

Baddeley (1992) defines an alternative measure for binary images from the Hausdorff's distance. Letting X be the common domain of two segmentation results I_1 and I_2 , the author proposes, in order to reduce the noise sensitivity, to replace the Max operator by a mean value computation:

$$D_B(I_1, I_2) = \left[\frac{1}{\text{card}(X)} \sum_{x \in X} |d(x, I_1) - d(x, I_2)|^P \right]^{\frac{1}{P}} \quad (9)$$

where I_1 and I_2 correspond to the contour pixels sets of the two segmentation results, $d(x, I) = \min_{y \in I} d(x, y)$ and $P \geq 1$. When P gets near to infinity, the Baddeley's distance is close to the Hausdorff's one.

Kara-Fhala (1995) gives a dissimilarity measure between two segmentation results which corresponds to an extended version of the Baddeley's distance:

$$D_B(I_1, I_2) = \left[\frac{1}{\text{card}(X)} \sum_{x \in X} |f_{I_1, I_2}(x)|^P \right]^{\frac{1}{P}} \quad (10)$$

with

$$f_{I_1, I_2}(x) = \sum_{i=1}^{\text{card}(R_{I_1})} d_w(x, R_{I_1}^i) - \sum_{i=1}^{\text{card}(R_{I_2})} d_w(x, R_{I_2}^i) - (\text{card}(R_{I_1}) - \text{card}(R_{I_2})) * L_w \quad (11)$$

where d_w is a limited distance, L_w its upper bound and $R_{I_s}^i$ is the i^{th} region of the segmentation result I_s . This method provides a global comparison of two segmentation results obtained with a region detection approach. The use of a limited distance makes it less sensitive to noise. Wilson, Baddeley, and Owens, (1997) extend the proposed measure from binary images to gray-level ones. Further works (Zamperoni & Starovoitov, 1996) and (Coquin, Bolon & Chehadeh, 1997) are based on a local distance computed on a neighborhood, using the pixels coordinates and their gray-level.

Vinet's Distance

This distance (Vinet, 1991) gives a dissimilarity measure between two segmentation results. If $C_1 = \{c_1^1, \dots, c_{NC_1}^1\}$ and $C_2 = \{c_2^2, \dots, c_{NC_2}^2\}$ are the two classes sets to be compared, the superposition table $T(C_1, C_2)$ is computed:

$$T(C_1, C_2) = [\text{card}(c_i^1 \cap c_j^2), i=1\dots NC_1, j=1\dots NC_2] \quad (12)$$

The components maximizing $\text{card}(c_i^1 \cap c_j^2)$ are conserved. Let C' be the selected components:

$$D(C_1, C_2) = N - \sum_{C'} \text{card}(c_i^1 \cap c_j^2) \quad (13)$$

where N is the number of pixels in the segmentation results. This method can be easily computed and corresponds to the computation of the correct classification rate. However, all the classes are not systematically matched, the biggest ones being privileged.

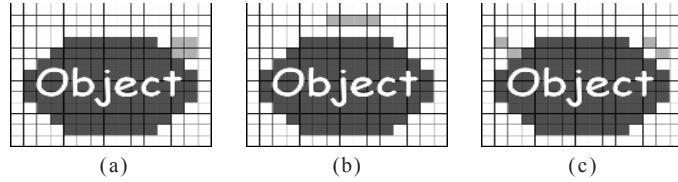
Pratt's Figure of Merit

This criterion (Pratt, Faugeras, & Gagalowicz, 1978) corresponds to an empirical distance between the ground truth contours I_t and those obtained with the chosen segmentation result I_s :

$$FOM(I_t, I_s) = \frac{1}{\text{Max}\{\text{card}(I_t), \text{card}(I_s)\}} \sum_{i=1}^{\text{card}(I_s)} \frac{1}{1+d^2(i)} \quad (14)$$

where $d(i)$ is the distance between the i^{th} pixel of I_s and the nearest pixel of I_t . This criterion, which has no theoretical proof, is not symmetrical and is sensitive to over-segmentation and localization problems. On the contrary, it does not express under-segmentation or shape errors. Figure 6 presents different situations having the same number of misclassified pixels and leading to the same criterion value. The four isolated pixels depicted in Figure 6a should belong to the object and to the background in Figure 6b. The proposed criterion considers these situations as equivalent although the consequences on the object size and shape are very different. Moreover, the criterion does not discriminate isolated misclassified pixels (Figure 6c) or a group of such pixels (Figure 6b), though the last situation is more prejudicial. However, this measure is one of the most used descriptors.

Figure 6. Different situations having the same number of misclassified pixels and leading to the same criterion value



Peli and Malah Measures

In order to evaluate a contour detection method, Peli and Malah (1982) use two statistical criteria characterizing the mean and variance of the detection error:

$$M = \frac{1}{\text{card}(I_s)} \sum_{i=1}^{\text{card}(I_s)} d(i) \quad (15)$$

$$V = \frac{1}{\text{card}(I_s)} \sum_{i=1}^{\text{card}(I_s)} d^2(i) \quad (16)$$

where $d(i)$ is the distance between the i^{th} pixel of the chosen segmentation I_s and the nearest pixel in the contour of the ground truth I_t .

Odet's Criteria

Different measurements have been proposed in Odet, Belaroussi, and Benoit-Cattin (2002) to estimate various errors in binary segmentation results:

$$ODI = \frac{1}{N_o} \sum_{k=1}^{N_o} \left(\frac{d_o(k)}{d_{TH}} \right)^n \quad (17)$$

$$ODP = \frac{1}{N_o} \sum_{k=1}^{N_o} \left(\frac{d_o(k)}{d_{TH}} \right)^n * \text{sign}(d_o(k)) \quad (18)$$

$$UDI = \frac{1}{N_u} \sum_{k=1}^{N_u} \left(\frac{d_u(k)}{d_{TH}} \right)^n \quad (19)$$

$$UDP = \frac{1}{N_u} \sum_{k=1}^{N_u} \left(\frac{d_u(k)}{d_{TH}} \right)^n * \text{sign}(d_u(k)) \quad (20)$$

where

- $d_o(k)$ is the distance between the k^{th} pixel belonging to the segmented contour and the nearest pixel of the reference contour;
- $d_u(k)$ is the distance between the k^{th} non-detected pixel and the nearest one belonging to the segmented contour;
- N_o corresponds to the number of over-segmented pixels;
- N_u corresponds to the number of under-segmented pixels;
- d_{TH} is the maximum distance allowed to search for a contour point; and
- n is a scale factor which permits different weighting of pixels according to their distance from the reference contour.

Among them, two divergence measures seem to be particularly interesting: *ODI* and *UDI*. The *ODI* criterion evaluates the divergence between the over-segmented pixels and the reference contour. The *UDI* one estimates the divergence between the under-segmented pixels and the calculated contour. The sign of the $d_o(k)$ and $d_u(k)$ distances define the relative position for the over- and under-segmented pixels. The threshold d_{TH} which has to be set for each application, allows the researcher to take the pixels into account differently with regard to their distance from the reference contour. The exponent n differently weights the estimated contour pixels that are close to the reference contour and those having a distance to the reference contour close to d_{TH} . With a small value of n , the first ones are privileged, leading to a precise evaluation.

Multi-Criteria Method

Correia and Pereira (2000) propose a measurement of segmentation quality that is initially dedicated to video sequences. This evaluation criterion is individually applied to each object detected in the scene. It mainly rests on the fact that a segmentation result should have contours very similar to the reference ones. When a perfect correspondence is not achieved, different characteristics can be taken into account:

- **Shape similarity:** The number of misclassified pixels and their distance to the reference contour are here considered;
- **Geometrical similarity:** The size and position of the reference and of the segmented contours are compared, scale differences are also estimated;
- **Inside similarity:** The complexity of the detected object surface is compared to the reference one; and
- **Statistical similarity:** Some statistical descriptors are estimated and compared in both cases.

Different weights are assigned to each similarity, taking into account the eye sensitivity and the relative importance accorded by experts to each case. The shape similarity is usually assigned the highest factor. This criterion reveals itself efficient when the object is predominant in the scene.

Ramon's Multi-Features Quality Measurement

This technique (Roman-Roldan, Gomez-Lopera, Atae-allah, Martinez-Aroza, & Luque-Escamilla, 2001) combines the computation of an objective divergence (between the extracted contours and those proposed by the experts) and a subjective evaluation

of the different possible errors. Each kind of error is modified by a penalty coefficient translating the relative importance attached to this error by the experts. The measurement is defined as:

$$Q = K \left[w \sum_{FP} \frac{1+b.n_b}{1+p.n_e + i_{bh}.n_h} + \sum_{FN} \frac{1+h.n_h}{1+c_{Euler}.i_{hb}.n_b} \right] \quad (21)$$

where

- FP corresponds to pixels that are wrongly defined as contour pixels;
- FN corresponds to contour pixels that have not been detected;
- n_b : number of FP contained in a $8*8$ neighborhood of one FP ;
- n_h : number of FN contained in a $8*8$ neighborhood of one FN ;
- n'_{bh} : number of FP directly connected to the central FN ;
- n'_{hb} : number of FN directly connected to the central FP ;
- n_e : number of real contour pixels contour contained in a $8*8$ neighborhood of one FP ; and
- $K, w, b, h, p, c_{Euler}, i_{bh}, i_{hb}$ correspond to normalization and weighting factors.

The lower the value of the criterion Q is, the better the segmentation result. The segmentation results are ranked by a group of experts. This order and the one obtained with the quality measurement Q are compared. The expression of the quality measurement contains a set of coefficients which are first determined on a benchmark and which can be modified according to a specific application or to different judges.

Evaluation Based on ROC Curves Analysis

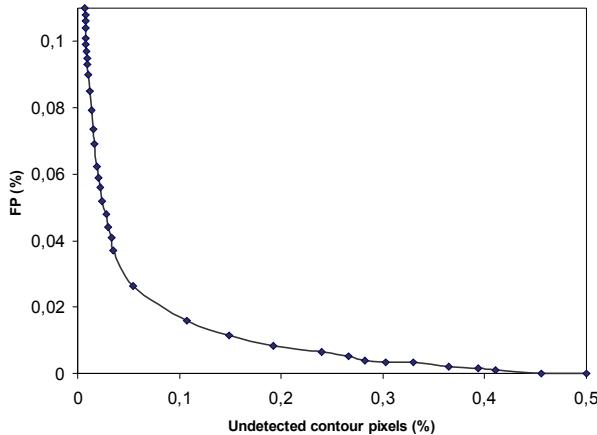
This method, proposed in Bowyer (2001) is based on receiver operating characteristic (ROC) curves. A ground truth is defined by the experts for each image and is composed of three areas: “contour area,” “empty area” (zone where no contour should be detected) and “neutral area” (zone in which inside content is unknown):

- If a segmentation method localizes a contour pixel in a “contour area,” it is labeled as TP ;
- If a segmentation method localizes a contour pixel in an “empty area,” it is labeled as FP ; and
- If a segmentation method localizes a contour pixel in a “neutral area,” it is not taken into account.

The evaluation method can then be split up into two steps:

- The detected contours are first compared to the theoretic ones in order to list the TP and FP numbers; and
- The segmentation method parameters are then adapted to obtain the learning ROC curve of an image. The parameter space is covered and the TP/FP rate is maximized. The ROC curves present the percentage of undetected contour pixels in X-axis and the percentage of FP in Y-axis (see Figure 7 for an example of such a curve).

Figure 7. Example of a possible ROC curve



The area under the ROC curve has to be minimized. The parameter values are first examined in a rough way. Their values are then refined as long as the improvement of the results remains over 5%.

Pareto Front

This method, proposed in Everingham, Muller, and Thomas (2001), allows the comparison of different algorithms a with different parameters vectors p . It is applied on an image set S and combines accuracy functions in a global function Φ :

$$H(a_p, D) = \Phi(f_1(a_p, D), \dots, f_m(a_p, D), c_1(a_p, D), \dots, c_n(a_p, D)) \quad (22)$$

where

- $f_i(a_p, D)$ corresponds to individual accuracy function (monotonic and increasing) in connection with a particular aspect of the algorithm a ; and
- $c_i(a_p, D)$ corresponds to individual cost function, which can be seen as negative accuracy function, in connection with a particular aspect of the algorithm a .

The Φ function can be defined as a linear function of the f_i and c_i . A projection of function Φ for each (f_i, c_i) pair can be drawn, the other's accuracy and cost functions being set to 0. Figure 8a presents an example of the obtained projection for a defined set of parameters, uniformity as accuracy function and number of regions as cost function. The best choices for the parameters are presented on the so-called Pareto front (see Figure 8b). The determination of this front requires the exploration of all possible values for each parameter in p and the evaluation of the chosen accuracy functions on the test set S . The

Figure 8. (a) accuracy/cost functions projection, (b) Pareto front

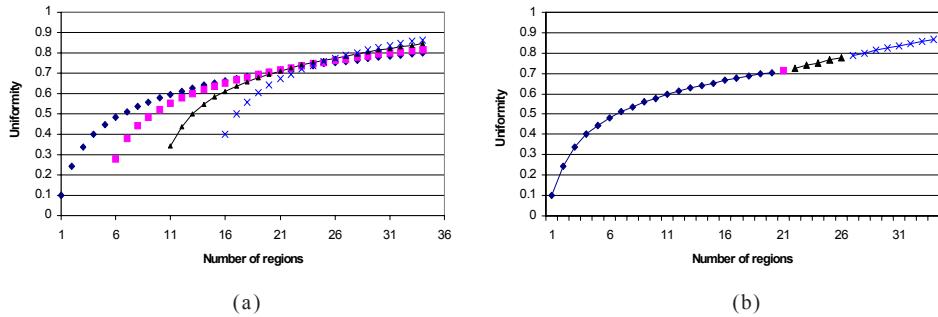
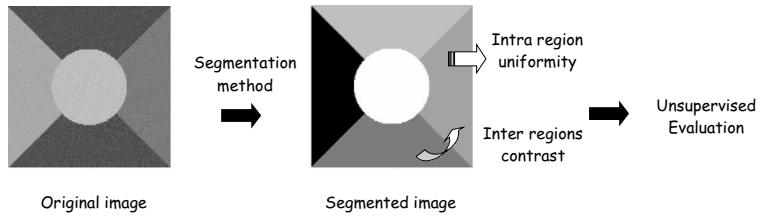


Figure 9. Computation on statistics on the segmentation result for its unsupervised evaluation



stability and the performance variations of $H(a_p, D)$ can also be considered. An algorithm with good stability over the test set S but a poor mean value may be preferred to other one presenting important variations and a higher mean value for the criteria.

Unsupervised Evaluation

Without any *a priori* knowledge, it is possible to quantify the quality of a segmentation result by using unsupervised criteria. Most of these criteria compute statistics on each region in the segmentation result. The majority of these quality measurements are established in agreement with the human perception. For region segmentation, the various criteria take into account the intra-region uniformity and inter-regions contrast (see Figure 9).

Intra-Region Uniformity Criteria

One of the most intuitive criteria being able to quantify the quality of a segmentation result is the intra-region uniformity. Weszka and Rosenfeld (1978) proposed such a criterion that measures the effect of noise on the evaluation of some thresholded images. Based on the same idea of intra-region uniformity, Nazif and Levine (1984) also defined a criterion that calculates the uniformity of a characteristic on a region based on the variance of this characteristic:

$$U = 1 - \sum_{k=1}^{N_R} \sum_{s \in R_k} \left[I(s) - \frac{1}{A_k} \sum_{s \in R_k} I(s) \right]^2 = 1 - \sum_{k=1}^{N_R} \sigma_k^2 \quad (23)$$

where N_R is the number of regions, $I(s)$ is the gray-level of the pixel s or any other characteristics (color, texture...), σ_k^2 is the gray-level variance of the region R_k and A_k its surface.

A standardized uniformity measure was proposed by Sahoo, Soltani, Wong, and Chen (1988). Thereafter, Levine and Nazif generalized this formula. Based on the same principle, the measurement of homogeneity of Cochran (Cocquerez & Devars, 1985) gives a confidence measure on the homogeneity of a region:

$$\epsilon = \frac{\max_k \sigma_k^2}{\sum_k \sigma_k^2} \quad (24)$$

where σ_k^2 is the gray-level variance of the region R_k . The region R is homogeneous if $\epsilon < T$, where T is an arbitrary threshold. The arbitrary choice of a threshold limits the evaluation quality.

It is also possible to use different attributes (texture, shape, etc.) and to estimate their dispersion of each region by the same approach.

Another criterion measuring the intra-region uniformity was developed by Pal and Pal (1989). It is based on a thresholding that maximizes the local second order entropy of object regions and the background:

$$H(T) = - \sum_{a=0}^T \sum_{b=0}^T p_{ab} \ln(p_{ab}) \quad (25)$$

where p_{ab} corresponds to the occurrence probability of the gray-level pair couple (a,b) in the object or background and T is a threshold value.

In the case of slightly textured images, these intra-region uniformity criteria proved to be effective and very simple to use. However, the presence of textures in an image generates bad results because small regions are privileged.

Inter-Regions Contrast

Complementary to the intra-region uniformity, Levine and Nazif (1985) defined a gray-level contrast measurement between two regions to evaluate the dissimilarity of regions in a segmentation result. The formula of total contrast is defined as follows:

$$C = \frac{\sum_{i=1}^{N_R} w_i \sum_{j=1}^{N_R} \frac{l_{ij}}{l_i} \frac{|m_i - m_j|}{m_i + m_j}}{\sum_{i=1}^{N_R} w_i} \quad (26)$$

where N_R is the number of regions, w_i is a weight associated to each region that can be its surface, m_i is the average gray-level of the region R_i , l_i is the length of the perimeter of the region R_i and l_{ij} the length of the border between the regions R_i and R_j . This type of criterion has the advantage of penalizing over-segmentation.

Intra- and Inter-Regions Contrast

Liu and Yang (1994) proposed a criterion considering that:

- the regions must be uniform and homogeneous;
- the interiors of the regions must be simple, without too many small holes; and
- the adjacent regions must present significantly different values for the uniform characteristics.

Borsotti, Campadelli, and Schettini (1998) identified limitations of this evaluation criterion and modified it, so as to penalize the segmentation results presenting many small regions as well as heterogeneous regions. These modifications enable the criteria to be more sensitive to small variations of segmentation:

$$Q(I) = \frac{1}{\alpha A} \sqrt{N_R} \sum_{k=1}^{N_R} \left[\frac{e_k^2}{1 + \log(A_k)} + \left(\frac{\chi(A_k)}{A_k} \right)^2 \right] \quad (27)$$

where $\chi(A_k)$ corresponds to the number of areas having a surface A_k , α a normalization factor, A the surface of the image and e_k the sum of the Euclidean distances between the color vectors of the pixels of the region R_k in the segmented image.

Zéboudj (1998) proposed a measure based on the combined principles of maximum inter-regions contrast and minimal intra-region contrast measured on a pixel neighborhood. One defines $c(s,t) = \frac{|I(s) - I(t)|}{L-1}$ as the contrast between two pixels s and t , with I_s the gray-level of the pixel s and L the maximum of the gray-level. The interior contrast $CI(i)$ of the region R_i is defined as follows:

$$CI(i) = \frac{1}{A_i} \sum_{s \in R_i} \max \{c(s,t), t \in W(s) \cap R_i\} \quad (28)$$

where A_i corresponds to the surface of the region R_i and $W(s)$ the neighborhood of the pixel s .

External contrast $CE(i)$ of a region R_i is defined as follows:

$$CE(i) = \frac{1}{l_i} \sum_{s \in F_i} \max \{c(s,t), t \in W(s), t \notin R_i\} \quad (29)$$

where l_i is the length of the border F_i of the region R_i . Lastly, the contrast of the region R_i is defined by the measurement $C(R_i) \in [0,1]$ expressed as follows:

$$C(R_i) = \begin{cases} 1 - \frac{CI(i)}{CE(i)} & \text{if } 0 < CI(i) < CE(i) \\ CE(i) & \text{if } CI(i) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

The global contrast is finally: $\frac{1}{A} \sum_{i=1}^{NR} A_i \times C(R_i)$, where A is the number of pixels in the image. This criterion has the disadvantage of not correctly taking into account the strongly textured regions.

Adaptive Criteria of Region Homogeneity

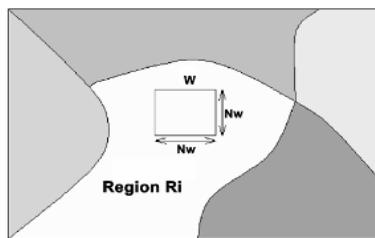
Considering the types of regions (textured or uniform) in the segmentation, Rosenberger (1999) presents a criterion that enables the estimation of intraclass homogeneity and interclass contrast. The proposed criterion quantifies the quality of a segmentation result as follows:

$$F(I) = \frac{\bar{D}(I) + 1 - \underline{D}(I)}{2} \quad (31)$$

where $\underline{D}(I)$ corresponds to the total intraclass disparity that quantifies the homogeneity of each region of image I , and $\bar{D}(I)$ corresponds to the total interclass disparity that measures the disparity of each region of the image I face to the others depending on the type of each region (uniform or textured).

The calculation of the disparity is adapted (see Figure 10). The originality of this criterion lies in its capacity to evaluate segmentation results of textured images.

Figure 10. Adaptive analysis of a region by a sliding window



Evaluation of a Contour Result

The Levine and Nazif (1985) criteria enable the evaluation of two types of contours (contours separating two objects from the scene and contours partitioning one object in different regions) without any color or texture difference between them.

Evaluation criteria in the continuous case were developed by Canny and extended by Demigny and Kamlé (1997) to the discrete case. These criteria were used for the development of optimal derivative filters for contour detection of the level type (Canny' filters, Deriche and Shen-Castan, to quote only the principal ones).

Three measurements using the fuzzy factor of contours defined by Han and Kim (2002), similar to ambiguity distance, can be used to determine the precision of contours detection: the existence, localization and arrangement.

Kitchen and Rosenfeld (1984) proposed an evaluation criterion of the contours map based on the local coherence of contours. This coherence is evaluated on the basis of two criteria of contours characterization: continuity and thinness. Another, more complex measure was proposed by Tan, Gelfand, and Delp (1992).

Lastly, there are often cases where a complete segmentation of the whole image is not necessary, but the user is rather interested in detecting the contour of a certain object in the image (or a part of an object). Therefore, evolved models (Spinu, 1997) are often used. They are founded on the characterization of a contour structure (namely an arc or a curve) by a functional computation (energy). The model of active contours (snakes for example) is one of most widely known, making it possible to model a contour like a curve (closed or not). The optimization of the criterion is used to determine the optimal contour of objects.

Markov Model

Geman and Graffigne (1987) showed that a segmentation result can be represented by a Markov field using binary contours map. Applied within a Bayesian framework, energy associated with the field result generally corresponds to an *a posteriori* probability of realization of this field. A good quality of a segmentation result corresponds to a high probability of realization of the field.

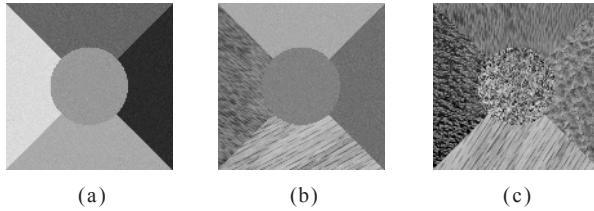
COMPARATIVE STUDY AND APPLICATIONS

Comparative Study

In this section, we present a comparative study of some relevant evaluation criteria and we illustrate their interest in the supervised and unsupervised cases. In the first part, we present the experimental protocol we used. Second, we present the comparison results in the supervised case and then in the unsupervised one.

Experimental Protocol

In the supervised case, we compared five evaluation criteria previously presented as the state-of-the-art (Ramon's multi-features quality measurement, Pratt, Haussdorff,

Figure 11. Examples of synthetic images: (a) uniform, (b) mixed, and (c) textured

Odet ODI and Odet UDI) on an image database composed of 100 natural images (Chabrier, Laurent, Emile, Rosenberger & Marché, 2004). This is an extract of 100 images from the ©Corel natural image database. Many experts segmented them and these segmentation results (taken from the database of Berkeley (Martin, Fowlkes, Tal, & Malik, 2001)) are merged for each image constituting the reference (ground truth). Each image of this database is also segmented by nine filters used as contour detection methods: canny, gradient magnitude, texture gradient, brightness gradient, color gradient, brightness/color/texture gradient, brightness/texture gradient, oriented energy and second moment matrix.

In order to compare the behavior of the different criteria toward under and over segmentation, three thresholds have been taken into account for each real image of this database to simulate various situations, such as under- and over-segmentation or a compromise between these extreme situations. This gives us 2,700 real segmentation results for the comparison of the five criteria and 100 merged expert results.

In the unsupervised case, we compared seven evaluation criteria previously presented in the state-of-the-art (criteria of Zéboudj, Borsotti, Rosenberger, Levine & Nazif (Intra), Levine & Nazif (Inter), Levine & Nazif (Intra-Inter) attributes dispersion) on a database composed of synthetic images (the evaluation criterion based on attributes dispersion is defined by Levine and Nazif by using the disparity of texture attributes instead the gray-level of pixels). This synthetic image database allows for a known reference (ground truth) to compare the results of the evaluation criteria. Each image of this database is composed of five regions: 100 images are composed of five uniform regions; 100 are composed of five textured regions and 100 with both types of regions (three uniform and two textured) (Chabrier, Rosenberger, Laurent, Emile, & Marché, 2004). An example of such synthetic images is given in Figure 11.

Two segmentation methods were used to segment these images: Fuzzy C-means (Krihnapuram & Keller, 1993) with three types of parameters (adapted for each type of image) and EDISON software (Comanicu & Meer, 2002). This gives us 1,200 segmentation results for the comparison of the seven criteria. To be able to evaluate the evolution of these criteria, we used a supervised criterion (Vinet's measure) as reference that measures the correct classification rate.

Supervised Evaluation

Figure 12 presents three images from the ©Corel natural image database. For each one, its expert segmentation result and its three segmentation results obtained with the

Figure 12. Examples of test images extracted from the ©Corel database. (1) original images, (2) corresponding ground truth-brightness gradient/color gradient/texture gradient segmentation method, (3) under-segmented, (4) normal, (5) over-segmented

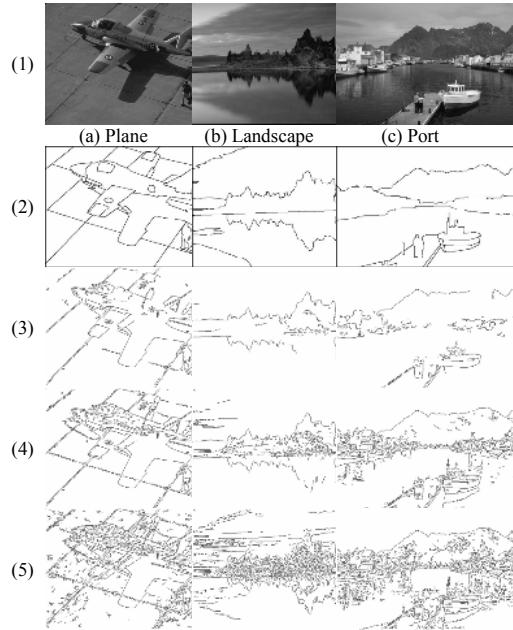


Table 1. Global classification rate of the best segmentation results for each criterion in the different situations (under, normal or over-segmentation) for 100 images extracted from the ©Corel database and nine segmentation methods

Criterion	Under segmented	Normal	Over segmented
Quality	7	20	73
FOM	0	99	1
Hausdorff	7	25	68
Odet ODI	49	36	15
Odet UDI	0	10	90

brightness/color/texture gradients segmentation method are presented. We used three values of the parameters of this method to obtain three types of segmentation results. The “normal” segmentation result is obtained by using the parameters of this method that maximize the correct classification rate between pixels belonging in a contour in the detection result and in the ground truth.

In this study, we computed the following evaluation criteria: Ramon’s multi-features quality measurement (quality), Pratt’s figure of merit (FOM), Hausdorff’s

distance and Odet's criteria (ODI and UDI) on the real and ground truth segmentation results presented in the experimental protocol. Table 1 presents the percentage of each segmentation type defined as the best one for each criterion. A good evaluation criterion must prefer the normal segmentation result.

According to the way we segmented the images, the normal segmentation result is always supposed to be best. A good evaluation criterion must have the highest classification rate for the normal result. As we can see in Table 1, the Pratt's figure of merit considers, with a classification rate equal to 99%, the normal segmentation result as the best one. On the contrary, quality and Hausdorff's distance privilege the over-segmented result. As expected, the ODI criterion measuring the over segmentation prefers under-segmented images (49%). UDI criterion measuring the under segmentation, prefers the over-segmented ones (90%).

These results illustrate the behaviors of the tested criteria on a natural image database using a ground truth made by experts. In the supervised case, one criterion stands out in our comparative study: Pratt's figure of merit. However, we can also figure out the two Odet criteria that are able to evaluate the over- and under-segmentation.

Figure 13. Examples of test images extracted from the synthetic image database: (a) uniform; (b) mixed; (c) textured; (1) original images; (2)(3)(4) FCM results with different parameters; (5) EDISON results

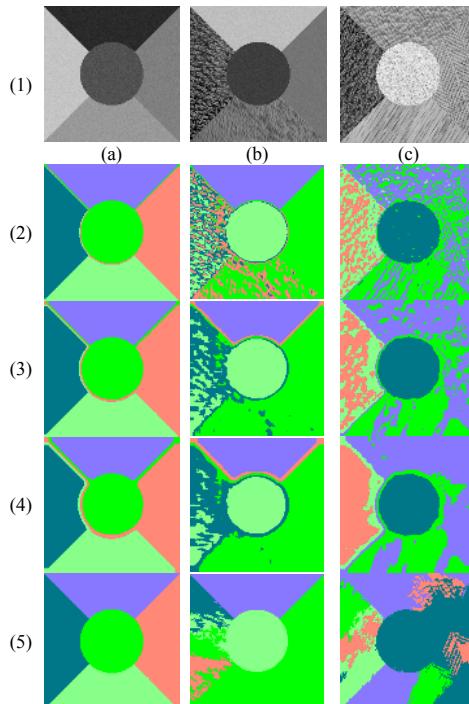


Table 2. Correlation factors between the different evaluation criteria and the Vinet's measure for the three synthetic image subsets

	Uniform	Mixed	Textured
Borsotti	-0.845	-0.264	-0.207
Zéboudj	0.909	-0.003	-0.142
Inter-Region	0.200	0.227	0.181
Intra-Region	0.158	0.030	0.256
Intra-Inter	0.226	0.096	0.010
Rosenberger	0.277	0.143	0.355
Attributes	-0.1317	-0.2329	-0.0215

Unsupervised Evaluation

Figure 13 presents three images from our synthetic image database and for each one its four segmentation results.

In this comparative study, we selected the following unsupervised evaluation criteria: Borsotti, Zéboudj, Rosenberger, Levine and Nazif intra-inter-intra/inter attributes dispersion. We use the Vinet's measure (correct classification rate) as reference because it is generally used to illustrate the quality of a region segmentation result in the literature. We compared the evolution of these criteria to the evolution of the Vinet's measure by computing the correlation factors between them. A correlation factor is a value between -1 and 1. The absolute value of this correlation factor gives us information about the similarity of the evolution between two criteria: the higher the absolute value is, the more similar the criteria are.

Table 2 presents the correlation factors between the results of each criterion and the result of the Vinet's measure for each subset of synthetic images. As we can see in this table, the criteria that seem best fitted to the uniform case are the Borsotti and Zéboudj ones. The one that grants the best results in the textured case is the Rosenberger. These conclusions on the efficiency of the selected evaluation criteria are based on their general behaviors compared to the Vinet's measure.

Applications

Segmentation evaluation criteria can be used for many applications. We illustrate four of them in this section. The first one concerns the objective comparison of segmentation results. The second application deals with one of the most difficult problems in image segmentation, namely, the choice of input parameters to process a given image. Another use is the possibility to define new segmentation algorithms by optimizing an evaluation criterion. Finally, a quantitative evaluation criterion can be used as a confidence measure for the fusion of different segmentation results of a single image or the different bands of a multi-components image.

Comparison of Segmentation Results

The first application is the comparison of different segmentation results. Suppose we have some segmentation results of a single image with the same level of precision obtained by different methods (the level of precision of a segmentation result is generally

Figure 14. Comparison of three segmentation results: (a) Original image; (b) PCM result; (c) FCM result; (d) EDISON result

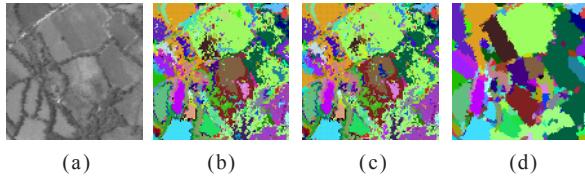


Table 3. Values of some unsupervised evaluation criteria of the segmentation results in figure 14 (values in boldface correspond to the best ones)

	FCM	PCM	EDISON
Borsotti	0.0222	0.0297	0.0155
Zeboudj	0.6228	0.6124	0.5428
Inter	0.0901	0.0889	0.1099
Intra	0.7258	0.7112	0.9693
Intra-Inter	0.5202	0.5239	0.5275
Rosenberger	0.6379	0.6328	0.6973

defined by the number of regions or classes). The previous evaluation criteria can be used to choose the “best” segmentation result. In order to illustrate the behaviors of the evaluation criteria, we present some results obtained from an aerial image. Figure 14 shows three segmentation results of a single image with the same level of precision (same number of classes). The three segmentation methods we used are Fuzzy-C Means (FCM) (Krihnapuram & Keller, 1993), Probabilistic Classification Method (PCM) (Krihnapuram & Keller, 1996) and EDISON (Comanicu & Meer, 2002). We used the new coloring post-processing presented in Section 1. This post-processing consists of affecting the same color for a region in the different segmentation results. The color representation of each segmentation result makes their comparison easier.

Table 3 gathers the value of some unsupervised evaluation criteria computed on these three segmentation results. Most of evaluation criteria choose the EDISON segmentation result. This choice is, as for us, visually correct. This enables us to compare the behavior of different segmentation methods in order to choose the most appropriate for a given application.

Parameters Fitting for Segmentation

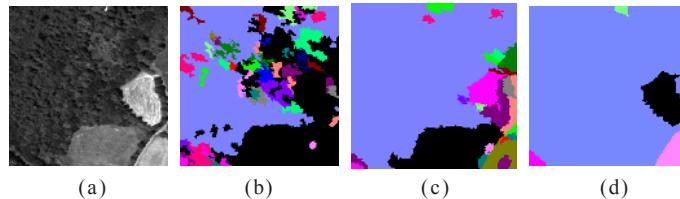
The second application is to facilitate the choice of the parameters of a segmentation method. Image segmentation generally needs the definition of some input parameters that are usually defined by the user. This task, sometimes arbitrary, can be

Table 4. (a) Number of regions for the segmentation results for each value of spatial bandwidth (column) and range bandwidth (row) by using the EDISON algorithm, (b) Evolution of the Zéboudj's contrast

	3	5	7	9	11	13	15	17		3	5	7	9	11	13	15	17	
5	82	84	82	80	78	79	73	75		2	0.368	0.346	0.338	0.332	0.313	0.314	0.309	0.317
5	70	59	58	48	36	39	31	33		5	0.310	0.295	0.305	0.239	0.289	0.228	0.201	0.303
8	30	31	24	24	18	12	13	9		8	0.296	0.285	0.357	0.300	0.217	0.309	0.242	0.316
11	13	13	15	14	12	8	5	6		11	0.295	0.332	0.265	0.211	0.213	0.284	0.249	0.266
14	5	10	8	9	7	5	6	4		14	0.362	0.421	0.342	0.367	0.371	0.244	0.221	0.231
17	2	6	8	7	5	3	4	5		17	0.141	0.406	0.358	0.362	0.353	0.323	0.340	0.179
20	3	5	7	6	5	3	3	3		20	0.014	0.317	0.290	0.336	0.288	0.293	0.309	0.308
23	1	2	4	5	3	3	2	3		23	0.000	0.000	0.284	0.223	0.268	0.289	0.211	0.108
26	1	1	1	1	3	4	3	2		26	0.000	0.000	0.000	0.000	0.009	0.079	0.089	0.102
29	1	1	1	1	1	2	1	4		29	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.017
31	1	1	1	1	1	2	2	1		31	0.000	0.000	0.000	0.000	0.000	0.008	0.007	0.000

(a)

(b)



(a)

(b)

(c)

(d)

automated via determining the best parameters by considering the value of an evaluation criterion. We carried out a study to facilitate the choice of two parameters of the EDISON segmentation method (spatial bandwidth and range bandwidth). Segmentation results with different values of the number of regions are obtained: precise (60-90 regions), intermediate (20-40 regions) and coarse (5-10 regions). Table 4 gives the evolution of the Zéboudj's criterion for eight values of the spatial bandwidth and 11 values of the range bandwidth. The color of the elements in Table 4 corresponds to segmentation results of the same type: black for precise results, gray for intermediate and light gray for coarse results. We show in Figure 15 the best segmentation results for each type.

Segmentation by Optimization

Another application is the possibility to define new segmentation methods based on the optimization of an evaluation criterion (Rosenberger & Chehdi, 2000, 2002). A genetic algorithm can be used as optimization method.

Genetic algorithms determine solutions of functions by simulating the evolution of a population until survival of only best-fitted individuals. Survivors are individuals obtained by crossing-over, mutation and selection of individuals from the previous generation. A genetic algorithm is defined by considering five essential data:

- **Genotype:** A set of characteristics of an individual such as its size. The segmentation result of an image I is considered as an individual described by the label of each pixel;
- **Initial population:** A set of individuals characterized by their genotypes;
- **Fitness function:** This function quantifies the fitness of an individual to the environment by considering its genotype. An evaluation criterion which enables the quantify of a segmentation result is used;
- **Operators on genotypes:** These define alterations on genotypes in order to evaluate the population during generations. Three types of operators exist:
 1. **Individual mutation:** Genes of an individual are modified in order to better adapt to the environment. As a segmentation result is defined (in our case) by the label of each pixel, the mutation operator consists in changing the label of few pixels in the segmentation result;
 2. **Selection of individual:** Individuals that are not adapted to the environment do not outlive to the next generation; and
 3. **Crossing-over:** Two individuals can reproduce by combining their genes.

For the crossing-over, an area of a segmentation result is replaced by an area from another segmentation result.

- **Stopping criterion:** This criterion allows the evolution of the population to be stopped. We choose to consider the stability of the standard deviation of the evaluation criterion of the population.

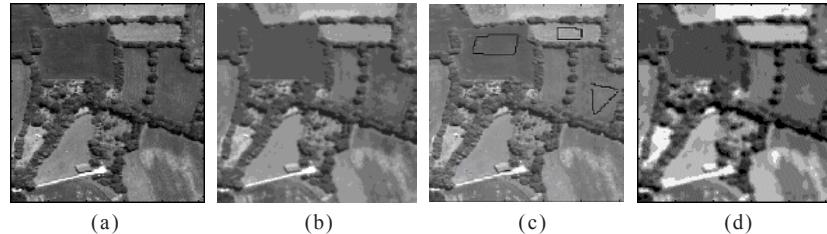
The execution of the genetic algorithm is realized in four steps:

1. Definition of the initial population and computation of the fitness function of each individual;
2. Selection and crossing-over of individuals of the population;
3. Evaluation of individuals in the population; and
4. Back to step 2 if the stopping criterion is not satisfied.

We illustrate segmentation results in Figure 16 results by the previous approach. Figure 16b is the obtained segmentation result by using the Rosenberger's criterion (unsupervised case).

In order to define the level of precision of the segmentation result, it is possible to additionally use a local ground truth. A local ground truth is defined as a small set of pixels with known class. It is used in the optimization process by computing the good classification rate (Vinet's measure) for each cluster of the local ground truth. The higher this value is, the more the result corresponds to the needed level of precision. Finally, the evaluation criterion is defined by the sum of an unsupervised and a supervised one. An example of a local ground truth is given in the Figure 16c. Figure 16d is the supervised segmentation result.

Figure 16. Segmentation results by optimization: (a) original image; (b) unsupervised result; (c) definition of the level of precision; (d) supervised result

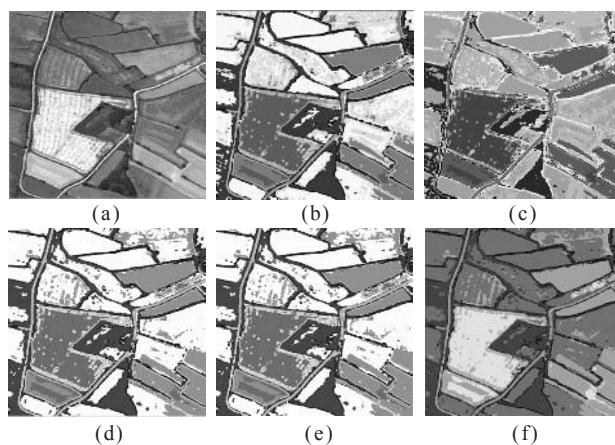


copyright 2005 IEEE Rosenberger and Chehdi (2002)

Table 5. Some statistics on the segmentation results before and after fusion by considering the value of the Rosenberger's criterion

Initial population	
Average value	0.903
Highest value	0.913
Lowest value	0.881
Standard deviation	0.008
Final population	
Average value	0.979
Highest value	0.979
Lowest value	0.979
Standard deviation	6.02e-08

Figure 17. Fusion of four segmentation results: (a) original image; (b)-(e) four segmentation results (5,10,12 and 15 classes); (f) fusion result (8 classes)



copyright 2005 IEEE Rosenberger and Chehdi (2000)

Fusion of Segmentation Results

An evaluation criterion can be used to fuse several segmentation results of a single image or of the different bands in the multi-components case. It is possible to use the same processing as described previously with two major differences. The first one is the initial population. In this case, it is composed of the different segmentation results to fuse. The segmentation results can be obtained by different segmentation methods or with the same one with different input parameters. Second, the mutation operator is not used in order to only exploit information present in the initial segmentation results.

Figure 17 shows an example of fusion of four segmentation results of a single image. Segmentation results were obtained by using the Fuzzy-C means algorithm with different input parameters (number of classes, windows size, etc.). We also used the Rosenberger's criterion. Figure 17f is the fusion result with eight classes. We obtain a good compromise for the number of classes to provide a good separation of fields while preserving some internal variations. Table 5 presents the value of the evaluation criterion for the different segmentation results before and after fusion. As we can see, the fusion process allows us to have a segmentation result with a quality of 0.979 that is much better than the best initial segmentation result of quality 0.913. The standard deviation of the fitness function in the final population is close to zero, so we obtain the optimal result.

CONCLUSIONS

Segmentation evaluation is a great challenge in image analysis. Some evaluation criteria were presented from the state-of-the-art to quantify the quality of a segmentation result. Supervised criteria need some *a priori* knowledge, such as a ground truth, to evaluate a segmentation result. Many metrics have been proposed in the literature in order to compute the similarity of the segmentation result with the ground truth. In order to compare these criteria, we used some natural images with ground truths manually made from Martin, Fowlkes, Tal, and Malik (2001). These images were segmented by a contour detection method with different values of one parameter. Each evaluation criterion was used to compare these results. A good evaluation criterion must choose the most similar segmentation result to the ground truth as the best one. The comparative study showed that the Pratt's figure of merit gave good results. The proposed methodology for the comparison of evaluation criteria has the advantage to be objective, automatic and relevant for natural images. Its main problem is the creation of these ground truths that necessitate lots of experts to guarantee the confidence of the result.

Unsupervised evaluation criteria can be used when no *a priori* knowledge is available. They generally compute some statistics on the segmentation result according to the original image. The comparison of unsupervised evaluation criteria must be realized in a supervised context. We used ground truth from synthetic images containing uniform and textured regions. Vinet's measure (correct classification rate) is used as a reference. A good unsupervised evaluation criterion is the one that has a similar behavior to the Vinet's measure without any *apriori* knowledge. In the unsupervised case, criteria that reveal themselves as reliable are Zéboudj's contrast, Borsotti's criteria for low textured and uniform images and the Rosenberger's criterion otherwise. The proposed methodology for the comparison of these evaluation criteria provides a significant and

objective result on a large database of segmentation results. That is why we worked on synthetic images, even if they do not represent all the complexity of natural images. As we compare the general behavior of these evaluation criteria with a known reference, we need a huge number of segmentation results. This large quantity of data could be difficult to obtain with experts.

Many promising applications are concerned with these evaluation criteria. They can provide an objective comparison of segmentation methods by evaluating some segmentation results with the same level of precision. Segmentation methods based on the optimization of an evaluation criterion can be defined. These evaluation criteria can assist an user in defining the optimal input parameters of a segmentation method for a given image.

Research perspectives on supervised evaluation criteria concern the use of information on objects to recognize. In this case, a segmentation result will be correct if it allows the recognition of known objects. This approach is very close to the interpretation for a given application. Concerning unsupervised criteria, future works will have to adapt the computation of statistics in the segmentation result in order to better take into account the type of regions. Another approach is based on McCane's observation (McCane, 1997). He showed that it is necessary to use the maximum criteria and to combine them. This approach consists in fusing different evaluation criteria in order to improve their performance.

REFERENCES

- Baddeley, A.J. (1992). An error metric for ebinary images. *Robust Computer Vision*, 59-78.
- Bartels, K.A., & Fisher, J.L. (1995). Multifrequency eddy current image processing techniques for nondestructive evaluation. In *Proceedings of the International Conference on Image Processing (ICIP)* (pp. 486-489).
- Beauchemin, M., Thomson, K.P.B., & Edwards, G. (1998). On the Hausdorff distance used for the evaluation of segmentation results. *Canadian Journal of Remote Sensing (CJRS)*, 24(1), 3-8.
- Borsotti, M., Campadelli, P., & Schettini, R. (1998). Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19, 741-747.
- Bowyer, K. (2001). Edge detector evaluation using empirical ROC curves. *Journal on Computer Vision and Understanding*, 84, 77-103.
- Chabrier, S., Laurent, H., Emile, B., Rosenberger, C., & Marché, P. (2004). A comparative study of supervised evaluation criteria for image segmentation. In *Proceedings of the International Conference European Signal Processing Conference* (pp. 1143-1146).
- Chabrier, S., Rosenberger, C., Laurent, H., Emile, B., & Marché, P. (2004). Evaluating the segmentation result of a gray-level image. In *Proceedings of the International Conference European Signal Processing Conference* (pp. 953-956).
- Coquin, D., Bolon, P., & Chehadeh, Y. (1997). Quantitative evaluation of filtered images. In *Proceedings of the Conference GRETSI*, 2 (pp. 1351-1354).
- Cocquerez, J-P., & Devars, J. (1985). Contour detection in aerial images: New operators. *Traitemet du signal*, 2, 45-65.

- Correia, P., & Pereira, F. (2000). Objective evaluation of relative segmentation quality. In *Proceedings of the International Conference on Image Processing (ICIP)*, 2 (pp. 308-311).
- Comanicu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24 , 603-619.
- Demigny, D., & Kamlé, T. (1997). A discrete expression of Canny's criteria for step edge detector performances evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11), 1199-1211.
- Everingham, M.R., Muller, H., & Thomas, B.T. (2001). Evaluating image segmentation algorithms using monotonic hulls in fitness/cost space. In *Proceedings of the British Machine Vision Conference (BMVC)* (pp. 363-372).
- Geman & Graffigne. (1987). Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, Berkeley (pp. 1496-1517).
- Han, J.H., & Kim, T.Y. (2002). Ambiguity distance: An edge evaluation measure using fuzziness of edges. *Fuzzy Sets and Systems*, 126, 311-324.
- Huet, F., & Philipp, S. (1998). Fusion of images interpreted by a new fuzzy classifier. *Pattern Analysis and Applications*, 1, 231-247.
- Jain, A. K., Duin, R.P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 4-37.
- Kara-Falah, R. (1995). *Image segmentation: Cooperation, fusion, evaluation*. PhD thesis, University of Savoie.
- Kitchen, L., & Rosenfeld, A. (1984). Scene analysis using region-based constraint filtering. *Pattern Recognition*, 17(2), 189-203.
- Krihnapuram, R., & Keller, J.M. (1993). Possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 30, 98-110.
- Krihnapuram, R., & Keller, J.M. (1996). The possibilistic c-means algorithm: Insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4, 385-393.
- Levine, M.D., & Nazif, A.M. (1985). Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 7, 155-164.
- Liu, J., & Yang, Y.-H. (1994). Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(7), 689-700.
- Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference Computer Vision* (pp. 416-423).
- McCane, B. (1997). On the evaluation of image segmentation algorithms. In *Proceedings of Digital Image Computing: Techniques and Applications*, Massey University, Albany Campus, Auckland, New Zealand (pp. 455-460).
- Michel, O., Baraniuk, R.G., & Flandrin, P. (1994). Time-frequency based distance and divergence measures. In *Proceedings of the IEEE International Symposium on Time-Frequency and Time-Scale Analysis* (pp. 64-67).
- Nazif, A.M., & Levine, M.D. (1984). Low level image segmentation: An expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 55-577.

- Odet, C., Belaroussi, B., & Benoit-Cattin, H. (2002). Scalable discrepancy measures for segmentation evaluation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (pp. 785-788).
- Pal, N.R., & Pal, S.K. (1989). Entropic thresholding. *Signal Processing*, 16, 97-108.
- Peli, T., & Malah, D. (1982). A study of edge detection algorithms. *Computer Graphics and Image Processing*, (20), 1-21.
- Pratt, W., Faugeras, O.D., & Gagalowicz, A. (1978). Visual discrimination of stochastic texture fields. *IEEE Transactions on Systems, Man, and Cybernetics*, 8, 796-804.
- Roman-Roldan, R., Gomez-Lopera, J.F., Atae-allah, C., Martinez-Aroza, J., & Luque-Escamilla, P.L. (2001). A measure of quality for evaluating methods of segmentation and edge detection. *Pattern Recognition*, 34, 969-980.
- Rosenberger, C. (1999). *Implementation of an adaptive image segmentation system*. PhD thesis, University of Rennes.
- Rosenberger, C., & Chehdi, K. (2000). Genetic fusion: Application to multi-components image segmentation. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 4, Istanbul (pp. 2219-2222).
- Rosenberger, C., & Chehdi, K. (2002, October 6-9). Supervised genetic image segmentation. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics (CDROM Proceedings)*, Hammamet, Tunisia (Vol. 5).
- Sahoo, P.K., Soltani, S., Wong, A.K.C., & Chen, Y.C. (1988). A survey of thresholding techniques. *Graphical Model and Image Processing (CVGIP)*, 41 (pp. 233-260).
- Spinu, C. (1997). *A multi-agents approach for image segmentation combining estimation and evaluation*. PhD thesis, Laboratory TIMC.
- Tan, H.L., Gelfand, S.B., & Delp. (1992). A cost minimization approach to edge detection using simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14, 3-18.
- Vinet, L. (1991). *Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques*. PhD thesis, University of Paris IX Dauphine.
- Weszka, J.S., & Rosenfeld, A. (1978). Threshold evaluation techniques. *IEEE Transactions on Systems, Man and Cybernetics (SMC)*, 8, 622-629.
- Wilson, D.L., Baddeley, A.J., & Owens, R.A. (1997). A new metric for gray-scale image comparison. *International Journal Computer Vision*, 24, 5-17.
- Zamperoni, P., & Starovoitov, V. (1996). On measures of dissimilarity between arbitrary gray-scale images. *International Journal of Shape Modeling*, 2 (pp. 189-213).
- Zéboudj, R. (1988). *Filtering, automatic threshold, contrast and contours: From pre-processing to image analysis*. PhD thesis, University of Saint Etienne.
- Zhang, Y.J. (1996). A survey on evaluation methods for image segmentation. In *Proceedings of the International Conference Computer Vision and Pattern Recognition (CVPR)*, 29(8), 1335-1346.

Chapter XIX

Objective Evaluation of Video Segmentation Quality

Paulo Lobato Correia, Technical University of Lisbon, Portugal

Fernando Pereira, Technical University of Lisbon, Portugal

ABSTRACT

The evaluation of image and video segmentation results assumes a critical role for the selection of appropriate segmentation algorithms, as well as the adjustment of their parameters for optimal segmentation performance in the context of a given application. The current practice for the evaluation of video segmentation quality is based on subjective testing, which is an expensive and time-consuming process. Objective segmentation quality evaluation techniques can alternatively be used, once appropriate algorithms become available. Currently this is a field under development and this contribution proposes evaluation methodologies and objective segmentation quality metrics both for individual objects and for complete segmentation partitions. Standalone and relative evaluation metrics are proposed, to be used when a reference segmentation is missing, or available for comparison, respectively.

INTRODUCTION

The major objective of image and video segmentation algorithms is to produce appropriate results for the particular goals of the application addressed. Therefore, segmentation algorithms' performance assessment assumes a crucial importance in evaluating the degree to which application targets are met.

The current practice for segmentation performance evaluation consists in subjective assessment by a representative group of human viewers (COST211, 2001). To be meaningful, such evaluations must follow precise methodologies, both in terms of test environment setup and grading techniques. Standard subjective evaluation methodologies have been established for video quality evaluation (ITU-R BT.500, 1995; ITU-T P.910, 1996; ITU-T P.930, 1996), targeting the evaluation of video degradation due to some types of processing, such as video coding, but not for segmentation quality evaluation.

Subjective evaluation of segmentation quality is usually considered as a “good enough” procedure, and often it is the only solution available. However, this type of evaluation has several drawbacks. Setting up the evaluation environment requires a considerable effort, besides requiring the presence of a significant number of evaluators to achieve statistically relevant results, becoming a time-consuming and expensive process. Also, the subjective nature of the evaluation may prevent the reproducibility of results, especially if precise and stable evaluation conditions and methodologies are not respected.

The alternative is to devise segmentation evaluation strategies that do not involve human evaluators. But, even if the development of segmentation algorithms is the topic of a large number of publications, the issue of their performance evaluation has not received comparable attention (Zhang & Gerbrands, 1994; Zhang, 1996; Rees & Grenway, 1999). This may be due to the difficulty in establishing a measure capable of adequately evaluating segmentation quality, except for very well constrained application scenarios.

Several non-subjective quality evaluation methodologies have been proposed since the 1970s, initially targeting the performance assessment of edge detectors. More recently, with the emergence of the MPEG-4 standard (ISO MPEG-4, 1999) and its ability to independently code arbitrarily shaped video objects, a new impetus to research on segmentation, as well as the development of segmentation quality evaluation methodologies, has been apparent (Erdem, Tekalp, & Sankur, 2001; Correia & Pereira, 2002; Mech & Marques, 2002; Correia & Pereira, 2003; Villegas & Marichal, 2004).

Some initial attempts tried to define analytical evaluation methods, assessing segmentation performance directly by examining the algorithms, but not requiring their implementation. The principles and properties of the algorithms’ components (e.g., complexity, efficiency, resolution of the results and processing approach) and their combination strategies were analyzed. These methods’ applicability to complex algorithms is limited, while requiring considerable human evaluation effort. Qualitative evaluation reports, focusing on the strengths and weaknesses of algorithms, or quantitative measures for particular techniques (Abdou & Pratt, 1979), could be produced as output.

Nowadays, several objective evaluation methods have been proposed to assess segmentation quality by using automatic measurement tools. Typically, objective evaluation operates on segmentation results produced by the algorithm being tested. As for subjective evaluation, two classes of techniques exist (Zhang, 1996):

- **Standalone objective evaluation or “goodness” evaluation:** Consists in evaluating a segmentation partition, understood as the set of objects that completely covers an image at a given instant without overlapping, on its own.

- **Relative objective evaluation or “discrepancy” evaluation:** Consists in evaluation by comparison against reference segmentation, playing the role of “ground truth.”

Objective segmentation evaluation outputs can be either qualitative—for example, good, acceptable or unacceptable, or quantitative—producing normalized evaluation scores to ease comparisons.

The ideal situation would be to have good objective segmentation quality evaluation tools that would automatically match the results of subjective evaluation—still the most reliable evaluation approach. The availability of objective evaluation tools would greatly ease the process of image and video segmentation technology evaluation, which is nowadays recognized as a major unsolved problem. Since no well-established methods for this purpose are currently available, this chapter proposes a solution for objective image and video segmentation quality evaluation.

SEGMENTATION QUALITY EVALUATION OVERVIEW: CURRENT PRACTICE AND OBJECTIVE ASSESSMENT

The development of new objective video segmentation quality evaluation methodologies and metrics not only builds upon information available from the current practice of subjective evaluation procedures and previously published objective segmentation evaluation material, but also upon the characteristics of the human visual system (HVS), and the vast experience on video quality evaluation—these aspects are addressed in this section.

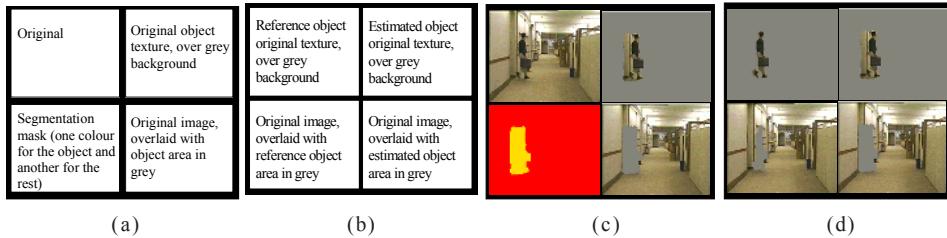
Subjective Video Segmentation Quality Evaluation

Subjective segmentation quality evaluation, either standalone or relative, is the most common evaluation procedure.

The main steps usually involved in subjective standalone evaluation are:

1. Selection of test sequences representative of the target segmentation scenario.
2. Selection of the relevant objects for each test sequence to assist the human evaluation procedure.
3. Application of the segmentation algorithm under test to the selected sequences.
4. Visual evaluation: Each object’s results may be viewed separately, preceded by a title screen or audio introduction explaining what to evaluate. Figure 1a and 1c show a typical display configuration, including original image, segmentation partition, portion of original image corresponding to the selected object over a neutral background and the remaining of original image with the object area replaced by a neutral color.
5. Segmentation quality scoring: integrating results of all temporal instants into a final score for each object, and the complete partition. A discrete scoring scale, between 1 (lowest quality) and 5 (highest quality), is typically used, from which a mean opinion score (MOS) can be computed.

Figure 1. Display layout for subjective segmentation quality evaluation: standalone (a), relative (b), and corresponding examples, in (c) and (d), respectively, for the main object of the sequence “hall monitor”



When a segmentation reference, or “ground truth,” is available for comparison, subjective relative evaluation can be used. The reference can be obtained automatically, for instance using chroma-keying techniques, manual segmentation or combining automatic and manual segmentation. Eventually, the output of an automatic segmentation algorithm providing sufficiently good results may be taken as reference.

Subjective relative evaluation differs from standalone evaluation by not requiring the selection of target objects; the visual evaluation can be improved by simultaneously displaying the estimated and reference segmentations—see example in Figure 1b and 1d.

To maximize reliability of evaluation results, it is advantageous to use, whenever applicable, the standard test methodologies developed for image and video quality evaluation, namely for viewing conditions, test material and evaluators selection and scoring methods.

Available Metrics for Standalone Objective Segmentation Quality Evaluation

Standalone objective evaluation should exploit any *a priori* information available about the target application. Since not much information is usually available, standalone segmentation quality typically has to be assessed based on the homogeneity of the identified objects and their disparity to neighbors (Levine & Nazif, 1985; Zhang, 1996).

The most relevant standalone evaluation metrics found in the literature can be classified as:

- **Intra-object homogeneity:** In some application scenarios, it makes sense to expect the interior of objects to have reasonably homogeneous texture. This homogeneity can be evaluated by measures such as the variance of texture (Weszka & Rosenfeld, 1978). A generic intra-object homogeneity metric is the variance of a selected feature f within an object O , weighted by the object’s area A (Levine & Nazif, 1985):

$$\sigma_j^2 = \sum_{i \in O_j} \frac{(f_i - \bar{f}_j)^2}{A_j} \quad (1)$$

- **Inter-object disparity:** Often it is expected that neighboring objects present significant disparity along their borders. A generic contrast, or disparity, metric is based on the difference of a given feature value f computed for two neighboring objects i and j (Levine & Nazif, 1985):

$$c_{ij} = c_{ji} = \frac{|\bar{f}_i - \bar{f}_j|}{\bar{f}_i + \bar{f}_j} \quad (2)$$

In standalone objective evaluations, the precise segmentation targets are not known. Therefore, it is important to gather as much *a priori* information as possible about the segmentation goals so that appropriate evaluation metrics can be selected (or developed). For instance, in a car license plate identification application, the area where the target object is located, the background color and the typical size, are *a priori* information that can improve the performance of the segmentation and of its evaluation.

Available Metrics for Relative Objective Segmentation Quality Evaluation

Relative objective segmentation quality evaluation exploits the availability of a reference segmentation to achieve a more accurate assessment. When the estimated segmentation is not perfect, the number and position of misclassified pixels provide valuable evaluation information about the shape fidelity and thus, the segmentation quality. Additionally, the observed differences to the reference in selected object features provide useful information when shape fidelity is not perfect. Also, the correctness in terms of the number of segmented objects should be used for evaluation purposes and the temporal dimension needs to be considered when dealing with video.

The most relevant metrics for relative objective segmentation quality evaluation found in the literature (Fram & Deutsch, 1975; Yasnoff, Mui, & Bacus, 1977; Weszka & Rosenfeld, 1978; Abdou & Pratt, 1979; Bryant & Bouldin, 1979; Levine & Nazif, 1985; Strasters & Gerbrands, 1991; Zhang & Gerbrands, 1994; Zhang, 1996; Villegas, Marichal, & Salcedo, 1999; Mech & Marques, 2002) are of two major classes:

- **Spatial accuracy:** Including metrics like:
 - **Shape fidelity:** A good segmentation must always present good shape fidelity. It can be evaluated by the percentage of misclassified pixels, weighted by their distance to the reference. This distance is essential as the same number of mismatched pixels may lead to very different shape errors. One such metric, for partitions with two objects, called spatial quality metric (SQM) (Villegas et al., 1999), is:

$$SQM = \sum_{d=1}^{D_{FG_{max}}} w_{MF}(d) \cdot Card(R_d \cap E^C) + \sum_{d=1}^{D_{BG_{max}}} w_{AB}(d) \cdot Card(R_d^C \cap E) \quad (3)$$

where E and R are the estimated and reference segmentations and MF and AB are the missing foreground and added background pixels. $Card(S)$ is the

cardinality of set S , and S^C its complementary set. D_{FG} stands for the distance between the incorrectly estimated foreground points and the closest reference foreground point and D_{BG} is similarly defined for the background. The sets R_i and R_i^c are defined as:

$$R_i = \{x \in R, d(x, R^C) = i\}, \quad R_i^c = \{x \in R^C, d(x, R) = i\} \quad (4)$$

with $d(x, S)$ being the distance from point x to the set S . Finally, the w_{MF} and w_{AB} weights are given by (Villegas et al., 1999):

$$w_{MF}(d) = 20.588 - \frac{132.429}{d + 6.6}, \quad w_{AB}(d) = 2 \cdot d \quad (5)$$

- **Spatial features similarity:** Selected object spatial features can be compared to evaluate segmentation quality. An example is the relative ultimate measurement accuracy ($RUMA_f$) metric defined in Zhang and Gerbrands (1994):

$$RUMA_f = \frac{|f_R - f_E|}{f_R} \quad (6)$$

where f_R and f_E are the feature values for the reference and estimated segmentations. Geometric features like area, perimeter or circularity can be considered for computing this metric.

- **Number of objects comparison:** A metric for comparing the number of objects in the reference R_N and estimated E_N segmentations, called fragmentation ($FRAG$), has been proposed in Strasters and Gerbrands (1991):

$$FRAG = \frac{1}{1 + p|R_N - E_N|^q} \quad (7)$$

where p and q are scaling parameters.

- **Temporal accuracy:** Evaluation using the temporal dimension has received less attention in the literature, the main metric considered being:
 - **Temporal stability:** A metric called “activity” compares the object area variations between the estimated and reference segmentations along time, A_t , in pixels (Wollborn & Mech, 1998):

$$Activity = \frac{1}{N} \cdot \sum_{t=1}^N \frac{|A_{t-1} - A_t|}{A_{t-1}} \quad (8)$$

Human Visual System Characteristics

Human visual system (HVS) characteristics should also be taken into consideration during segmentation quality evaluation. For instance, HVS behavior is affected by

factors like (Lambrecht, 1996; Osberger, Bergman & Maeder, 1998; Tan, Ghanbari, & Pearson, 1998; Hekstra et al., 2002):

- **Motion activity:** In the presence of moving objects, still objects and spatial edges become less important, and objects exhibiting distinct motion properties usually get more attention.
- **Masking:** Perception of the various objects is affected by each other's presence and noise (Hamada, Miyaji, & Matsumoto, 1997).
- **Channel sensitivity:** The HVS luminance channel is more sensitive than the chrominance channels.
- **Saturation effect:** Viewers' ability to perceive further changes in image quality after it exceeds certain thresholds, either towards worst or better quality, is limited.
- **Asymmetric tracking:** Generally, humans tend to remember unpleasant experiences. Thus, it is often preferable to have a stable segmentation rather than a more precise one which presents significant quality problems from time to time.

The above HVS characteristics indicate that both spatial and temporal features of objects should be taken into account for segmentation quality evaluation. For instance, when an object exhibits significant motion activity, its motion fidelity assumes a more significant role than the spatial accuracy aspect.

Also, in a segmentation partition, objects are not always equally important and the more relevant ones should be segmented more precisely. Factors influencing human visual attention should be taken into account to determine an object's relevance. Examples are: distinct motion characteristics, position (centre of image), contrast, size, shape, orientation, colour and brightness. Also, higher level factors should be taken into account in most applications, like the presence of people, faces and text.

Useful Metrics from the Video Quality Evaluation Field

The methodologies and metrics developed in the area of video quality assessment over many years also provide important background information for segmentation quality evaluation, as some of the effects to be measured are similar. For instance, those metrics computing differences in an object's spatial, temporal and spatiotemporal characteristics may be adapted for relative video segmentation quality evaluation, providing valuable segmentation quality hints.

From the video quality metrics available in the literature, three have been judged especially relevant for the development of segmentation quality evaluation metrics:

- **Spatial perceptual information (SI):** SI provides a measure of the spatial detail in an image, taking higher values for more complex scenes. Specified in ITU-T Recommendation P.910 (ITU-T P.910, 1996), based on the Sobel edge detector, SI can be computed for each object k at instant t :

$$SI(k_t) = \sqrt{\frac{1}{N} \cdot \sum_i \sum_j (Sobel(k_t))^2 - \left(\frac{1}{N} \cdot \sum_i \sum_j (Sobel(k_t)) \right)^2} \quad (9)$$

- **Temporal perceptual information (TI):** TI provides a measure of the temporal changes in a video sequence, taking higher values for high motion sequences (ITU-T P.910, 1996). It can be computed for each object k at instant t :

$$TI(k_t) = \sqrt{\frac{1}{N} \cdot \sum_i \sum_j (k_t - k_{t-1})^2 - \left(\frac{1}{N} \cdot \sum_i \sum_j (k_t - k_{t-1}) \right)^2} \quad (10)$$

- **Criticality:** This metric builds on the SI and TI metrics, providing a spatiotemporal evaluation of the complexity of a sequence (Fenimore, Libert, & Wolf, 1998):

$$crit(k) = 4.68 - 0.54 \cdot p_1(k) - 0.46 \cdot p_2(k) \quad (11)$$

with

$$p_1(k) = \log_{10}(mean_{time}(SI_{rms}(k_t) \cdot TI_{rms}(k_t))) \quad (12)$$

$$p_2(k) = \log_{10}(\max_{time}(|abs(SI_{rms}(k_t) - SI_{rms}(k_{t-1}))|)) \quad (13)$$

$$SI_{rms}(k_t) = \sqrt{\frac{1}{N} \cdot \sum_i \sum_j (Sobel(k_t))^2} \quad (14)$$

$$TI_{rms}(k_t) = \sqrt{\frac{1}{N} \cdot \sum_i \sum_j (k_t - k_{t-1})^2} \quad (15)$$

The concepts, ideas and metrics discussed in this section can be used for segmentation quality evaluation purposes.

VIDEO SEGMENTATION QUALITY EVALUATION PROPOSALS

The major goal of objective segmentation quality evaluation systems is being able to mimic, using automatic procedures, the results that formal subjective evaluations would produce. Objective evaluation results can then be used to compare the performance of different segmentation algorithms, or parameter configurations, in view of a given set of application requirements. Alternatively, evaluation results can be used to optimize the segmentation algorithm's performance by adjustment of its parameters' configuration, using feedback mechanisms.

Given that no formal procedures for subjective segmentation quality evaluation have been defined until now, no mean opinion scores are readily available for setting precise targets for the objective metrics to meet. Nevertheless, informal subjective evaluation can be used to rank different segmentation results, thus establishing a target for the objective evaluation methodologies.

Figure 2. Example of segmentation partition with one object (the head) having better segmentation quality



When assessing the performance of segmentation algorithms, two types of measurements can be envisaged:

- **Individual object segmentation quality evaluation:** Each of the objects identified by the segmentation algorithm is independently evaluated.
- **Overall segmentation quality evaluation:** The complete partition produced by the segmentation algorithm is evaluated. Besides evaluating each individual object, it is important to check if the correct objects were detected. Also, the most relevant objects, having a stronger impact in the overall quality perception, should receive higher weights in the developed evaluation metrics.

Individual object segmentation quality evaluation is needed since each object may be independently stored in a database, or reused in a different context, depending on the adequacy of its segmentation quality for the new purpose targeted.

Overall segmentation quality evaluation, determining if the segmentation goals for a certain application have been globally met, is also of great importance. In fact, the target segmentation quality may vary for different objects; for instance, in personal communications the head region assumes a higher relevance, being less tolerant to segmentation errors—see example in Figure 2.

In general, a good segmentation quality is required for those video objects that will be potentially reused and manipulated individually.

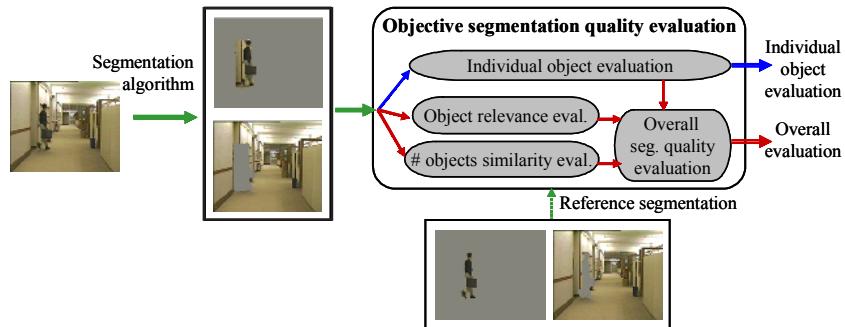
Segmentation Quality Evaluation Methodology

The general methodology for individual object segmentation quality evaluation proposed, consists in three major steps:

1. Apply the segmentation algorithm to the selected image or video sequence;
2. Select the object which segmentation is to be evaluated; and
3. Compute the relative or standalone objective segmentation quality evaluation.

For overall segmentation quality evaluation, a segmentation partition, including several objects, is evaluated as a whole. In this case, the overall segmentation quality evaluation methodology proposed is:

Figure 3. Summary of objective segmentation quality evaluation steps



1. Apply the segmentation algorithm to the selected contents;
2. Assess each object's individual segmentation quality, using either standalone or relative evaluation;
3. Weight individual segmentation quality scores with the corresponding object's relevance in the scene. Object relevance reflects the ability of an object to attract human visual attention and thus its likeliness of being reused;
4. Adjust the segmentation quality score to reflect the accuracy in detecting the targeted number of objects; and
5. Compute the final, overall segmentation quality evaluation score.

The segmentation quality measures can be quantitative (usually normalised to the [0,1] range) or qualitative (e.g., good, acceptable or unacceptable).

When dealing with video sequences, the quality measures computed for each time instant must be condensed into a single value, for example, by temporal average or median.

The main steps of individual object and overall objective segmentation quality evaluation are summarised in Figure 3.

Individual Object Standalone Segmentation Quality Evaluation

In the following, a set of features and corresponding metrics relevant for standalone evaluation are presented. These elementary metrics are established based on the expected object's feature values (intra-object metrics), and the disparity to neighbors (inter-object metrics).

Composite metrics for individual object standalone segmentation quality evaluation are proposed, recognizing that the applicability and importance of each elementary metric depends on the type of video content and application addressed.

Whenever needed, elementary metric results are normalized (m_n) to the [0, 1] interval, with highest scores corresponding to better segmentation results, using:

$$m_n = \max \left[0, \left(\frac{1}{1 + \frac{m}{m_{lim}}} - 0.5 \right) \cdot 2 \right] \quad (16)$$

This assumes error metrics producing values (m) close to zero for good segmentations, and values higher than m_{lim} corresponding to very low segmentation quality. When metric values are not naturally bounded, m_{lim} is determined by exhaustive testing.

Intra-Object Homogeneity Metrics

Spatial and temporal intra-object features can be evaluated in terms of homogeneity. The selected spatial features, and corresponding metrics, are:

- **Shape regularity:** Evaluated by geometrical features such as the compactness (*compact*), or a combination of circularity and elongation (*CE*) for object k :

$$\text{compact}(k) = \max \left(\frac{\text{comp}(k)}{75}, \quad 1 \right) \quad (17)$$

$$\text{CE}(k) = \max \left(\text{circ}(k), \quad \max \left(\frac{\text{elong}(k)}{5}, \quad 1 \right) \right) \quad (18)$$

with

$$\text{comp}(k) = \frac{P^2(k)}{A(k)}, \quad \text{circ}(E) = \frac{4 \cdot \pi \cdot A(k)}{P^2(k)}, \quad \text{elong}(k) = \frac{A(k)}{(2 \cdot T(k))^2} \quad (19)$$

where A , P and T are the object's area, perimeter and thickness—as defined in Serra (1993).

- **Spatial uniformity:** Evaluated by the spatial perceptual information (*SI*)—see equation 9—and texture variance (σ_{text}^2) metrics:

$$\sigma_{text}^2(k) = \frac{3 \cdot \sigma_Y^2(k) + \sigma_U^2(k) + \sigma_V^2(k)}{5} \quad (20)$$

where Y , U and V are the three colour components for object k .

For normalisation using equation 16, m_{lim} is set to 128 for *SI* and to 256 for σ_{text}^2 .

The selected **temporal features** and metrics are:

- **Temporal stability:** Smooth evolution of object features provides an indication of temporal stability. Following testing, the selected features were area, elongation and criticality:

$$A_{diff} = |A(k_t) - A(k_{t-1})| \quad (21)$$

$$elong_{diff} = |elong(k_t) - elong(k_{t-1})| \quad (22)$$

$$crit_{diff} = |crit(k_t) - crit(k_{t-1})| \quad (23)$$

Normalisation considers n_{lim} as the maximum feature value for the two considered instants.

- **Motion uniformity:** Evaluated by the criticality (*crit*), or the object's motion vector variance (σ_{mot}^2):

$$\sigma_{mot}^2(k) = \sigma_x^2(k) + \sigma_y^2(k) \quad (24)$$

where σ_x^2 and σ_y^2 denote *x* and *y* motion vector component variances.

For normalization m_{lim} is set to 5 for *crit* and 12 for σ_{mot}^2 .

Inter-Object Disparity Metrics

The disparity of a selected object's features against its neighbors' provides an indication of whether objects were correctly identified as separate entities. Both local features, computed along object boundaries, as well as features for the complete object, can be considered. The selected disparity metrics are:

- **Local contrast:** Evaluates the contrast between the inside and outside of an object's border:

$$contrast(k) = \frac{1}{4 \cdot 255 \cdot N_b} \cdot \sum_{i,j} (2 \cdot \max(DY_{ij}(k)) + \max(DU_{ij}(k)) + \max(DV_{ij}(k))) \quad (25)$$

where N_b is the number of border pixels for the object and $DY_{ij}(k)$, $DU_{ij}(k)$ and $DV_{ij}(k)$ are the differences between object k border pixels Y, U and V components and its 4-neighbors.

- **Feature difference to neighbors:** Evaluates the differences in a selected object's features to its neighbors'. Any spatial or temporal feature, relevant for the target application, can be selected.

As objects often move against a relatively stable background, a metric for motion uniformity difference is considered particularly interesting:

$$motUnif_{diff}(k) = \frac{1}{N} \cdot \sum_{j \in NS_k} |motUnif(j) - motUnif(k)| \quad (26)$$

where N and NS_k are the number and set of neighbors of object k , and each object's motion uniformity is:

$$motUnif(k) = crit(k) + \sigma_{mot}^2(k) \quad (27)$$

Individual Object Standalone Evaluation Composite Metrics

The standalone elementary metric's usefulness and applicability strongly depend on the video content and application characteristics. Thus, a single general-purpose composite metric for standalone evaluation cannot be established.

The approach taken here is to select two major content classes, representative of different application scenarios (Correia & Pereira, 2004), and propose composite metrics for each of them. The selected classes are:

- **Stable content:** Several applications, like video telephony and video conferencing, are characterized by temporally stable video contents, including objects with reasonably regular shapes and well contrasted to their neighbors.
- **Uniformly moving content:** Applications characterized by contents with strong but uniform motion, different from their neighbors' motion, are found, for instance, in sports events. Less spatial contrast is expected and shape regularity is not as important.

Stable content standalone segmentation quality evaluation should include shape regularity, temporal stability and local contrast metrics. The proposed composite metric, ($SQIo_{stdStable}$), for object k is obtained by temporal averaging over N instantaneous values:

$$SQIo_{stdStable}(k) = \frac{1}{N} \cdot \sum_{t=1}^N SQIo_{stdStable}(k_t) \quad (28)$$

with

$$SQIo_{stdStable}(k_t) = intra(k_t) + inter(k_t) \quad (29)$$

$$\begin{aligned} intra(k_t) = & 0.30 \cdot (0.5 \cdot CE(k_t) + 0.5 \cdot compact(k_t)) + \\ & 0.33 \cdot (0.33 \cdot A_{diff}(k_t) + 0.33 \cdot elong_{diff}(k_t) + 0.33 \cdot crit_{diff}(k_t)) \end{aligned} \quad (30)$$

$$inter(k_t) = 0.37 \cdot contrast(k_t) \quad (31)$$

The adopted weights reflect the higher impact of spatial versus temporal metrics for this type of content, and were adjusted after exhaustive testing.

Uniformly moving content's proposed composite metric, ($SQIo_{stdMoving}$), for object k includes shape regularity, motion uniformity, local contrast to neighbors and neighboring object features difference metrics:

$$SQIo_{stdMoving}(k) = \frac{1}{N} \cdot \sum_{t=1}^N SQIo_{stdMoving}(k_t) \quad (32)$$

with

$$SQio_{stdMoving}(k_t) = intra(k_t) + inter(k_t) \quad (33)$$

$$intra(k_t) = 0.28 \cdot (0.5 \cdot CE(k_t) + 0.5 \cdot compact(k_t)) + 0.29 \cdot crit(k_t) \quad (34)$$

$$inter(k_t) = 0.19 \cdot contrast(k_t) + 0.24 \cdot motUnif_{diff}(k_t) \quad (35)$$

The selected weights reflect a more balanced combination of the spatial and temporal aspects.

Individual Object Relative Segmentation Quality Evaluation

Individual object relative segmentation quality evaluation is based on dissimilarity (or error) metrics when comparing the estimated segmentation results against a reference.

The error metrics (e) are converted into similarity values (s_n) in the $[0, 1]$ interval, using:

$$s_n = \begin{cases} \frac{1}{1 + \frac{e}{e_{lim}}} - 0.5 & e < e_{lim} \\ 2 & e \geq e_{lim} \end{cases} \quad (36)$$

where e_{lim} is the metric's maximum value, or a value sufficiently high to represent bad quality, reflecting the HVS saturation effect in error perception. Perfect object segmentation achieves a similarity value of one.

In the following, several error metrics are defined; the corresponding similarity metrics are denoted starting with $s_$.

Spatial Accuracy Metrics

Good segmentations must present object shapes very similar to the reference. When shape match is not perfect, selected object features can be compared so that more objectionable segmentation errors are detected, lowering the segmentation quality scores.

The selected spatial accuracy features, and metrics, are:

- **Shape fidelity:** Evaluated by a metric similar to the one proposed in Villegas, et al. (1999), weighting misclassified pixels in the estimated (E) object with their distances to the reference (R):

$$dwsf(k) = \sum_i w_{MF}(i) \cdot Card\{p \in (R \cap E^C) : d(p, R^C) = i\} + \sum_i w_{AB}(i) \cdot Card\{p \in (E \cap R^C) : d(p, R) = i\} \quad (37)$$

- where $d(p, S)$ is the minimum Euclidean distance from pixel p to set S , with w_{MF} and w_{AB} given by equation 5.
- Geometrical similarity:** Evaluated by the observed object differences in area (A_d), position (pos) and a combination of elongation and compactness (EC):

$$A_d(k) = |A(R) - A(E)| \quad (38)$$

$$pos(k) = \sqrt{(X_c(R) - X_c(E))^2 + (Y_c(R) - Y_c(E))^2} \quad (39)$$

$$EC(k) = \left| \frac{elong(R) - elong(E)}{10} + \frac{comp(R) - comp(E)}{150} \right| \quad (40)$$

- where $X_c(k)$ and $Y_c(k)$ are the x and y coordinates of the object's gravity center.
- Edge content similarity:** Evaluated using the SI metric, defined in equation 9, and comparing the edges detected by a Sobel filter ($edge$), at a given time instant:

$$edge(k) = avg |Sobel(R) - Sobel(E)| \quad (41)$$

$$SI_d(k) = |SI(R) - SI(E)| \quad (42)$$

- Statistical data similarity:** Evaluated comparing the average brightness and redness of objects ($stat$), to reflect higher HVS sensitivity to bright and red areas:

$$stat(k) = \frac{3 \cdot |\bar{Y}(R) - \bar{Y}(E)| + |\bar{V}(R) - \bar{V}(E)|}{4 \cdot 255} \quad (43)$$

Temporal and Spatiotemporal Accuracy Metrics

The selected metrics, exploiting the temporal dimension of videos, are:

- Temporal perceptual information:** Evaluated comparing the instantaneous values of the temporal perceptual information, defined in equation 10:

$$TI_d(k) = |TI(R) - TI(E)| \quad (44)$$

- Criticality:** Evaluated by comparing the difference in criticality, as defined in equation 11:

$$crit_d(k) = |crit(R) - crit(E)| \quad (45)$$

Relative Segmentation Quality Evaluation Composite Metric

The relative segmentation quality evaluation of individual objects can be done using a single composite metric, which combines the elementary metrics capturing those differences to the reference that may affect segmentation quality.

In order to adjust the weights assigned to each elementary metric, a set of informal subjective tests were conducted at “Instituto de Telecomunicações,” in Lisbon, using the MPEG-4 video test set (Correia, 2002).

The proposed composite metric for individual object relative segmentation quality evaluation ($SQio_{rel}$), for object k , is obtained by temporal averaging the N instantaneous values:

$$SQio_{rel}(k) = \frac{1}{N} \cdot \sum_{t=1}^N SQio_{rel}(k_t) \quad (46)$$

with,

$$SQio_{rel}(k_t) = spatial(k_t) + temporal(k_t) + spatTemp(k_t) \quad (47)$$

$$\begin{aligned} spatial(k_t) &= shapeFidelity(k_t) + geom(k_t) + edge(k_t) + statistical(k_t) \\ temporal(k_t) &= 0.154 \cdot s_TI_d(k_t), \quad spatTemp(k_t) = 0.078 \cdot s_crit_d(k_t), \\ shapeFidelity(k_t) &= 0.48 \cdot s_dwsf(k_t), \\ geom(k_t) &= 0.048 \cdot s_A_d(k_t) + 0.0336 \cdot s_pos(k_t) + 0.0144 \cdot s_EC(k_t) \\ edge(k_t) &= 0.048 \cdot s_edge(k_t) + 0.048 \cdot s_SI_d(k_t), \text{ and } statistical(k_t) = 0.096 \cdot s_stat(k_t). \end{aligned}$$

Shape fidelity takes the largest weight (around 50%) as it is the main indication of a mismatch with the reference. Temporal fidelity receives the second highest weight, reflecting the importance of temporal information in terms of HVS. The remaining metrics account for a little over one third of the total weight, distinguishing segmentations that present spatial and temporal errors.

Overall Segmentation Quality Evaluation

Overall segmentation quality is evaluated following the methodology illustrated in Figure 3. It weights individual object segmentation quality with the object’s relevance, and also takes into account the accuracy in detecting the correct number of objects (similarity of objects) for the target application.

Proposals for object relevance evaluation, similarity of objects and the overall segmentation quality composite metric are presented below.

Contextual Object Relevance Evaluation

When evaluating a complete partition, the individual objects that most capture human attention should receive higher weights in the overall segmentation quality evaluation. This requires the ability to evaluate each object’s relevance, taking into account the context where objects are found, like contrast to neighbors or object

position. Each object's contextual relevance metric then assumes absolute values (RC_{abs}) between 0 and 1, the higher values corresponding to more relevant objects.

For overall segmentation quality evaluation, it is desirable that object relevancies at a given instant sum to 1, to keep the segmentation quality score in the [0, 1] range. This leads to the definition of a relative contextual object relevance metric (RC_{rel}), computed from the corresponding absolute scores for all the objects (O_N) in the segmentation partition:

$$RC_{rel}(k_t) = \frac{RC_{abs}(k_t)}{\sum_{j=1}^{O_N} RC_{abs}(j_t)} \quad (48)$$

The contextual relevance metric proposed in Correia and Pereira (2000) is adopted here for overall segmentation quality evaluation purposes:

$$RC_{rel}(k) = \frac{1}{N} \cdot \sum_{t=1}^N RC_{rel}(k_t) \quad (49)$$

where N is the number of temporal instants considered, relative relevance values are computed using equation 48 and the absolute contextual relevance for object k is:

$$RC_{abs}(k_t) = 0.3 \cdot motion_activ(k_t) + 0.25 \cdot comp(k_t) + 0.13 \cdot high_level(k_t) + 0.1 \cdot shape(k_t) + 0.085 \cdot bright_red(k_t) + 0.045 \cdot (contrast(k_t) + position(k_t) + size(k_t)) \quad (50)$$

The elementary metrics in equation 50, defined in Correia and Pereira (2000), correspond to features capturing visual attention, such as object motion, texture complexity or detection of objects bearing semantic information like faces; also large, centered and contrasted objects, or objects with some types of shapes and colors, seem to be preferred by human viewers and are more valued by the metrics used.

Object relevance metrics can also be seen as a measure of the likeliness that one object will be further processed, manipulated and used, as users tend to select those objects that capture their attention most.

Similarity of Objects

Partitions with missing or extra objects indicate a deviation from the segmentation goals. To account for this aspect in the overall segmentation quality metric, an object's similarity metric (OS) is proposed, for both standalone and relative evaluation.

For standalone evaluation, object similarity must often be restricted to a comparison between the estimated (E_N) and targeted (T_N) number of objects. For some applications, T_N may be known *a priori*, even if in general it can vary along time. In these cases, the proposed instantaneous object comparison metric (OC) is:

$$OC(t) = \frac{\min(E_N(t), T_N(t))}{\max(E_N(t), T_N(t))} \quad (51)$$

Whenever small variations in the number of objects are expected, a temporal stability (TS) measure can be applied:

$$TS(t) = \frac{\min(E_N(t-1), E_N(t))}{\max(E_N(t-1), E_N(t))} \quad (52)$$

The proposed object similarity metric combines OC and TS :

$$OS(t) = OC(t) \cdot TS(t) \quad (53)$$

For relative evaluation, the target number of objects is always known from the reference. In this case, the proposed OS metric, at instant t , is:

$$OS(t) = \frac{A_M(t)}{A_I(t)} \quad (54)$$

where $A_I(t)$ is the total image area and $A_M(t)$ is the portion of that area covered by the objects for which there is a correct match with the reference segmentation.

This OS metric implicitly takes into account the difference between the estimated and reference number of objects, but it is also strongly influenced by the relative size of missing/extraneous objects, assuming that large non-matched objects lead to worse segmentation results.

An OS metric representative of the complete sequence can be obtained by temporal averaging.

Overall Segmentation Quality Evaluation Metric

The overall segmentation quality evaluation metric combines the individual object quality measures, standalone or relative, and the object's relevance and similarity factors. The instantaneous values for each of these three components are combined, in order to reflect the variations in any of them that may occur along time.

The proposed overall segmentation quality evaluation metric (SQ) is:

$$SQ = \frac{1}{N} \cdot \sum_{t=1}^N \left[OS(t) \cdot \sum_{k=1}^{O_N} (SQio(k_t) \cdot RC_{rel}(k_t)) \right] \quad (55)$$

where N is the sequence length and $SQio$ is the individual object segmentation quality value for object k (either $SQio_{rep}$, $SQio_{stdStable}$ or $SQio_{stdMoving}$). The inner sum is performed for all the objects in the estimated partition at time instant t .

This SQ metric provides better overall segmentation quality when the individual object quality is higher for the most relevant objects. An incorrect match between target and estimated objects also penalizes the segmentation quality.

Video Segmentation Quality Evaluation Results

Results obtained with the segmentation quality evaluation metrics proposed above, both standalone and relative, are discussed below, after presenting the set of test sequences and segmentation partitions used.

Test Sequences and Segmentation Partitions

A subset of the MPEG-4 test sequences, with different spatial complexity and temporal activity characteristics, were selected, together with several segmentation partitions with different qualities, to illustrate the proposed segmentation quality evaluation metrics' potential and limitations. Experiments have shown that segmentation quality results are rather independent of the sequences' spatial resolution, thus the QCIF format was chosen to limit the algorithm execution time.

Subsets from three sequences, each with 30 representative images, are used here to illustrate the obtained results:

- **Akiyo**, images 0 to 29—Low temporal activity and not very complex texture; two objects of interest: woman and background—see sample in Figure 4.

Figure 4. Image 29 of sequence Akiyo: original (a) and reference (b), seg1 (c), seg2 (d), seg3 (e) and seg4 (f) segmentation partitions

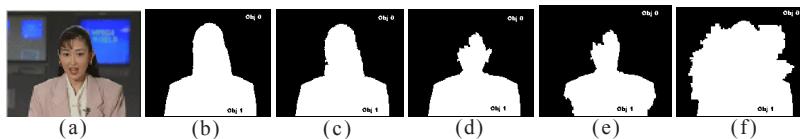


Figure 5. Image 30 of sequence Stefan: original (a) and reference (b), seg1 (c), seg2 (d), seg3 (e) and seg4 (f) segmentation partitions

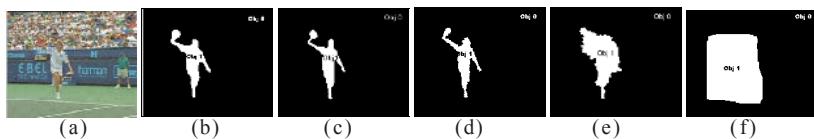
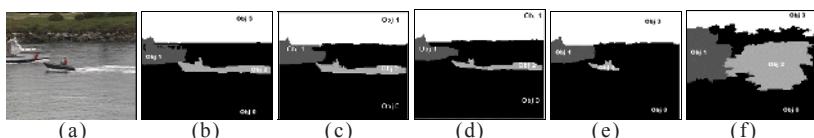


Figure 6. Image 29 of sequence Coastguard: original (a) and reference (b), seg1 (c), seg2 (d), seg3 (e) and seg4 (f) segmentation partition



- **Stefan**, images 30 to 59—High temporal activity and relatively complex texture; two objects of interest: tennis player and background—see sample in Figure 5.
- **Coastguard**, images 0 to 29—Moderate temporal activity and moderately complex texture; four objects of interest: water, large boat, small boat plus water tail and land—see sample in Figure 6.

The reference segmentation partitions shown are made available by the MPEG group. The estimated partitions have segmentation qualities ranging from a close match with the reference to more objectionable segmentations.

Standalone Segmentation Quality Evaluation Results

For standalone segmentation quality evaluation, the two previously considered content classes (stable and moving) are analyzed, as standalone elementary metrics are only applicable under certain circumstances.

All available segmentations are evaluated, including the MPEG reference ones, whose quality may vary depending on how it was generated. For each test sequence, results include: overall segmentation quality temporal evolution, temporal average of instantaneous segmentation quality for individual objects and overall partition.

Test sequence *Akiyo* represents the stable content class, as its objects present relatively regular shapes and a limited amount of motion. For the test partitions presented in Figure 4, a human observer's subjective evaluation would most likely list the *reference* and *seg1* as having the best quality, followed by *seg2*, then *seg3* and finally *seg4*.

Results of the proposed objective metrics are presented in Figure 7, showing three segmentation quality groups for the *woman* object: Best quality is achieved by the *reference*, *seg1* and *seg2*, while *seg3* achieves intermediate quality and *seg4* gets the worst result. In this case, the best evaluation result is not for the *reference* segmentation, as part of the woman's hair is intensely illuminated, and when included into the *woman* object it leads to lower contrast to the background than if omitted, as happens with *seg1* and *seg2*. *Seg4*, for which the *woman* object captures a significant part of the *background*, is clearly identified as the worst segmentation.

Figure 7. Overall and individual object standalone segmentation quality evaluation results for sequence *Akiyo*

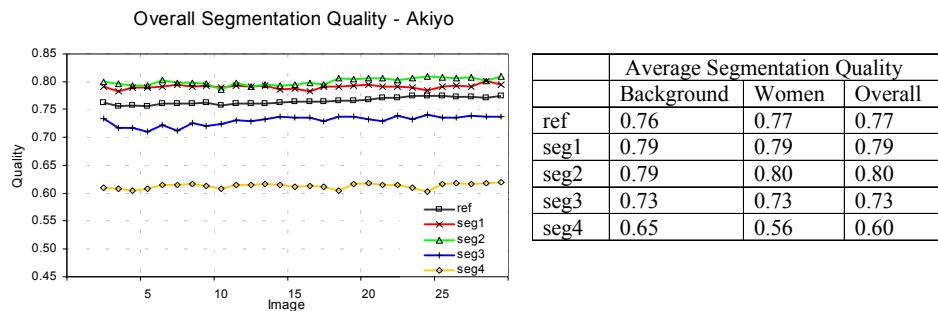
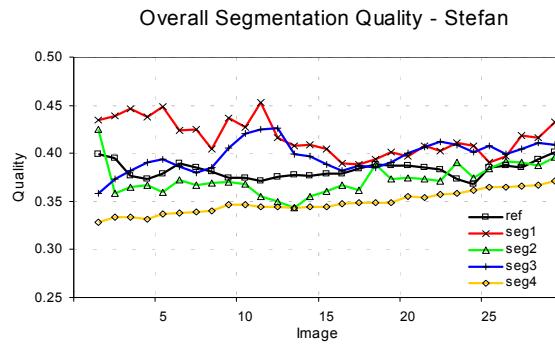


Figure 8. Overall and individual object standalone segmentation quality evaluation results for sequence Stefan

	Average Segmentation Quality		
	Background	Player	Overall
ref	0.33	0.43	0.38
seg1	0.34	0.49	0.42
seg2	0.32	0.43	0.38
seg3	0.34	0.45	0.39
seg4	0.32	0.37	0.34



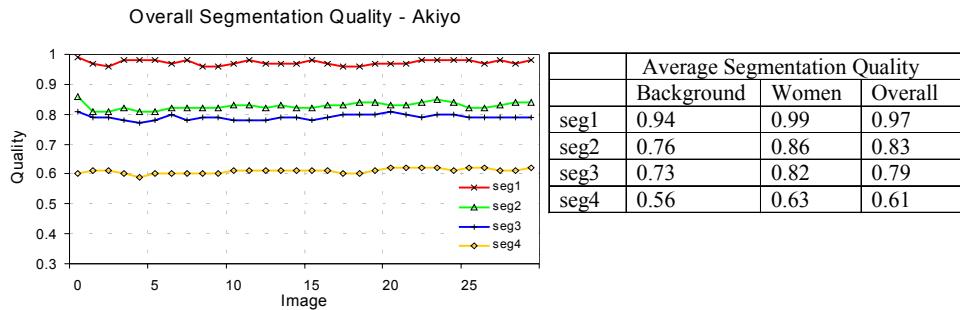
Test sequence *Stefan* represents the uniformly moving content class. Contents are spatially more complex and include higher temporal activity. For this sequence, a human observer would tend to rank the *reference* and *seg1* as having the best quality, closely followed by *seg2*, then *seg3* and finally *seg4*.

Objective evaluation results, presented in Figure 8, give the highest overall segmentation quality to *seg1*, followed by the *reference*, *seg2* and *seg3*; *seg4* gets the worst result. In fact, *seg1* is more precise than the *reference*, which is smoother and sometimes includes *background* fragments in the *player* object. The *reference* and *seg2* are correctly classified into the next quality group. *Seg3* receives a ranking higher than expected because it always includes the moving *player* object, which is not strongly contrasted to the surrounding *background* area. *Seg4* is correctly ranked as the worst.

The *Stefan* overall segmentation results are always quite low since the standalone objective evaluation metrics do not find the objects to be very homogeneous in either texture or motion, and cannot conclude that the best segmentations are reasonably good for a human observer.

Analyzing the above standalone segmentation quality evaluation results shows that the proposed metrics are able to rank the various segmentation partitions as a human observer would, but results must be interpreted in a qualitative way, rather than as definitive segmentation quality scores.

Figure 9. Overall and individual object-relative segmentation quality results for sequence *Akiyo*



Relative Segmentation Quality Evaluation Results

Relative evaluation results are obtained by comparison against the reference partitions, thus promising more precise results.

Figure 9 includes results for sequence *Akiyo*, showing a very high overall segmentation quality (0.97 on average) for *seg1*, which agrees with human observation since shape differences to the *reference* are minor. *Seg2* and *seg3* get intermediate segmentation quality scores, the differences reflecting the observed size and geometry variations in the *woman* object, in both cases including the moving areas while the missing parts' texture is quite homogeneous. *Seg4* gets the lowest quality score as expected, but with a relatively high overall quality score (0.61). A human observer would, in principle, assign a lower quality to *seg4* since recognizing the *woman* object as a semantically important object (a person) would lead to less tolerance to any distortions. By not detecting and valuing semantically important objects, like faces, *seg4* is considered as simply adding a portion of the *background* to the *woman* object, and the quality metric is not as much affected as might be expected.

Results for sequence *Stefan* are included in Figure 10. *Seg1* has very precise contours, getting the best overall segmentation quality scores. *Seg2* is ranked second as it presents relatively small segmentation mismatches. *Seg3* presents significant mismatches for the *player* object, but the resulting shape still resembles the *reference*. *Seg4* gets the worst result as expected, but even if the *player* object shape is very different from the *reference*, it completely includes the moving object, leading to a quality score not as low as might be expected.

For the sequence *Coastguard*, the expected human observer overall segmentation ranking would be *seg1* as the best, followed by *seg2* and *seg3*, *seg4* being the worst. *Seg4* has the lowest quality for all objects. In segmentations *seg1*, *seg2* and *seg3*, the *water* and *land* objects do not show very significant differences, with the errors having approximately the same impact to the human viewer. Perceived errors for the *large boat* object would lead to ranking segmentations according to their numbering order: *seg1*, *seg2* and *seg3*. The same happens for the *small boat* object as *seg1* is very similar to the

Figure 10. Overall and individual object-relative segmentation quality results for sequence Stefan

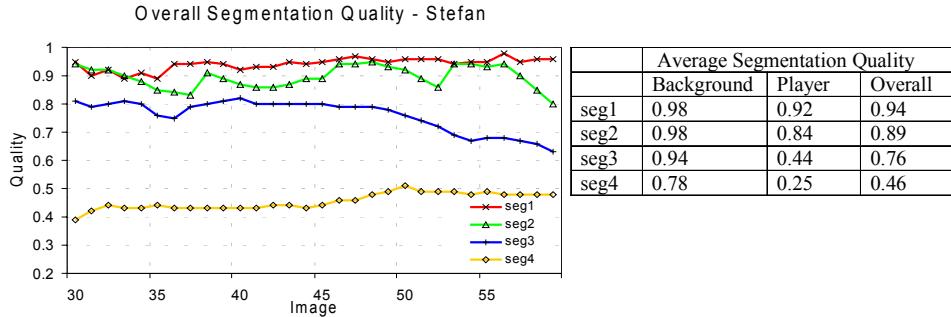
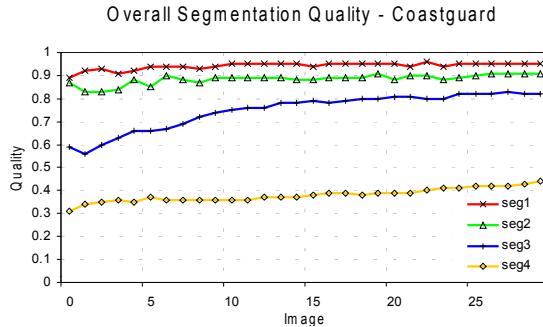


Figure 11. Overall and individual object-relative segmentation quality results for sequence Coastguard

	Average Segmentation Quality				
	Water	Large Boat	Small Boat	Land	Overall
seg1	0.97	0.93	0.92	0.98	0.94
seg2	0.92	0.81	0.89	0.96	0.88
seg3	0.87	0.71	0.60	0.92	0.75
seg4	0.59	0.29	0.22	0.62	0.38



reference, seg2 shows some shape differences and seg3 does not include the water tail. Overall, the segmentation quality of seg3 tends to increase as the *large boat* enters the scene and the differences resulting from the missing *water tail* in the *small boat* object are less valued.

Comparing the objective evaluation results presented in Figure 11 with the subjectively expected results, a rather good match is observed both for individual object and overall segmentation quality results.

As shown in the examples above, relative segmentation quality evaluation obtained with the proposed metric generally agrees with the subjective evaluation. It is, however, recognized that the inclusion of metrics for detecting and accounting for semantically relevant objects, such as people, where segmentation errors are more objectionable, could improve the performance.

Relative Evaluation Results for an Incorrect Number of Objects

To illustrate the behavior of the proposed overall relative segmentation quality evaluation metrics when an incorrect number of objects is detected, four partitions derived from the *Coastguard* reference segmentation were created by merging or splitting some objects—see Figure 12.

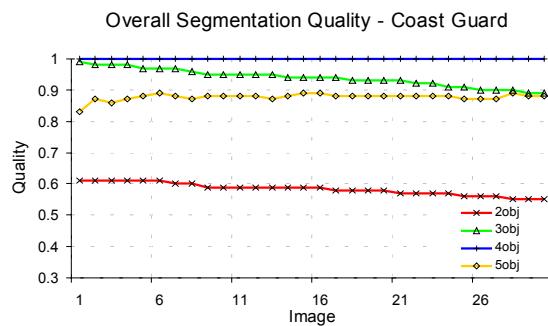
Figure 13 includes overall relative segmentation quality evaluation results for these four segmentation partitions.

Figure 12. Segmentation partitions considering 2, 3, 4 and 5 objects, for image 29 of sequence Coastguard



Figure 13. Overall relative segmentation quality evaluation for Coastguard segmentations with two (2obj), three (3obj), four (4obj) and five objects (5obj)

	Average Segmentation Quality				
	Water	Large Boat	Small Boat	Land	Overall
2 obj	0.59	-	1	-	0.58
3 obj	0.88	-	1	1	0.94
4 obj	1	1	1	1	1
5 obj	1	1	0.70	1	0.88



For the *2obj* segmentation, the evaluation for the *small boat* object was perfect (score 1) as it exactly matches the reference. The *background* was compared to the *water* reference object; as the reference object does not include the *large boat* and *land* areas, a lower quality score results (average of 0.59). The overall quality score (average of 0.58) is lower than the relevance-weighted average of the individual object segmentation qualities since the non-matched area between estimated and reference partitions is accounted for through the object's similarity metric.

The *3obj* segmentation additionally detects the *land* object, resulting in a quality increase for the *water* object (average of 0.88). Since the other objects have a quality score of 1 and the area corresponding to the non-detected *large boat* is small (in the sequence portion considered), the overall segmentation quality result is high (average of 0.94). Notice the instantaneous quality score decreases as the *large boat* enters the scene, due to the object's similarity penalization.

The *4obj* segmentation, being the reference segmentation, achieves a perfect overall quality.

Finally, *5obj* has one object that could not be matched to the reference: The *water tail* is considered unmatched, meaning an imperfect segmentation of the *small boat* object. The overall segmentation quality average value is 0.88.

From a human observer's point of view, the *3obj* segmentation is less objectionable than the *5obj* segmentation, as long as the large boat is mostly outside of the scene, but once it enters the scene, then the *5obj* segmentation becomes preferable, even with the *small boat* segmentation mismatch. This behavior is correctly reflected in the temporal evolution of the objective overall segmentation quality shown in the chart of Figure 13.

As shown by the above example, the proposed object's similarity metric is effective in reducing the overall segmentation quality score when target and estimated objects are not correctly matched.

CONCLUSIONS

Some video analysis tasks are very intuitive for human beings, but constitute a challenge for computer algorithms, except in rather well constrained scenarios. One such task is video segmentation, which provides data essential for subsequent higher-level of abstraction analysis, like object recognition or semantic interpretation of actions, but also for achieving better subjective quality in video coding, even in frame-based coding environments.

This chapter addressed the problem of video segmentation quality objective evaluation. Both relative and standalone segmentation quality evaluation metrics were proposed, to be used when a reference segmentation is available, or not, to compare against the estimated segmentation. These metrics can be applied both to individual objects and to overall scenes.

Individual object evaluation is of interest whenever objects are viewed as independent entities that may be reused in different contexts. Overall evaluation considers a segmentation partition structuring the complete video scene. In this case, the individual evaluation of the most relevant objects is given a larger weight, as they have a greater impact on the overall perceived quality. Also the similarity between the estimated and targeted objects influences the overall evaluation.

The features to consider for standalone objective evaluation depend on the assumptions that can be made for the application scenario considered. Consequently, different composite metrics were proposed for stable and for moving content scenarios. Moreover, standalone evaluation results should be interpreted qualitatively rather than quantitatively.

Relative segmentation quality evaluation provides more reliable and discriminative results. Summarizing, no well-established solution is yet available for objective segmentation quality evaluation, and the methodology and metrics proposed here are expected to provide a relevant contribution to this important area.

FUTURE TRENDS

Future work related to objective video segmentation quality evaluation includes two major areas: improvement of evaluation metrics, and using evaluation results to improve the behaviour of segmentation algorithms and other applications exploiting segmentation information.

Objective evaluation metrics can be further developed, for instance including the ability to detect those objects with a central role in specific applications, and whose quality variations are more objectionable. More generally, the introduction of semantic information for segmentation quality evaluation is expected to improve the meaningfulness of the evaluation results. Techniques for gathering such semantic information, include object classification; examples of relevant object classes are faces, facial elements (eyes, nose, mouth and ears), the human body, vehicles and their license plates, text or defects in some type of product being manufactured. These object classes should have a higher impact in the segmentation quality results.

Also the semantic interpretation of the actions taking place in the video scenes allows further semantic insight to be acquired, enabling more sophisticated applications to be developed and decisions to be taken. In terms of segmentation quality evaluation, the temporal portion of a sequence when more relevant actions take place should be more critically evaluated, with segmentation errors contributing more to penalise the segmentation quality. Examples include the determination of the type of action undertaken, using human pose detection and interpretation, or the detection of certain types of events, such as goal scoring or penalty kicks in a football match.

The most obvious application of objective segmentation quality evaluation results, besides assessing the adequacy of a segmentation algorithm for a given application, is to improve the behaviour of the segmentation algorithms themselves, by using a feedback mechanism. For instance, when segmentation quality decreases due to merging too many objects, the contrast to neighbours elementary metric is expected to become less discriminative, and can deploy a feedback mechanism to ask the segmentation algorithm not to merge so many objects. Another example, when user interaction with the segmentation process is supported for instance, is to let the user signal the objects of interest; or, when correcting automatic segmentation results, is to have the segmentation algorithms learn from user interactions and the latter proposing similar actions for equivalent situations.

Segmentation quality evaluation results can also be exploited by applications that benefit from the availability of segmentation results. For instance, in object-based video

coding applications, segmentation quality evaluation can provide useful information for the rate control algorithms, or to decide on the amount of error resilience information to devote to each object, thus contributing to increase the perceived video quality at the receiver.

Segmentation quality together with relevance metrics, like the ones proposed in this chapter, can also be applied in different fields, such as object-based description, for example, to select the amount of detail to store about an object, or content adaptation, for example, to select the amount of detail to keep in a transcoding application.

While it is clear that objective segmentation quality evaluation metrics have a large range of potential applications which are just starting to be explored, the segmentation quality metrics themselves can still be further developed and customized for specific applications.

REFERENCES

- Abdou, I., & Pratt, W. (1979). Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5), 753-763
- Bryant, D., & Bouldin, D. (1979). Evaluation of edge operators using relative and absolute grading. In *Proceedings of the IEEE Conference on Pattern Recognition Image Processing* (pp. 138-145). Chicago: IEEE.
- Correia, P. (2002). *Video analysis for object-based coding and description*. Portugal: Instituto Superior Técnico, Technical University of Lisbon.
- Correia, P., & Pereira, F. (2000). Estimation of video object's relevance. In *Proceedings of the 5th European Signal Processing Conference (EUSIPCO 2000)* (pp. 925-928). Tampere, Finland: EURASIP.
- Correia, P., & Pereira, F. (2002). Standalone objective segmentation quality evaluation. *EURASIP Journal on Applied Signal Processing*, 2002(4), 389-400.
- Correia, P., & Pereira, F. (2003). Objective evaluation of video segmentation quality. *IEEE Transactions on Image Processing*, 12(2), 186-200.
- Correia, P., & Pereira, F. (2004). Classification of video segmentation application scenarios. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5), 735-741.
- COST211quat European Project (2001), *Call for AM comparisons*. Retrieved on February 2005, from <http://www.iva.cs.tut.fi/COST211/Call/Call.htm>
- Erdem, C., Tekalp, A., & Sankur, B. (2001). Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In *Proceedings of IEEE International Conference on Image Processing (ICIP)* (Vol. 2, pp. 69-72). Thessaloniki, Greece: IEEE.
- Fenimore, C., Libert, J., & Wolf, S. (1998, October). Perceptual effects of noise in digital video compression. In Society of Motion Picture and Television Engineers (SMPTE) (Eds.), *140th SMPTE Technical Conference*, Pasadena, CA.
- Fram, J., & Deutsch, E. (1975). On the quantitative evaluation of edge detection schemes and their comparison with human performance. *IEEE Transactions on Computers*, 24(6), 616-628.
- Hamada, T., Miyaji, S., & Matsumoto, S. (1997, November). Picture quality assessment system by three-layered bottom-up noise weighting considering human visual

- perception. In Society of Motion Picture and Television Engineers (SMPTE) (Eds.), *139th SMPTE Technical Conference*, New York.
- Hekstra, A., Beerends, J., Ledermann, D., Caluwe, E., Kohler, S., Koenen, R., et al. (2002) PVQM—a perceptual video quality measure. *Signal Processing: Image Communication*, 17(10), 781-798
- International Standards Organisation (1999). *ISO/IEC 14496 (MPEG-4): Information technology—coding of audio-visual objects*, ISO.
- International Telecommunication Union—Radiocommunication Sector. (1995). *Recommendation BT.500-7: Methodology for the Subjective Assessment of the Quality of Television Pictures*. Switzerland: ITU.
- International Telecommunication Union—Telecommunication Sector. (1996). *Recommendation P.910: Subjective Video Quality Assessment Methods for Multimedia Applications*. Switzerland: ITU.
- International Telecommunication Union—Telecommunication Sector. (1996). *Recommendation P.930: Principles of a Reference Impairment System for Video*. Switzerland: ITU.
- Lambrecht, C. (1996). A working spatio-temporal model of the human visual system for image restoration and quality assessment applications. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP) (pp. 2293-2296). Atlanta, GA.
- Levine, M., & Nazif, A. (1985) Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2), 155-164.
- Mech, R., & Marques, F. (2002). Objective evaluation criteria for 2d-shape estimation results of moving objects. *EURASIP Journal on Applied Signal Processing*, 2002(4), 401-409.
- Osberger, W., Bergman, N., & Maeder, A. (1998). A technique for image quality assessment based on a human visual system model. In *Proceedings of the 9th European Signal Processing Conference* (EUSIPCO) (pp. 1049-1052). Rhodes, Greece.
- Rees, G., & Grenway, P. (1999). Metrics for image segmentation. In *Workshop on Performance Characterisation and Benchmarking of Vision Systems* (pp. 20-37). Essex, UK.
- Serra, J. (1993). *Image analysis and mathematical morphology*, 1. Academic Press.
- Strasters, K., & Gerbrands, J. (1991). Three-dimensional image segmentation using a split, merge and group approach. *Pattern Recognition Letters*, 12, 307-325.
- Tan, K., Ghanbari, M., & Pearson, D. (1998). An objective measurement tool for MPEG video quality. *Signal Processing*, 70, 279-294.
- Villegas, P., & Marichal, X. (2004). Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 13(8), 1092-1103.
- Villegas, P., Marichal, X., & Salcedo, A. (1999). Objective evaluation of segmentation masks in video sequences. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)* (pp. 85-88). Berlin, Germany.
- Weszka, J., & Rosenfeld, A. (1978). Threshold evaluation techniques. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8), 622-629.

- Wollborn, M., & Mech, R. (1998). *Doc. ISO/IEC JTC1/SC29/WG11 M3448: Refined procedure for objective evaluation of video object generation algorithms*. Tokyo, Japan: ISO.
- Yasnoff, W., Mui, J., & Bacus, J. (1977). Error measures for scene segmentation. *Pattern Recognition*, 9, 217-231.
- Zhang, Y. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335-1346.
- Zhang, Y., & Gerbrands, J. (1994). Objective and quantitative segmentation evaluation and comparison. *Signal Processing*, 39, 43-54.

Chapter XX

A Summary of Recent Progresses for Segmentation Evaluation

Yu-Jin Zhang, Tsinghua University, Beijing, China

ABSTRACT

This chapter provides a summary of the recent (especially since 2000) progress for the evaluation of image and video segmentation. It is seen that much more effort has been expended on this subject recently than several years ago. A number of works are based on previously proposed principles, and several works have made modifications to and improvements on previous techniques, still other works have presented a few new ideas. The generality and complexity of the evaluation methods and performance criteria used in these works have been thoroughly compared. As the research in this field is still on the rise, some existing problems and several future directions are also highlighted.

INTRODUCTION

After 40 years of development, image segmentation has grown into a major research field of image engineering (Zhang, in press). As discussed in Chapter I, the research for image segmentation has been carried on in three layers. The first, and also the basic one, is the layer of algorithm development, the second, or middle layer, is algorithm evaluation and the third is the top layer of systematic study of evaluation methods.

A large number of image segmentation techniques have been proposed and utilized in various applications, which form a solid base in the first layer of image segmentation research. While development of segmentation algorithms has attracted significant attention, relatively fewer efforts have been spent on their evaluation. Moreover, most

efforts spent on segmentation evaluation are just for designing new evaluation methods and/or new performance metrics, which is only in the middle layer of image segmentation research. Attempts to characterize the existing evaluation methods, that is, those at the top layer of image segmentation research, are still very much needed.

The quality assessment of the resulting image and video segmentation partition is of crucial importance to evaluate the degree to which the application targets are met. One of the early works in the top layer of image segmentation study was made in Zhang (1993). In 1996, the first comprehensive survey on segmentation evaluation, which provides a thorough review of different existing methods (at that time) for segmentation evaluation, as well as a discussion and comparison of their applicability, advantages and limitations, was published (Zhang, 1996). In 2001, a complementary review of the progress from 1996 to 2000 was made (Zhang, 2001).

As the accumulated effort on segmentation evaluation, more than 100 evaluation papers can be found in the literature now. This number already makes a comprehensive survey a challenging task; instead, only a summary of recent progress will be provided here.

Referring to the three layers of segmentation research, this chapter will focus on the new developments in segmentation evaluation techniques and performance metrics (second layer) and will make a rough characterization of these segmentation evaluation methods (third layer). Most discussions are focused on the literature published after 2000.

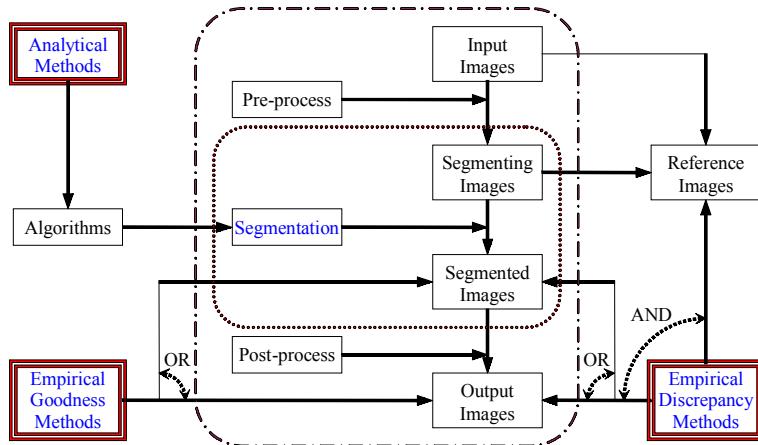
BACKGROUND

General Framework

A general framework for segmentation and its evaluation is depicted in Figure 1 (Zhang, 1996). The input image obtained by sensing is first (optionally) preprocessed to produce the segmenting image for the real segmentation (in its strict sense) procedure. The segmented image can then be (optionally) post-processed to produce the output image. In Figure 1, the part enclosed by the rounded square with dotted lines corresponds to the segmentation procedure in its narrow-minded sense, while the part enclosed by the rounded square with dash-point lines corresponds to the segmentation procedure in its general form.

Three groups of evaluation methods are shown by double-frame boxes in Figure 1. Note that there is an OR condition between both arrows leading to the boxes containing “segmented image” and “output image,” both from the “empirical goodness method” and “empirical discrepancy method.” Moreover, there is an AND condition between the arrow from “empirical discrepancy method” to “reference image” and the two (OR) arrows going to “segmented image” and “output image.” The analysis methods treat the algorithms for segmentation directly. The empirical goodness methods judge the segmented image or output image so as to indirectly assess the performance of algorithms. For applying empirical discrepancy methods, the reference image (often called ground truth, GT) is necessary. It can be obtained manually or automatically from the input image or segmenting image. The empirical discrepancy methods compare the segmented image

Figure 1. General framework for segmentation and its evaluation



or output image to the reference image and use their difference to assess the performance of algorithms.

Terminology and Taxonomy

In the following, some pairs of terms used in segmentation evaluation are first introduced and explained to set the background for further discussion.

Intra-Technique Evaluation vs. Inter-Technique Evaluation

The processes of segmentation evaluation can be classified into two types (Zhang, 1997): (1) intra-technique evaluation, which characterizes the behavior of a particular segmentation algorithm with different settings and parameters, whose purpose is to achieve the best performance in treating various images; (2) inter-technique evaluation, which ranks the performance of several segmentation algorithms, whose purpose is to provide guidelines in choosing suitable algorithms according to applications. The inter-technique evaluation can also help to promote new developments by effectively taking the strongpoints of several algorithms (Zhang, 1994).

Goodness Methods vs. Discrepancy Methods

According to the requirement about reference images, empirical evaluation methods can be classified into two classes: goodness methods and discrepancy methods. The former class can perform the evaluation without the help of reference images, while the latter class needs some reference images to arbitrate the quality of segmentation. The former is also called a stand-alone one and the latter is also called a reference one; stand-alone evaluation itself is sensitive to the type of applications and/or image/video

content, and yields a more qualitative rather than quantitative evaluation value (Correia, 2003b).

Subjective Criteria vs. Objective Criteria

One essential element and critical factor in segmentation evaluation are the criteria used in the evaluation method for judging the performance of segmentation. The (evaluation) criteria are also called (performance) metrics, measures, indices, and so forth. Empirical goodness methods use subjective criteria which reflect some desirable properties of segmented images, while empirical discrepancy methods use objective criteria which indicate the difference between the segmented images and reference images (Zhang, 1996).

Global Criteria vs. Local Criteria

Among objective criteria, some of them are global, which do not take into account local discrepancies of the segmented regions. The value of such criteria will be mainly determined by an average comportment, and the effect of a localized substantial discrepancy will be concealed by comparatively good behavior elsewhere (Moretti, 2000). The local criteria have the reverse characteristics.

Single Metric vs. Composite Metric

From the point of view of using evaluation criteria, either a single metric or several metrics have been used in different evaluation processes. When several metrics are used, often a compound metric is formed and can be called a composite metric. As the behavior of an algorithm is based on many factors, a single metric, or even a composite metric, for an entire assessment can hardly reach an optimal solution, since the combination of different metrics is often too empirical to be effective. On the other hand, using several metrics could better cover the various aspects of the algorithm, but a final score is still needed.

Image Segmentation Evaluation vs. Video Segmentation Evaluation

Compared to static (2D) images, video has one more dimension (time). Image segmentation evaluation can be performed only in spatial space while video segmentation evaluation should be performed in both spatial and temporal spaces. Inspired from spatial space, some evaluation criteria can be defined in parallel along the temporal axis. For example, temporal stability is a metric computing the difference in the object size for consecutive time instants; temporal features similarity is the difference between the values of selected object features associated to the temporal dimension and computed for the reference and segmented images. On the other side, those metrics designed for video segmentation quality measures can be converted to metrics for image segmentation quality measures by removing the motion and temporal related portions.

Empirical Methods and Criteria

Empirical evaluation is practically more usable than analysis evaluation. Empirical methods can be classified into goodness and discrepancy method groups. In Zhang (1996), eight empirical evaluation classes (three from goodness method group and five

Table 1. A new list of groups of empirical methods and criteria

Class	Criterion name	Method group
G-1	Intra-region uniformity	Goodness
G-2	Inter-region contrast	Goodness
G-3	Region shape	Goodness
G-4	Moderate number of regions	Goodness
D-1	Number of mis-segmented pixels	Discrepancy
D-2	Position of mis-segmented pixels	Discrepancy
D-3	Number of objects in the image	Discrepancy
D-4	Feature values of segmented objects	Discrepancy
D-5a	Region consistency	Discrepancy
D-5b	Grey level difference	Discrepancy
D-5c	Symmetric divergence (cross-entropy)	Discrepancy
S1	Amount of editing operations	Special
S2	Visual inspection	Discrepancy like
S3	Correlation between original image and bi-level image	Goodness like

from discrepancy method group) have been identified according to the performance criteria used. All these classes have been grouped in a table in (Zhang, 2001).

In this study, the table in Zhang (2001) is reproduced as Table 1 with the following supplements and modifications:

1. It would be wonderful if the number of regions segmented from an image corresponding to the real situation (the discrepancy criterion “number of objects in the image” count for this). In case no ground truth were available, it would be expected to get a modest result. Taking this factor into account, a criterion class called “moderate number of regions” is added to Table 1 (see below for their utilization).
2. Among the five discrepancy classes (Zhang, 1996), one includes three miscellaneous criteria. In Table 1, these criteria are separately listed.
3. In (Zhang, 1996), five special methods are discussed. The first three methods, though they cannot fall clearly into goodness group or discrepancy group, are listed in Table 1 as the criteria used in these methods have been employed in recent works (see below). The fourth method has not been justified and the fifth is better counted as an evaluation strategy. So the last two are not included in the table.

MAIN THRUST

Getting into the new century, the research on segmentation evaluation has attracted a lot of attention in the professional community. In the following, some evaluation works (mainly since 2000) are first introduced and discussed and then a summary of these works will be provided. Among these works, some are based on existing techniques, some are

made with modifications / improvements on these techniques and some have introduced new and different approaches.

Evaluation Based on Existing Techniques

In Zhang (1996), a comprehensive survey for early evaluation methods is presented. Since then, many subsequent works have been made based on those methods. Some typical examples in the last few years are discussed below. Note that several of them use the number of mis-segmented pixels as the judging criterion, but often expressed (using classification terms) by different combinative ratios, such as accuracy and precision. These ratios take among: (1) true positive (TP); (2) true negative (TN); (3) false positive (FP), also called miss detect (MD); and (4) false negatives (FN), also called false alarm (FA).

Using both spatial and temporal consistency information, a single objective metric is formed for the segmentation evaluation (Cavallaro, 2002). An objective evaluation of video segmentation quality in spatio-temporal context is proposed. The metric was defined based on objective and perceptual errors. The former mainly considered the number of false pixels, both positively and negatively. The latter one tried to match human perception by assigning weights. The spatial context was introduced to weight the false pixels according to their distance from the reference boundary, while temporal context assigned weights inversely proportional to the duration of an error in order to evaluate the quality variation over time. The overall metric was eventually formulated as a nonlinear combination of the number of false pixels and distances, weighted by the temporal context factor.

A comparative empirical evaluation of representative segmentation algorithms selected from four classes of techniques (two statistical and two deterministic) for detecting moving shadows has been made with a benchmark suite of indoor and outdoor video sequences (Prati, 2003). Two quantitative metrics: good detection (low probability of misclassifying a shadow point) and good discrimination (the probability of classifying non-shadow points as shadow should be low, that is, low false alarm rate) are employed. The first one corresponds to minimize FN (the shadow points classified as background/foreground), while the second one relates to minimize FP (the foreground/background points detected as shadows). A metric based on spatial accuracy and temporal stability that aims at evaluating information differently than the FPs and FNs, depending on their distance from the borders of the mask, and taking into account the shifting (instability) of the mask over time (Villegas, 1999) has also been selected. In addition, two metrics for moving object detection evaluation (Onoguchi, 1998), namely the detection rate (DR) and the false alarm rate (FAR), have also been used with modification. In addition to the above quantitative metrics, the following qualitative measures are also considered: robustness to noise, flexibility to shadow strength, width and shape, object independence, scene independence, computational load and detection of indirect cast shadows and penumbra. The results are voted along the following ranges: “very low,” “low,” “medium,” “high” and “very high.” A comparison on multiple algorithms is given on a pixel-level, providing a frame-by-frame comparison. However, the final result of shadow on the object-level is not considered.

An evaluation of eight different threshold algorithms for shot change detection in a surveillance video has been made by Rosin (2003). Pixel-based evaluation is applied by

using TP, TN, FP and FN, but the authors indicated that this can sometimes give misleading rankings, and some combination of them should be used.

Among a comparison of twelve automatic global thresholding methods (Lievers, 2004), eight are point-dependent algorithms and four are region-dependent algorithms (see Chapter I). Some multimodal images have been tested, wherein the authors defined a cost function for selecting the appropriate thresholds. This cost function is based on intra-class variations, so it is no surprise that the best algorithm found is a minimum cross-entropy method.

A survey of 36 image thresholding methods in total, with a view to assess their performance when applied to remote sensing images and especially in oceanographic applications, has been conducted by Marcello (2004). The algorithms have been categorized into two groups, local and global thresholding techniques. These techniques have been applied to 100 visible, IR (infra-red) and microwave (synthetic aperture radar) remote sensing images. For performance judgment, only visual inspection is carried out.

Four different shadow suppression algorithms have been evaluated, using video from a nightly soccer match with some shadow because of the lighting used (Renno, 2004). The evaluation metrics used are all based on the number of correctly detected pixels on a frame-basis. The metrics included the detection rate, the false positive rate, the signal-to-noise ratio (proportional to the ratio of TP over FP) and the tracking error (the average distance between ground truth boxes and tracked targets). Finally, using an average of all values over time, the performances of shadow segmentation of the four shadow suppression algorithms are compared.

To extract the complex categories and spectrally heterogeneous land cover, high spatial resolution satellite images should be used. A number of segmentation techniques have been proposed. Four of these algorithms (taken from the two main groups of segmentation algorithms—boundary-based and region-based) were compared with a visual segmentation of IKONOS panchromatic images (Carleer, 2004). Two empirical discrepancy evaluation criteria are used: the number of mis-segmented pixels in the segmented images compared with the visually segmented reference images, and the ratio between the number of regions in the segmented image and the number of regions in the reference segmentation.

For 3D images, the segmentation can be either carried out for each sliced 2D image made parallel or rotated, or performed directly by using a volume-based 3D algorithm without slicing. A comparison of the above three segmentation algorithms with four 3D ultrasound images has been conducted (Ladak, 2004). The judging parameter used is the percent difference in volume (volume error) between automatically segmented objects and the manually determined ground truth that was determined by a trained person. The times needed for editing the segmented objects obtained by using the three kinds of algorithms to fit the ground truth are also compared.

Evaluation Made with Modifications / Improvements

To develop new evaluation methods and/or new performance metrics, some modification and improvements to earlier proposed evaluation techniques have been made. According to the classification of evaluation methods, three tracks, corresponding to analytical, empirical goodness and empirical discrepancy methods, can be followed.

Since simple analytical method only is not suitable for a useful evaluation, the related works are grouped in two classes.

1. Using Goodness Criteria

For the quantitative evaluation of the performance segmentation algorithms, a goodness function that does not require any user-set parameter or threshold values has been proposed (Liu, 1994):

$$F(I) = \frac{1}{1000N^2} \sqrt{R} \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}} \quad (1)$$

where I represents the segmented image with size N^2 , in which the total number of regions R , A_i and e_i are the area and the average (color) error (the sum of the Euclidean distances between the values of the pixels of region i and the values of region i in the segmented region) of the i -th region. It is clear that the smaller the value of $F(I)$, the better the segmentation result should be.

The problem associated with equation (1) is that $F(I)$ has a very strong bias toward segmentations with very few regions (the presence of many regions in the segmented image is penalized only by the global measure $R^{1/2}$) unless these regions have very little variation in property values. To solve this problem, substituting the term $R^{1/2}$ in equation (1) with a new term weighting the frequency of regions' size with their respective sizes gives a variation (Borsotti, 1998):

$$F'(I) = \frac{1}{1000N^2} \sqrt{\sum_{A=1}^{\max} [R(A)]^{1+1/A}} \times \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}} \quad (2)$$

where $R(A)$ is the number of regions having exactly area A , and \max , the area of the largest region in the segmented image. The exponent $1+1/A$ enhances the small regions' contribution, so the sum grows as the number of small regions increases.

If the segmentation has lots of small regions, the new weighting term will be a much larger penalty than $R^{1/2}$ in F . Thus for a segmentation resulting in a large number of regions, $F'(I)$ will correctly rank such a result as very poor while $F(I)$ will incorrectly rank it as a good segmentation.

A problem for both $F(I)$ and $F'(I)$ is that they highly penalize segmentations with a large number of regions and only when the squared error in all regions gets very small will a segmentation with more than a few regions be evaluated as best. Thus, a further refinement gives another function (Borsotti, 1998):

$$Q(I) = \frac{1}{1000N^2} \sqrt{R} \times \sum_{i=1}^R \left[\frac{e_i^2}{1 + \log A_i} + \left(\frac{R(A_i)}{A_i} \right)^2 \right] \quad (3)$$

Again $R^{1/2}$ is used to penalize segmentations that have a lot of regions. However, the influence that the $R^{1/2}$ has is greatly reduced by dividing the squared (color) error by

$1 + \log A_i$, which causes this error to have a much bigger influence in $Q(I)$ as compared to its influence in both $F(I)$ and $F'(I)$. So $Q(I)$ has a very strong bias against regions with large area unless there is very little variation.

Further improvement is made by introducing the expected region entropy (Zhang, 2004), which serves in a similar capacity to the term involving the squared (color) error used in equation 1 to equation 3. The expected region entropy of image is simply the expected entropy across all regions where each region has weight (or probability) proportional to its area. Since an over-segmented image will have very small expected region entropy, the expected region entropy must be combined with another term or factor that penalizes segmentations having a large numbers of regions since there would otherwise be a strong bias to over-segment an image. This latter term is layout entropy. The final evaluation measure is the sum of expected region and layout entropies.

Based on the principles of intra-region uniformity and inter-region contrast, three goodness measures to evaluate quantitatively the performance of video object segmentation and tracking methods have been proposed (Erdem, 2004). These measures make use of spatial differences of color and motion along the boundary of the estimated video object plane and temporal differences between the color histogram of the current object plane and its predecessors. They can be used to localize (spatially and/or temporally) regions where segmentation results are good or bad; and/or they can be combined to yield a single numerical measure to indicate the goodness of the boundary segmentation and tracking results over a sequence. The authors show that under certain assumptions, the time-consuming annotation of GT is not necessary. However, when more than segmentation only is required, GT will have to be generated anyway (Desurmont, 2005).

2. Using Discrepancy Criteria

The numbers of pixels correctly segmented and wrongly segmented are important in many evaluation tasks (Zhang, 1996). Instead of using ratios among four quantities: TP, TN, FP and FN, which come from classification, another combinative use of these quantities is to draw a 2D curve in which two axes correspond to two of the above four quantities. Such a curve is called a receiver operating curve (ROC). Note: some use an ROC curve to indicate receiver operating characteristics (ROC) curve.

For evaluation with pixel-based metrics, it has been shown that the obtained ROC curves can be used to extract useful information about the system performance when changing external parameters that describe the conditions of the scene (Oberti, 1999). ROC curves can also be used to find the optimal working point for a set of parameters. In this work, ROC is defined as the curve of FP vs. FN. This work has been extended in Oberti (2000, & 2001). ROC curves have also been used to display the performance of multiple segmentation algorithms (Gao, 2004).

In an empirical study of the performance of five vessel segmentation algorithms using discrepancy methods, ROC curves have also been used (Niemeijer, 2004). The ROC in this study has been defined as a curve of TP vs. FP.

In recognition tasks, three groups of factors, precision, accuracy and efficiency for evaluating related segmentation methods, could be considered in assessing how good a segmentation method is and in comparing it with other methods (Udupa, 2002). Precision factors assess the reliability of the method; accuracy factors describe the validity of the method; efficiency factors determine the human operator time and

computational time required to complete a segmentation task. To characterize the range of behavior of a segmentation method from an accuracy point of view, a delineation operating characteristic (DOC) analysis is proposed (Udupa, 2004). The DOC curve is a curve of TP vs. FP, the same as Niemeijer (2004). The DOC curves have been used for comparing the accuracy of three segmentation methods: thresholding, fuzzy connectedness and fuzzy c-means (Udupa, 2004).

Recently, a perturbation detection rate (PDR) analysis has been proposed (Kim, 2004). It measures the sensitivity of a background subtraction algorithm in detecting low contrast targets against background. Four background subtraction algorithms are evaluated for their segmentation performance. The study showed PDR analysis has some advantage over ROC analysis, but the PDR analysis does not consider detection rates through the video frame or over time.

There is often more than one object in the image to be segmented, so two types of metrics (per-region and inter-region) can be defined for regions in an image. Per-region metrics are used for individual objects, one example is the sum of three metrics (Correia, 2000a): spatial accuracy, temporal accuracy (ITU, 1996) and spatio-temporal accuracy (Wolf, 1997). The overall metric for one object was the weighted average of all the items in the three metrics. To weight the importance of objects, an object relevance metric is introduced (Correia, 2003b). For overall segmentation quality evaluation purposes, the relevance of a region must be evaluated taking into account the context where it is found. The contextual relevance metric reflects the importance of a region in terms of the human visual system (HVS), and can be computed by the combination of a set of metrics expressing the features which are able to capture the viewer's attention (Correia, 2000b).

Another study used four metrics for evaluation of video segmentation (Li, 2003). They are metrics for contour-based spatial matching, temporal consistency, user workload and time consumption. The first two metrics can be considered as combined extensions of the number and position of mis-segmented pixels. The last two metrics are somewhat closely related.

New Ideas Emerged from Evaluation

In recent evaluations, some novel ideas, compared to early ones surveyed in Zhang (1996), have made an appearance. Following are some representative examples.

The performance of segmentation algorithms is influenced by many factors. Since one metric would be not enough to judge all properties of segmentation algorithms, different methods, especially different evaluation metrics, should be combined. One early work of this kind was made a quarter of a century ago (Yasnoff, 1979). A number of works have followed this idea; see references in Zhang (1996). To avoid merging multiple metrics which might have different, or even incompatible, objectives into a single metric which is hardly comprehensive, an approach to formulate the evaluation problem as determining the Pareto front in a multidimensional fitness space is proposed (Everingham, 2002).

The above idea is followed in a multi-metric evaluation protocol for evaluating the performance of user-assisted video object extraction systems (Li, 2003). With four different evaluation metrics, one 4D fitness space has been built and a search in multidimensional space is performed to find the best choice of a system with optimal parameters. Though the orthogonality of a multi-metric was not easy to verify, the value along each axis has been determined separately.

Another cooperative framework is also proposed in which different effectiveness measures judge the performance of the segmentation in different ways, and these measures are combined by using a machine-learning approach which combines the results obtained from different measures (Zhang, 2005). Three strategies based on training for combinations are used: Weighted majority (WM), Bayesian and support vector machine (SVM). For this study, five evaluation criteria are taken. With 199 image samples, 108 are taken as a training set and 91 are taken as an evaluation set. Using different strategies, the combination can be considered to provide three new evaluation criteria. The comparison of these three new criteria with their composing (base) criteria is conducted as follows. For one image, two segmentations are made by human and machine, respectively. The experimenter must make sure that the human segmentation looks clearly better than machine segmentation, and then, using different evaluation criteria to verify the segmentation results, count the number of the consistencies. The better criteria should provide a higher score. Experimental results showed that combined criteria yield better results than any of the single criterion.

Video has one more dimension compared with that of still image, and thus the segmentation of video frequently consists of extracting a set of objects from a number of frames. In this regard, the evaluation of video segmentation quality could have two targets (Correia, 2003a, 2003b): (1) Individual object segmentation quality evaluation in which a single object identified by the segmentation algorithm is independently evaluated in terms of its segmentation quality. (2) Overall segmentation quality evaluation in which the complete set of objects (for the whole scene) identified by the segmentation algorithm is globally evaluated in terms of its segmentation quality.

An outlined framework for performance evaluation of a video content analysis (VCA) system (in a particular project for traffic and surveillance applications in which segmentation is the first step) is proposed (Desurmont, 2005). Four main components of this framework are: (1) creation of ground-truth (GT) data; (2) available evaluation of data sets; (3) performance metrics; and (4) presentation of the evaluation results. For reasons of effectiveness and straightforwardness, the evaluation should be performed in different semantic levels: (1) pixel-level; (2) object-based evaluation per frame; (3) object-based evaluation over an object's lifetime; (4) object-features level; and (5) behavior of objects. On the other hand, matching the descriptions from VCA with that of GT is still an open research topic to be explored, as such evaluation results may not necessarily represent the true performance due to the complexity of matching objects, events or behaviors in videos.

Statistics and Classification

Segmentation evaluation is a complex task that involves a number of process stages and has many aspects. For example, a five-step procedure to quantitatively assess a segmentation algorithm by using empirical performance evaluation methodology is proposed (Mao, 2001). It includes: (1) creating test data set with ground-truth; (2) formulating evaluation metrics; (3) choosing optimal free parameters of the algorithm if any; (4) evaluating the algorithm according to metric values; and (5) calculating the statistical significance of the above evaluation. Similarly, three essential elements in a performance evaluation protocol: ground truth acquisition, matching procedure and quantitative metrics definition have been identified (Liu, 1999).

Above all, a meaningful and computable evaluation criterion is essential in the whole evaluation methodology. Based on the above review for recent evaluation works, the criteria used in these works are summarized in the following tables (only empirical criteria are considered). To make these tables more comprehensive, those criteria listed in Zhang (2001) are also included. Thus, these tables show a general "image" of segmentation evaluation in the last 10 years.

Table 2 gives a list of evaluation works using existing techniques. Most works are based on discrepancy criteria, in which the criteria belonging to class D-1 appears more frequently than others. Still few works used goodness criteria, but the criterion in class G-3 has not been used. On the other hand, newly defined classes G-4 and D-5a have been exploited by some works.

Table 3 gives a list of evaluation works with modification to existing techniques. Most works are still based on discrepancy criteria, in particular on class D-1. Several works made use of the combination of criteria from different classes. Newly defined class S-1 has also been used by two works.

Table 4 gives a list of evaluation works with some novelties. The first three consider the problem of combining different criteria.

Table 2. A list of evaluation works using existing techniques

Method #	Source	Criteria used	Method #	Source	Criteria used
M-1	(Hoover, 1996)	D-5a	M-10	(Huo, 2000)	D-1, D-4
M-2	(Zhang, 1997)	D-4	M-11	(Cavallaro, 2002)	D-1, D-2
M-3	(Borsotti, 1998)	G-1, G-2, G-4	M-12	(Prati, 2003)	D-1
M-4	(Xu, 1998)	S-3	M-13	(Rosin, 2003)	D-1
M-5	(Chang, 1999)	D-5a	M-14	(Lievers, 2004)	G-1
M-6	(Yang, 1999)	D-1	M-15	(Marcello, 2004)	S-2
M-7	(Mattana, 1999)	D-4	M-16	(Renno, 2004)	D-1, D-4
M-8	(Rosenberger, 2000)	G-1, G-2	M-17	(Carleer, 2004)	D-1, D-3
M-9	(Betanzos, 2000)	D-1	M-18	(Ladak, 2004)	D-1, S-1

Table 3. A list of evaluation works with modified criteria

Method #	Source	Criteria used (modification)
M-19	(Oberti, 1999)	D-1 (ROC, curve of FP vs. FN)
M-20	(Correia, 2000a)	D-1 (with spatial and temporal extension)
M-21	(Gao, 2000)	D-1 (ROC, curve of FP vs. FN)
M-22	(Udupa, 2002)	D-1, S-1 like (efficiency)
M-23	(Li, 2003)	D-1 and D-2, (contour matching, temporal consistency), S-1
M-24	(Zhang, 2004)	G-1, G-2, G-4 (using region entropy)
M-25	(Erdem, 2004)	G-1, G-2 (with extension to color, motion, color histograms)
M-26	(Niemeijer, 2004)	D-1 (ROC, curve of TP vs. FP)
M-27	(Udupa, 2004)	D-1 (DOC, curve of TP vs. FP)
M-28	(Kim, 2004)	D-1 (PDR, modified detection rate)

Table 4. A list of evaluation works with novelties

Method #	Source	Novelty
M-29	(Everingham, 2002)	Finding out the Pareto front in a multi-dimensional fitness space
M-30	(Li, 2003)	Finding out the Pareto front in a 4-D fitness space
M-31	(Zhang, 2005)	Using weighted majority (WM), Bayesian and support vector machine (SVM)
M-32	(Correia, 2003a)	Using contextual relevance metric to match human visual system (HVS)
M-33	(Desurmont, 2005)	Performing evaluation in different semantic levels

Table 5. Comparison of methods in Table 2 and Table 3

Method #	Generality	Complexity	Method #	Generality	Complexity
M-1	General	Medium	M-15	General	High (Human) ⁶
M-2	General	Medium	M-16	General	Med./High
M-3	Numerous objects ¹	Medium/High	M-17	Numerous objects ¹	Low/Medium
M-4	Tree structure ²	High	M-18	General	High (Human) ⁶
M-5	Particular ³	Medium	M-19	General	Medium
M-6	General	Medium	M-20	General	Medium/High
M-7	General	Low/Medium	M-21	General	Medium
M-8	General	Medium/High	M-22	General	Medium
M-9	General	Medium	M-23	General	High (Human) ⁶
M-10	General	Medium	M-24	Numerous objects ¹	Medium
M-11	Video ⁴	Medium/High	M-25	Video ⁴	High
M-12	General	High	M-26	General	Medium
M-13	Video ⁴	Medium	M-27	General	Medium
M-14	Thresholding ⁵	Medium	M-28	Video ⁴	Medium

Note:

1. More appropriate for treating images composed of numerous object regions;
2. Only applicable for segmentation algorithms with tree structure;
3. In conditions that the five categories of regions defined by authors could be suitably determined;
4. Only usable for video segmentation evaluation (mostly because temporal factor is critical);
5. Only suitable for evaluating thresholding technique; and
6. Human visual factors are involved, so the complexity becomes high.

To compare different methods for segmentation evaluation, the following four factors are considered, taking into consideration the techniques and measures used in evaluation (Zhang, 1993, 2001): (1) generality for evaluation; (2) subjective versus objective and qualitative versus quantitative; (3) complexity for evaluation; and (4) evaluation requirements for reference images.

Among the above four factors, some of them are related to the method groups. For example, most empirical criteria provide quantitative results. On the other hand, the subjective versus objective and the consideration of segmentation application are closely related and can be determined according to either the criteria belonging to goodness or discrepancy group. As only empirical methods are compared here, the focus

will be put on the generality and the complexity for evaluation. Such obtained comparison results for the methods listed in Table 2 and Table 3 are given in Table 5.

FUTURE TRENDS

Though various progresses have been made recently, it seems that the results obtained in the field of segmentation evaluation are still far from satisfactory. A number of factors still limit the advancements of segmentation evaluation and, in turn, the performance improvements of segmentation algorithms:

1. There is no common mathematical model or general strategy for evaluation.
2. It is difficult to define wide-ranging performance metrics and statistics.
3. The testing data used in evaluation are often not representative enough for actual application.
4. Appropriate ground truths are hard to determine objectively.
5. Often large costs (both time and effort) are involved in performing comprehensive evaluations.

Some urgent and important research directions should be:

1. *Make evaluation in considering the final goal of segmentation.*
The purpose of segmentation, especially the goals of analysis tasks, should be considered in segmentation evaluation (Zhang, 1992a). A number of works in this direction have been made (see Tables 2 and 3). One sample indication is “a good segmentation evaluation method would not only enable different approaches to be compared, but could also be integrated within the target recognition system to adaptively select the appropriate granularity of the segmentation which in turn could improve the recognition accuracy” (Zhang, 2005). Segmentation is not an isolated process, the success of its following processes would be an indication of its quality.
2. *Combine multiple metrics efficiently.*
Using several criteria for evaluation was considered very early by Yasnoff (1979); a number of works have followed this initiative and recent works (see Table 4) also show this tendency. The tactic for combining multiple metrics has evolved from simply making weighted sums to complicated machine learning techniques. Further works to combine multiple metrics to cover different aspects of an algorithm’s performance in wide ranges of applications are still needed.
3. *Construct common databases for segmentation evaluation.*
The problems of evaluation related to testing data and ground truth are often caused by the lack of common databases. Generating synthetic images is relatively simple to implement (Zhang, 1992b). How to design realistic images is a critical factor. Recording real images and segmenting them manually can also provide a solution for constructing image databases with ground truths, which may need considerable labor. Some primary works in this direction have been made (Hoover,

1996; Martin, 2001; Niemeijer, 2004); additional efforts should be put toward more general databases for segmentation evaluation tasks.

CONCLUSIONS

A number of evaluation works are summarized in this chapter. Based on a general framework for segmentation and its evaluation, a discussion on terminology and taxonomy of segmentation evaluation is first made. Then, recent progress in segmentation evaluation is examined, which is focused on the principles and the criteria used in different studies.

It seems that though much effort has been put in this subject recently, no (or very few) radical progress has been reported. Many evaluation works used previously proposed methods and criteria for particular applications; some evaluation works made improvements on early works by using similar principles. To probe further, some existing problems and several future directions for segmentation evaluation are indicated in this chapter.

ACKNOWLEDGMENT

This work has been supported by the grants NNSF-60573148 and RFDP-20020003011.

REFERENCES

- Betanzos, A., Varela, A., & Martínez, C. (2000). Analysis and evaluation of hard and fuzzy clustering segmentation techniques in burned patient images. *Image and Vision Computing*, 18(13), 1045-1054.
- Borsotti, M., Campadelli, P., & Schettini, R. (1998). Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8), 741-747.
- Carleer, A. P., Debeir, O., & Wolff, E. (2004). Comparison of very high spatial resolution satellite image segmentations. *SPIE*, 5238, 532-542.
- Cavallaro, A., Gelasca, E. D., & Ebrahimi, T. (2002). Objective evaluation of segmentation quality using spatio-temporal context. In *Proceedings of the International Conference on Image Processing* (Vol. 3, pp. 301-304).
- Chang, K. I., Bowyer, K. W., & Sivagurunath, M. (1999). Evaluation of texture segmentation algorithms. In *Proceedings of the Computer Vision and Pattern Recognition* (Vol. 1, pp. 294-299).
- Correia, P. L., Pereira, F. M. B. (2000a). Objective evaluation of relative segmentation quality. In *Proceedings of the International Conference on Image Processing* (Vol. 1, pp. 308-311).
- Correia, P. L., & Pereira, F. M. B. (2000b). Estimation of video object's relevance. In *Proceedings of the EUSIPCO* (pp. 925-928).
- Correia, P. L., & Pereira, F. M. B. (2003a). Methodologies for objective evaluation of video segmentation quality. *SPIE*, 5150, 1594-1600.

- Correia, P. L., & Pereira, F. M. B. (2003b). Objective evaluation of video segmentation quality. *IEEE Image Processing*, 12(2), 186-200.
- Desurmont, X., Wijnhoven, R., & Jaspers, E., et al. (2005). Performance evaluation of real-time video content analysis systems in the CANDELA project. *SPIE*, 5671, 200-211.
- Erdem, C. E., Sankur, B., & Tekalp, A. M. (2004). Performance measures for video object segmentation and tracking. *IEEE Image Processing*, 13(7), 937-951.
- Everingham, M., Muller, H., & Thomas, B. (2002). Evaluating image segmentation algorithms using the Pareto Front. In *Proceedings of the European Conference on Computer Vision* (Vol. 4, pp. 34-48).
- Gao, X., Boult, T. E., & Coetzee, F., et al. (2000). Error analysis of background adaptation. In *Proceedings of the Computer Vision and Pattern Recognition* (Vol. 1, pp. 503-510).
- Hoover, A., Jean-Baptiste, G., & Jiang, X., et al. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Pattern Analysis and Machine Intelligence*, 18(7), 673-689.
- Huo, Z., & Giger, M. L. (2000). Evaluation of an automated segmentation method based on performances of an automated classification method. *SPIE*, 398, 16-21.
- ITU-T. (1996). Recommendation P.910: Subjective video quality assessment methods for multimedia applications.
- Kim, K., Chalidabhongse, T. H., & Harwood, D., et al. (2004). Background modeling and subtraction by codebook construction. In *Proceedings of the International Conference on Image Processing* (Vol. 5, pp. 3061-3064).
- Ladak, H. M., Ding, M., Wang, Y., et al. (2004). Evaluation of algorithms for segmentation of the prostate boundary from 3D ultrasound images. *SPIE*, 5370, 1403-1410.
- Li, N., Li, S., & Chen, C. (2003). Multimetric evaluation protocol for user-assisted video object extraction systems. *SPIE*, 5150, 20-28.
- Lievers, W. B., & Pilkey, A. K. (2004). An evaluation of global thresholding techniques for the automatic image segmentation of automotive aluminum sheet alloys. *Materials Science and Engineering*, A. 381(1-2), 134-142.
- Liu, J., & Yang, Y.-H. (1994). Multiresolution color image segmentation. *IEEE Pattern Analysis and Machine Intelligence*, 16(7), 689-700.
- Liu, W., & Dori, D. (1999). Principles of constructing a performance evaluation protocol for graphics recognition algorithms. In *Performance characterization and evaluation of computer vision algorithms* (pp. 97-106).
- Mao, S., & Kanungo, T. (2001). Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Pattern Analysis and Machine Intelligence*, 23(3), 242-256.
- Marcello, J., Marques, F., & Eugenio, F. (2004). Evaluation of thresholding techniques applied to oceanographic remote sensing imagery. *SPIE*, 5573, 96-103.
- Martin, D., Fowlkes, C., & Tal, D., et al. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference on Computer Vision* (Vol. 2, pp. 416-423).
- Mattana, M. F., Facon, J., & Britto, A. S. (1999). Evaluation by recognition of thresholding-based segmentation techniques on Brazilian bankchecks. *SPIE*, 3572, 344-348.

- Moretti, B., Fadili, J. M., & Ruan, S., et al. (2000). Phantom-based performance evaluation: Application to brain segmentation from magnetic resonance images. *Medical Image Analysis*, 4(4), 303-316.
- Niemeijer, M., Staal, J., & Ginneken, B., et al. (2004). Comparative study of retinal vessel segmentation methods on a new publicly available database. *SPIE*, 5370, 648-656.
- Oberti, F., Granelli, F., & Regazzoni, C.S. (2000). Minimax based regulation of change detection threshold in video surveillance systems. In G. L. Foresti, P. Mähönen, and C. S. Regazzoni (Eds.). *Multimedia video-based surveillance systems* (pp. 210-233). Kluwer Academic Publishers.
- Oberti, F., Stringa, E., & Vernazza, G. (2001). Performance evaluation criterion for characterizing video-surveillance systems. *Real-Time Imaging*, 7(5), 457-471.
- Oberti, F., Teschioni, A., & Regazzoni, C. S. (1999). ROC curves for performance evaluation of video sequences processing systems for surveillance applications. In *Proceedings of the International Conference on Image Processing* (Vol. 2, pp. 949-953).
- Onoguchi, K. (1998). Shadow elimination method for moving object detection. In *Proceedings of the International Conference on Pattern Recognition* (Vol. 1, pp. 583-587).
- Prati, A., Mikic, I., & Trivedi, M. M., et al. (2003). Detecting moving shadows: Algorithms and evaluation. *IEEE Pattern Analysis and Machine Intelligence*, 25(7), 918-923.
- Renno, J. R., Orwell, J., & Jones, G. A. (2004). Evaluation of shadow classification techniques for object detection and tracking. In *Proceedings of the IEEE International Conference on Image Processing* (pp. 143-146).
- Rosenberger, C., & Chehdi, K. (2000). Genetic fusion: Application to multi-components image segmentation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (Vol. 4, pp. 2223-2226).
- Rosin, P. L., & Ioannidis, E. (2003). Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14), 2345-2356.
- Udupa, J. K., LeBlanc, V. R., & Schmidt, H., et al. (2002). A methodology for evaluating image segmentation algorithms. *SPIE*, 4684, 266-277.
- Udupa, J., & Zhuge, Y. (2004). Delineation operating characteristic (DOC) curve for assessing the accuracy behavior of image segmentation algorithms. *SPIE*, 5370, 640-647.
- Villegas, P., Marichal, X., & Salcedo, A. (1999). Objective evaluation of segmentation masks in video sequences. In *Proceedings of the Workshop Image Analysis for Multimedia Interactive Services* (pp. 85-88).
- Wolf, S., & Webster, A. (1997). Subjective and objective measures of scene criticality. In ITU-T Study Groups 9 and 12, and ITU-R Study Group 11, *ITU experts meeting on subjective and objective audiovisual quality assessment methods*. Geneva, Switzerland, ITU, SG 12 document number JRG010.
- Xu, Y., Olman, V., & Uberbacher, E. C. (1998). A segmentation algorithm for noisy images: Design and evaluation. *Pattern Recognition Letters*, 19(13), 1213-1224.
- Yang, J., & Huang, S. C. (1999). Method for evaluation of different MRI segmentation approaches. *IEEE Network Simulator*, 46(6), 2259-2265.
- Yasnoff, W. A., Galbraith, W., & Bacus, J. W. (1979). Error measures for objective assessment of scene segmentation algorithms. *AQC*, 1, 107-121.

- Zhang, H., Fritts, J. E., & Goldman, S. (2004). An entropy-based objective evaluation method for image segmentation. *SPIE*, 5307, 38-49.
- Zhang, H., Fritts, J. E., & Goldman, S. A. (2005). A co-evaluation framework for improving segmentation evaluation. *SPIE*, 5809, 420-430.
- Zhang, Y. J. (1993). Comparison of segmentation evaluation criteria. In *Proceedings of the 2nd International Conference on Signal Processing* (Vol. 1, pp. 870-873).
- Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335-1346.
- Zhang, Y. J. (1997). Evaluation and comparison of different segmentation algorithms. *PRL*, 18(10), 963-974.
- Zhang, Y. J. (2001). A review of recent evaluation methods for image segmentation. In *Proceedings of the 6th International Symposium on Signal Processing and Its Applications* (pp.148-151).
- Zhang, Y. J. (2006). A study of image engineering. In M. Khosrow-Pour (Ed.), *Encyclopedia of information science and technology* (2nd ed.) [online].
- Zhang, Y. J., & Gerbrands, J. J. (1992a). Segmentation evaluation using ultimate measurement accuracy. *SPIE*, 1657, 449-460.
- Zhang, Y. J., & Gerbrands, J. J. (1992b). On the design of test images for segmentation evaluation. In *Proceedings of the EUSIPCO* (Vol. 1, pp. 551-554).
- Zhang, Y. J., & Gerbrands, J. J. (1994). Objective and quantitative segmentation evaluation and comparison. *Signal Processing*, 39(3), 43-54.

About the Authors

Yu-Jin Zhang (PhD, State University of Liège, Belgium) is a professor of image engineering at Tsinghua University, Beijing, China. Previously, he was with the Delft University of Technology, The Netherlands. He spent a year as a visiting professor at National Technological University, Singapore. His research interests are mainly in the area of image engineering, including: image processing, image analysis and image understanding, as well as their applications. He has published more than 200 research papers and 10 books including two monographs: *Image Segmentation and Content-based Visual Information Retrieval* (Science Press). He is editor of *Semantic-Based Visual Information Retrieval* (IRM Press, Idea Group Inc.). He is vice president of China Society of Image and Graphics and the director of the academic committee of the society. He is deputy editor-in-chief of the *Journal of Image and Graphics* and associate editor of *Pattern Recognition Letters*. He was program co-chair of The First International Conference on Image and Graphics (ICIG 2000) and The Second International Conference on Image and Graphics (ICIG 2002). He is a senior member of IEEE.

* * * *

Mongi Abidi is a W. Fulton professor with the Department of Electrical and Computer Engineering at the University of Tennessee, Knoxville (USA), which he joined in 1986. Dr. Abidi received his MS and PhD degrees in electrical engineering from the University of Tennessee. His interests include image processing, multi-sensor processing, three-dimensional imaging and robotics. He has published over 120 papers in these areas and co-edited the book *Data Fusion in Robotics and Machine Intelligence* (Academic Press, 1992). He is the recipient of the 1994-95 Chancellor's Award for Excellence in Research and Creative Achievement and the 2001 Brooks Distinguished Professor Award. He is a member of IEEE, Computer Society, Pattern Recognition Society, SPIE, Tau Beta Pi, Phi Kappa Phi and Eta Kappa Nu honor societies. He also received the First Presidential Principal Engineer Award prior to joining the University of Tennessee.

Aichouche Belhadj-Aissa received a bachelor's degree in electronic engineering from the Ecole Nationale Polytechnique, Algiers, and a master's degree and PhD in image processing from the Electronic Institute of U.S.T.H.B. University, Algeria. She is currently a professor of image processing, remote sensing and S.I.G. at the U.S.T.H.B and the head of the team "GIS and Geo-Referencing Data Fusion," which is responsible for signal and image processing. Her research interests include satellite image analysis, texture and shape analysis and modelling, fusion and classification of objects, radar SAR interferometry and polarimetry, S.I.G.

Giovanni Bellettini received a PhD in functional analysis and applications from the International School for Advanced Studies of Trieste. He is full professor of mathematical analysis of the engineering faculty of the University of Roma, "Tor Vergata." His research interests include semi continuity and approximation problems in minimal surface theory and calculus of variations, with applications to phase transitions, statistical mechanics and image segmentation. He has worked in the area of geometric evolution problems, including mean and crystalline mean curvature flows. He is currently working in the fields of partial differential equations and mathematical physics.

Sébastien Chabrier is a PhD student at ENSI, France. He obtained his master's degree from the University of Clermont Ferrand. He belongs to the Laboratory of Vision and Robotics of Bourges in the Signal, Image and Vision research unit. His research interest concerns essentially image segmentation evaluation.

Paulo Lobato Correia is an assistant professor at the Electrical and Computer Engineering Department, Instituto Superior Técnico, Technical University of Lisbon, Portugal, and a researcher at the Image Group of Instituto de Telecomunicações. He has participated in several national and international research projects, being the national representative for the European COST 292 project. He is a member of the *EURASIP/Elsevier Signal Processing Journal* editorial board and an elected member of the EURASIP Administrative Committee. His current research interests include video analysis and processing, namely video segmentation, objective video segmentation quality evaluation, content-based video description and biometric recognition.

Shengyang Dai is a PhD student in the Electrical and Computer Engineering Department of Northwestern University (USA). He is interested in various topic of computer vision, including motion detection, tracking, image segmentation, content-based retrieval and machine learning. He received his BS and MS in electrical engineering from Tsinghua University, Beijing, China.

Abraham Duarte received his BE in physics (electronics specialty) from the Complutense University of Madrid and his PhD from the Rey Juan Carlos University, Spain. He is an assistant professor in the Department of Computer Science, Statistics and Telematics at Rey Juan Carlos University. He is a reviewer for several journals and belongs to the program committee of several conferences. He is member of HEUR and EU/ME metaheuristic networks. He has published more than 20 papers in international journals and conferences and has participated in several research projects. His main research interest

focuses on the development of solution procedures for optimization problems, specifically, metaheuristics applied to computer vision and combinatorial problems.

Bruno Emile is an assistant professor at IUT of Châteauroux, France. He obtained his PhD from the University of Nice. He belongs to the Laboratory of Vision and Robotics of Bourges in the Signal, Image and Vision research unit. His research interests concern unsupervised segmentation evaluation and texture analysis. He also works on motion estimation and road tracking.

Francisco Escolano received a BS in computer science from the Polytechnic University of Valencia, Spain, and a PhD in computer science from the University of Alicante, Spain. He is an associate professor with the Department of Computer Science and Artificial Intelligence of the University of Alicante. He has been a postdoctoral fellow with Dr. Norberto M. Grzywacz at the Biomedical Engineering Department, University of Southern California, and he has also collaborated with Dr. Alan Yuille at the Smith-Kettlewell Eye Research Institute, USA. Recently, he has visited the Liisa Holm's Bioinformatics Lab at the University of Helsinki. He is the head of the Robot Vision Group of the University of Alicante, whose research interests are focused on the development of efficient and reliable computer-vision algorithms for biomedical applications, active vision and robotics and video-analysis.

Minoru Etoh received his BE and MSEE from Hiroshima University and his PhD from Osaka. He led an image communication research team in Matsushita Electric and participated in MPEG standardization. He joined NTT DoCoMo (Japan) and was appointed CEO of DoCoMo Communications Laboratories USA, Inc., where he was conducting several research groups in mobile network architecture, terminal software and mobile multimedia technologies. He is now at NTT DoCoMo as managing director of multimedia laboratories. He is also a visiting professor at Nara Institute of Science and Technology.

Felipe Fernandez is a teaching professor with the faculty of computer science, Universidad Politécnica of Madrid, Spain. He received a MSc in electronic engineering and a PhD in computer science. He was an external consultant at the Institute National de Recherche en Informatique et en Automatique (INRIA). He has collaborated on different industrial and research projects sponsored by UPM, MRS, MEC, CEE, CICYT, MCT, CAM and EU. He has co-authored seven books and has published more than 60 papers in international journals and conferences. He has worked as application engineer at company Omron Electronics Corp., and he currently works at multinational Robert Bosch GmbH as a development engineer. His current research interests lie in the area of hardware-software design, computer parallel architectures and computational intelligence.

Irene Yu-Hua Gu received a PhD in electrical engineering from Eindhoven University of Technology, The Netherlands. She is a professor in the Department of Signals and Systems, Chalmers University of Technology, Sweden. Before joining Chalmers, she was a research fellow at Philips Research Institute IPO (NL) and Staffordshire University, UK, and a lecturer at the University of Birmingham, UK. Her current research interests include

image processing, video segmentation and surveillance, object tracking, video communications and signal processing with applications to electric power systems. She has been the author/co-author of approximately 80 publications in refereed journals and conferences, and co-authored one book. Dr. Gu has served as an associate editor for the *IEEE Transactions on Systems, Man and Cybernetics*, the Chair of the *Signal Processing Chapter of IEEE Swedish Section*, and is a member of the editorial board for the *EURASIP journal on Applied Signal Processing*. Dr. Gu is a senior member of the IEEE:

Vasile Gui received an engineering degree in electronics, Technical University of Timisoara (Romania), and a PhD in measurements. He lectures in electronics, signal processing, machine vision and computer graphics as faculty of electronics and telecommunications, Technical University of Timisoara, where he is a professor. He published several research papers and two books. He holds two patents. His research interests include image filtering, segmentation and analysis, motion estimation, multimedia, biomedical and industrial applications of image processing.

Sangkyu Kang received a BA in electrical engineering and a MS in image engineering from Chung-Ang University, Korea. His research interests include image restoration for motion blur removal, automatic shape extraction and analysis, HCI and real-time object tracking for a smart surveillance system. He is currently a senior research engineer at Digital Media Research Laboratory, LG Electronics, Korea, developing multimedia solutions for mobile platforms.

Osamu Katai received his BE, ME and PhD degrees from Kyoto University (Japan) where he has been, and continues as, a professor in the Department of Systems Science, Graduate School of Informatics. From 1980 to 1981, he had been a visiting researcher at the National Research Institute for Information Science and Automation, France. His current research interests are on the harmonious symbiosis of artificial systems with natural systems, including ecological design, ecological interface design, environmental spatial design, community design, etc.

Hiroshi Kawakami received BE, ME and PhD degrees from Kyoto University (Japan). He has been an instructor with the Department of Information Technology, Faculty of Engineering, Okayama University. He has also been an associate professor with the Department of Systems Science, Graduate School of Informatics, Kyoto University. His current research interests are on ecological and emergent system design, co-operative synthesis methods and knowledge engineering.

Assia Kourgli, PhD, received a bachelor's degree in electronic engineering and a master's degree in image processing from the Electronic Institute of U.S.T.H.B. University, Algeria. She is currently a lecturer of C++ programming at the U.S.T.H.B. University. Her research interests include satellite image analysis, texture analysis, image segmentation and classification, parametric and non-parametric texture synthesis and modelling, multiscale processing.

Wing Hong Lau received BSc and PhD degrees in electrical and electronic engineering from the University of Portsmouth. He is currently an associate professor in the

Department of Electronic Engineering, City University of Hong Kong. His current research interests are in the area of digital signal processing, digital audio engineering and visual speech signal processing. Dr. Lau is currently the chairman of the IEEE Hong Kong Section. He is the financial chair of the TENCON 2006. Dr. Lau was the recipient of the IEEE Third Millennium Medal. He was the registration co-chair of the ICASSP 2003 and ISCAS 1997. He was the chairman of the IEEE Hong Kong Joint Chapter on CAS/COM for 1997 and 1998.

Hélène Laurent is an assistant professor at ENSI, France. She obtained her PhD from the University of Nantes. She belongs to the Laboratory of Vision and Robotics of Bourges in the Signal, Image and Vision research unit. Her research interests concern supervised segmentation evaluation and object recognition. She also worked on the use of time frequency representation for segmentation applications and diagnosis.

Shu Hung Leung received his first class honor BSc degree in electronics from the Chinese University of Hong Kong and his MSc and PhD degrees, both in electrical engineering, from the University of California, USA. He was an assistant professor with the University of Colorado, USA. Currently, he is an associate professor in the Department of Electronic Engineering at City University of Hong Kong. Dr. Leung is also the leader of the Digital and Mobile Communication Team, Department of Electronic Engineering. His current research interests are in digital communications, speech and image processing, intelligent signal processing and adaptive signal processing. He has served as the program chairman of the Signal Processing Chapter of the IEEE Hong Kong Section and now as the chairman of the chapter. He serves as a technical reviewer for a number of international conferences and IEEE Transactions, IEE proceedings and Electronics Letters. He is listed in the Marquis Who's Who in Science and Engineering and the Marquis Who's Who in the World.

Zhenyan Li received both BEng and MEng degrees in electrical and communication engineering from Harbin Institute of Technology, P.R. China. She is currently pursuing a PhD in electrical and electronic engineering at Nanyang Technological University, Singapore. Her research interests include content-based video analysis, pattern recognition and computer vision.

Alan Wee-Chung Liew received his BEng with first class honors in electrical and electronic engineering from the University of Auckland, New Zealand and his PhD in electronic engineering from the University of Tasmania, Australia. He worked as a research fellow and later as a senior research fellow at the Department of Electronic Engineering and the Department of Computer Engineering and Information Technology, respectively, both at the City University of Hong Kong. He is currently an assistant professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong. His current research interests include bioinformatics, medical imaging and computer vision and pattern recognition. He has served as a technical committee member in a number of international conferences and as a technical reviewer for a number of journals in IEEE Transactions, IEE proceedings, bioinformatics and computational biology. Dr. Liew is a member of the Institute of Electrical and Electronic Engineers (IEEE),

and his biography is listed in the Marquis Who's Who in the World and Marquis Who's Who in Science and Engineering.

Weisi Lin graduated from Zhongshan University, China with a BSc and MSc, and from King's College, London University, with a PhD. He has worked in Zhongshan University, Shantou University, China, Bath University, UK, National University of Singapore, Institute of Microelectronics, Singapore, and Centre for Signal Processing, Singapore. He has led over 10 successful industrial-funded projects in digital multimedia and published 70 refereed papers in international journals and conferences. He is currently the lab head of visual processing at the Institute for Infocomm Research, Singapore. His current research interests include image processing, visual distortion metrics and perceptual video compression.

Miguel A. Lozano received a BS in computer science from the University Alicante, Spain. He has been a teaching assistant with the Department of Computer Science and Artificial Intelligence, University of Alicante. He has visited Edwin Hancock's Computer Vision & Pattern Recognition Lab at the University of York, and more recently, Liisa Holm's Bioinformatics Lab at the University of Helsinki. He is a member of the Robot Vision Group of the University of Alicante and his research interests include computer vision and robotics.

Hong Lu received BEng and MEng degrees in computer science and technology from Xidian University, P.R. China, and a PhD from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. She worked as a lecturer and researcher at the School of Computer Science and Technology, Xidian University, P. R. China. She also worked as a research student in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Currently, she is a lecturer and researcher in the Department of Computer Science and Engineering, Fudan University, P.R. China. Her research interests include image and video processing, pattern recognition and computer vision.

Riccardo March received the laurea degree in electronic engineering cum laude from the University of Pisa. He joined the Italian National Research Council where he is a senior researcher in applied mathematics. His research interests are variational methods in computer vision, free discontinuity problems and the variational approximation of functional. He is also interested in the application of the theory of Pade's approximation to super-resolution in signal processing.

Pablo Martinez is a professor of computer science at the University of Extremadura, Spain. He is the head scientist of the Neural Networks and Signal Processing Group (GRNPS). He has held visiting researcher positions at NASA Goddard Space Flight Center and the Department of Electrical Engineering, University of Maryland, USA. His main research interests include remote sensing, digital image analysis, hardware-based architectures, operating systems management and configuration and neural network-based pattern recognition.

Domingo Mery received a MSc in electrical engineering, Technical University of Karlsruhe, Germany, and a PhD with distinction, Technical University of Berlin. He was a research assistant at the Department of Electrical Engineering at the Catholic University of Chile. He was a research scientist at the Institute for Measurement and Automation Technology at the Technical University of Berlin with the collaboration of YXLON X-Ray International. He was a recipient of a scholarship from the Konrad-Adenauer-Foundation, and also a recipient of a scholarship from the German Academic Exchange Service (DAAD) for his PhD work. He was a research associate at the Department of Computer Engineering at the University of Santiago, Chile. He has been an associated professor at the Department of Computer Science at the Pontificia Universidad Católica de Chile. Currently, he is the chair of the Computer Science Department. His research interests include image processing for fault detection in aluminum castings, image processing in food industry, x-ray imaging, real-time programming and computer vision.

Antonio S. Montemayor received a BE in applied physics from the Universidad Autónoma de Madrid. He joined the Universidad Rey Juan Carlos, Spain, where he is an assistant professor. He is currently finishing his PhD on high performance video processing under the supervision of Professor Angel Sánchez and Felipe Fernandez. He has authored or co-authored more than 20 papers in international journals and conferences and has participated in several research projects. His main research interest focuses on optimization techniques in computer vision and image processing.

R. Nagarajan obtained his BE (Hons), from Madras University, MTech from IIT Kanpur, India, and PhD from Madras University, India. He has worked in teaching and research positions in a number of universities in India, Iraq, Malaysia, the UK, and Japan. He has been working the field of robotics for several years. Recently he took up working in medical robotics. He has developed with his research students new sets of stereo and color image processing methods toward the application of navigational assistance for the blind. He is now working in developing a robotic automation system for lifting patients from hospital beds. His research group has acquired several awards which include the recent International Invention excellence award from Geneva. He has served in various committees of International conferences around the world. He has initiated the series of international conferences on AI and applications in Universiti Malaysia Sabah. Nagarajan is a fellow of IE, India, and a senior member of IEEE, USA.

Shigeko Nomura is presently a doctorate student in the Laboratory for Theory of Symbiotic Systems, Kyoto University, Japan. He received his BSc, BE, and MSc degrees from ITA at Aerospace Technical Center, EPUSP at University of São Paulo and FEELT at Federal University of Uberlândia in Brazil, respectively. He won a doctoral scholarship grant from the Japanese Government (Monbukagakusho). His current research interests include morphological image analysis, pattern recognition, auditory pattern perception and artificial neural networks.

Ee Ping Ong received BEng and PhD degrees in electronics and electrical engineering from the University of Birmingham, UK. He was with the Institute of Microelectronics, Singapore. Thereafter, he joined the Centre of Signal Processing, Nanyang Technologi-

cal University, Singapore. He has been with the Institute for Infocomm Research, Singapore, where he is currently a scientist. He is an IEEE senior member and also currently the chairman of IEEE Consumer Electronics Chapter, Singapore. His research interests include motion estimation, video object segmentation and tracking, perceptual image/video quality metrics and coding.

Franco Pedreschi accomplished his studies in seafood processing engineering at Universidad Nacional Agraria, Peru. He developed his master's and doctorate degrees in the Department of Chemical Engineering and Bioprocesses of Pontificia Universidad Católica de Chile. His doctoral dissertation was about structural changes in potato tissue during frying. He has pursued post doctoral studies in the Department of Food Engineering of Lund University, Sweden studying changes in physical properties of potatoes during frying. Currently he is an assistant professor at the University of Santiago in the Department of Food Science and Technology.

Fernando Pereira is a professor in the Electrical and Computer Engineering Department, Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal. His research interests include video analysis, processing, coding and description and multimedia interactive services. He has worked on ISO/MPEG for many years, notably as head of the Portuguese delegation, chairman of the MPEG Requirements group, and chair of many MPEG-4 and MPEG-7 ad hoc groups. He won the 1990 Portuguese IBM Award and an ISO Award for Outstanding Technical Contribution for his work in MPEG-4 development. Pereira received an MSc and a PhD in electrical and computer engineering from IST.

Antonio Plaza received a PhD in computer science from the University of Extremadura, Spain where he is currently an associate professor with the Computer Science Department. He has also been visiting researcher at the University of Maryland, NASA Goddard Space Flight Center and Jet Propulsion Laboratory. His main research interests include the development and efficient implementation of hyper spectral image analysis algorithms on parallel computing systems and hardware-based computer architectures. He has authored or co-authored more than 100 publications, including journal papers and conference proceedings, and currently serves as regular manuscript reviewer for more than 15 highly-cited journals.

Javier Plaza received a MSc in computer science from the University of Extremadura, Spain, where he is currently an assistant professor. His current research work is focused on the development of efficient implementations of nonlinear mixture model-based algorithms for abundance estimation of materials in hyper spectral scenes. He is also involved in the design and configuration of commodity cluster computing architectures for high-performance hyper spectral analysis. Other major research interests include development of quantitative and comparative applications for remote sensing, and the study of configuration and training of neural network architectures for hyper spectral image analysis.

Christophe Rosenberger is an assistant professor at ENSI of Bourges, France. He obtained his PhD from the University of Rennes I. He belongs to the Laboratory of Vision

and Robotics of Bourges in the Signal, Image and Vision research unit. His research interests concern image processing evaluation, quality control by artificial vision and pattern recognitions applications. He also works on the segmentation and interpretation images for robotic applications.

Besma Rouai-Abidi is a research assistant professor with the Department of Electrical and Computer Engineering at the University of Tennessee, USA. She occupied a post-doctorate position with the Oak Ridge Institute of Science and Energy and was a research scientist at the Oak Ridge National Laboratory. She was an assistant professor at the National Engineering School of Tunis, Tunisia. She obtained two MS degrees in image processing and remote sensing with honors from the National Engineering School of Tunis. She received her PhD from The University of Tennessee. Her general areas of research are in image enhancement and restoration, sensor positioning and geometry, video tracking, sensor fusion and biometrics. She is a member of IEEE, SPIE, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi and The Order of the Engineer.

Angel Sanchez received his MSc and PhD degrees in computer science from the Technical University of Madrid, Spain. Currently he is an associate professor in the Department of Computer Science, Statistics and Telematics at Rey Juan Carlos University, Madrid. His research interests are in the areas of computer vision, image analysis, biometrics and artificial intelligence.

Hong Shen received his PhD in electrical engineering from the Electrical, Computer and Computer Systems Engineering Department at Rensselaer Polytechnic Institute, USA. He received his BEng from the Electronic Engineering Department at Tsinghua University, China. He worked as a device engineer in the Institute of Semiconductors, Academic Sinica, China. He has worked with Siemens Corporate Research, USA. Currently he is a researcher and project manager working mainly in the area of computer-aided diagnosis for medical imaging applications. His research interests include image analysis, computer vision and imaging instrumentation.

Takayuki Shiose received his BE, ME and PhD degrees from the Faculty of Precision Engineering, Kyoto University (Japan). He was engaged as a research fellow at the Japan Society for the Promotion of Science and Research Associate at the Graduate School of Science and Technology, Kobe University. Now he is a research associate at the Graduate School of Informatics, Kyoto University and a visiting researcher at ATR Network Informatics Labs. His current research interests are on the assistant systems for a proficient-skill transfer from the viewpoint of ecological psychology.

Yap-Peng Tan received the BS degree in electrical engineering from National Taiwan University, Taiwan, ROC, and MA and PhD degrees in electrical engineering from Princeton University, USA. He was the recipient of IBM Graduate Fellowship from the IBM T.J. Watson Research Center, Yorktown Heights, New York. He was with Intel and at Sharp Labs of America. He has been a faculty member at the Nanyang Technological University, Singapore. His current research interests include image and video processing, content-based multimedia analysis, computer vision and pattern recognition. He is

a member of the Technical Committee on Visual Signal Processing and Communication of the IEEE Circuits and Systems Society, and the principal inventor on 10 US patents in the areas of image and video processing.

Bee June Tye received an MSc in telecommunications and information systems from the University of Essex, UK. She worked in AT&T Consumer Products (S) Pte Ltd and Institute of Microelectronics, Singapore. She joined Hewlett-Packard Singapore under the Business Printing Division, where she was a subject matter expert. She lead and provided expertise to functional project teams and participated in cross-functional initiatives. Currently, she is with Dell Global BV, Singapore branch. Her key interests include image and color processing, color matching for printers, printer drivers and print applications.

David Valencia is a research associate at the University of Extremadura, Spain, where he is currently pursuing a MSc in computer science. His research interests are on the development of parallel implementations of algorithms for hyper spectral imaging, with particular emphasis on commodity cluster-based systems (both homogeneous and heterogeneous) and hardware-based architectures. He is also involved in the calibration of hyper spectral analysis instruments and neural network-based applications.

Shilin Wang received his BEng degree in electrical and electronic engineering from Shanghai Jiaotong University, China and his PhD from the Department of Computer Engineering and Information Technology, City University of Hong Kong. He has been with the School of Information Security Engineering, Shanghai Jiaotong University, where he is currently an assistant professor. His research interests include image processing and pattern recognition.

Farrah Wong obtained her BE (Hons), MSc and PhD degrees in electronics engineering from Universiti Malaysia Sabah. She is presently working as a lecturer at the School of Engineering and Information Technology, Universiti Malaysia Sabah. She is a member of IEEE. Her research interests include robotics, image processing, neural network, fuzzy logic and machine vision.

Xiangyang Xue received BS, MS and PhD degrees in communication engineering from Xidian University, P. R. China. He works with the Department of Computer Science and Engineering, Fudan University, P. R. China, where he is currently a professor. His research interests include multimedia information process and retrieval, image and video coding and video streaming.

Sazali Yaacob received his bachelor of engineering degree in electrical engineering from Universiti Malaya. Upon graduation, he worked at a oil palm estate before joining Universiti Malaya's academic staff. He pursued his master of science degree in system engineering at the University of Surrey and later his doctor of philosophy in control engineering from the University of Sheffield, UK. He was promoted to associate professor by Universiti Malaysia Sabah and consequently appointed as the first dean of the School of Engineering and Information Technology. He is now serving at Northern

University College of Engineering Malaysia as a professor and dean for the School of Mechatronics. He has published work in national and international journals and conference proceedings. He has supervised a number of postgraduate students at both the master's and doctorate levels. His research interests are in artificial intelligence applications in the fields of acoustics, modeling and control. He was also granted the title of Chartered Engineer by the Engineering Council, UK and is a member of the Institution of Electrical Engineer, UK.

Keiji Yamanaka received his PhD in electrical and computer engineering from Nagoya Institute of Technology, Japan. He is currently working on the faculty of electrical engineering, Federal University of Uberlândia, Brazil, teaching undergraduate and graduate courses. He has several master's and doctoral students under his supervision and his research interests include neuron-computing, artificial intelligence and pattern recognition.

Liang Zhao received a BS, Wuhan University, China, and MSc and PhD degrees from Aeronautic Institute of Technology, Brazil, all in computer science. He worked as a software engineer and was a postdoctoral fellow at the National Institute for Space Research, Brazil. Dr. Zhao joined the University of São Paulo, Brazil, where he is currently an associate professor in the Department of Computer Science and Statistics. He was a visiting researcher in the Department of Mathematics, Arizona State University, USA. His research interests include artificial neural networks, nonlinear dynamical systems, complex networks, bioinformatics, pattern recognition and image processing.

Index

Symbols

- Γ -convergence theory 46
- Γ -limit 54
- 2-D image 2, 4
- 3-D image 4
- 3D medical image segmentation 250, 265

A

- abrupt shot boundary (ASB) 191
- ACO 74
- active appearance model (AAM) 259
- active contour model (ACM) 10, 166
- active contours 36, 381
- active shape model (ASM) 10, 161, 259
- adaptive lightning method 331
- adaptive threshold 142, 192, 201
- adaptive-focus statistical model 166
- airborne visible infra-red imaging spectrometer (AVIRIS) 270, 272, 280, 285
- algorithm evaluation 423
- animation 1
- annealing 74
- ant colony optimization 74
- application-oriented segmentation 12
- artificial intelligence (AI) 11, 361

- artificial neural network 320, 334, 337
- association rules 241
- asymmetric tracking 400
- audio scene (a-scene) 194
- audio-visual speech recognition 293
- automatic speech recognition (ASR) 293
- AutoS 318, 322, 332, 337

B

- background (BKG) 281
- Bayesian and information theory 17
- Bayesian framework 381
- Bayesian graphical model 28
- Bayesian inference 17-18, 44, 114
- Bayesian learning 114
- Bayesian theory 8
- belief propagation 18, 26
- benchmark image 366
- Bethe free energy 18, 35
- Bhattacharyya distance 42
- bifurcation parameter 96
- bilateral filters 125-126
- blind navigation 355, 362
- blurring effect 157
- boundary detection 294
- Brownian motion 22

C

camera 161
 cameraman 88
 Canny' filters 381
 Canon remote capture software 350
 catchment basin 271
 cerebral spinal fluid (CSF) 281
 change detection mask (CDM) 142
 channel sensitivity 400
 chaos 94
 chaotic oscillatory correlation 94, 103
 chaotic synchronization 94-95, 97, 103
 character image segmentation 319
 chromaticity 194
 clustering 6, 8, 18, 74, 94, 100, 106, 241
 coarea formula 56
 color image segmentation 7, 209
 color space analysis 294
 colour chart 350
 colour image 340
 colour rendering index 349
 comparative study 365
 competitive strategy 79
 complex image 75
 composite metric 403, 406, 419, 426
 comprehensive survey 424
 computable scene (c-scene) 194
 computation time 73
 computational complexity 141
 computer aided diagnosis (CAD) 250, 285
 computer tomography (CT) 250
 computer vision 250, 341
 conjugate gradient (CG) 58, 308
 content retrieval 141
 content-based video analysis 188
 contextual object relevance evaluation 409
 continuous closed contours 73
 contour area 375
 contour structure 381
 cooperative strategy 79
 cost function 18
 crisp clustering 296
 CSF 281
 cubic volume of interest 253
 cut weight 75

D

deepness 275, 277, 283
 deformable active contours 165
 deformable model 257
 deformable template 17, 26, 28, 44
 deformable templates matching 17
 delineation operating characteristic (DOC)
 432
 desynchronization 94
 detection rate (DR) 194, 428, 432
 deterministic 11
 digital colour camera (DCC) 349
 Dirichlet functional 47
 discrepancy evaluation 396
 discrepancy method 425
 dissimilarity measure 298
 divergence 8
 drawing 1
 dynamic programming 8, 28, 36

E

edge-linking 152
 EDISON algorithm 367, 387
 elementary metrics 403
 empirical discrepancy method 424
 empirical evaluation 426
 empirical goodness method 424
 empty area 375
 energy 75, 381
 Epanechnikov kernel 120, 130
 equidistant reference (ER) 322
 Euclidean distance (ED) 118, 166, 219,
 274, 283, 379
 evaluation criteria 365
 evaluation method 5, 12, 375, 396, 424
 evolutionary graph-based image segmenta-
 tion 72
 expectation maximization (EM) 114, 225
 exponential entropy 357
 extracted features 342
 extraction 11, 18

F

false alarm (FA) 428
 false alarm rate (FAR) 428

- false negative 428
- false positive 428
- FCH sequence 199, 202
- feature extraction 18, 341
- figure of merit 383
- filtering 39, 47, 113, 116, 120, 199
- final object mask 143, 152, 383
- finite element method (FEM) 258
- first-in-first-out (FIFO) 194
- food imagery 340
- fractal 8, 230
- fragmentation (FRAG) 399
- frame color histogram (FCH) 196, 202
- frame differencing 144
- FUB (Fondazione Ugo Bordoni) 156
- functional computation 381
- future directions 423
- fuzzy algorithm 74
- fuzzy c-mean (FCM) 295, 305, 315, 367
- fuzzy c-means with shape function (FCMS) 293, 305
- fuzzy clustering 296, 300, 357
- fuzzy image processing (FIP) 357
- fuzzy logic 10, 342, 362
- fuzzy system 238

- G**
- Gabor filter 8, 241
- Gaussian distribution 163
- Gaussian filter 47, 215, 304
- Gaussian Markov random field 231
- Gaussian mixture model 8
- Gaussian noise 144, 149, 281, 300
- generalized likelihood ratio (GLR) 193
- generative approach 19
- generic algorithm 8
- genetic programming 74
- geodesic active contour 165
- geodesic distance map 253
- geometric reward 40
- geostatistics 232
- Gibbs distribution 23, 114, 216
- Gibbs random field 8
- global criteria 426
- global priors 251
- globally coupled map (GCM) 104
- goodness evaluation 395
- goodness method 425
- gradual shot boundary (GSB) 191
- graph cuts 75, 209, 220
- graphics 1
- GRASP 74
- gray matter (GM) 281
- ground glass nodules (GGN) 255
- ground truth (GT) 424, 433
- grouping 17, 18, 36

- H**
- Hammersley-Clifford theorem 216
- hard clustering 296
- Hausdorff length 55
- Heaviside function 100, 104
- hierarchical social (HS) 72
- highest confidence first (HCF) 209, 213
- Hough transform 19, 254, 255
- human factor 12
- human visual system (HVS) 396, 399, 432

- I**
- IKONOS 429
- image acquisition 341
- image analysis 2, 320
- image engineering 2
- image pre-processing 341
- image primitives 254, 261
- image processing 2, 92
- image segmentation 1, 3, 46, 341, 426
- image understanding 2
- independent component analysis 241
- information capturing 319
- information theory 18
- infra-red 429
- intensity image 341
- intensity reward 39
- inter-frame histogram difference (IFHD) 197
- inter-frame mode matching 129
- inter-group movement 78
- inter-object disparity 398
- inter-object metrics 403
- interaction energy 29

intra-frame local mode estimation 130
 intra-group movement 78
 intra-object homogeneity 397
 intra-object metrics 403
 intraclass variance 345
 ISODATA 271, 283
 iterated conditional mode (ICM) 213

J

junction classification 37
 junction detection 36
 junction model 36

K

kalman filter 144
 knowledge based segmentation 250
 Kodak 350

L

Lagrange multipliers 18
 Langevin diffusion 23
 Laplacian 48
 least-median-square (LMS) 149
 Lebesgue integral 53
 Lieven & Luthon's method (LL) 310
 lighting condition 194
 linear predictive filter 144
 lip segmentation 294, 306
 local criteria 426
 local optima 36
 logarithmic entropy 357
 Lyapunov exponents 99

M

magnetic resonance (MR) 250, 270
 magnetic resonance imaging (MRI) 250,
 280
 Mahalanobis distance 163, 166, 178
 Markov fields 241
 Markov model 8, 381
 Markov random field (MRF) 8, 18, 28,
 114, 209, 255, 295
 masking 400
 mathematical model 11

mathematical morphology 212, 229, 238,
 320
 max flow algorithm 75
 maximum-likelihood 193, 195
 mean opinion score (MOS) 396
 mean shape 165
 mean shift 213
 MeasTex 230
 median filter 144
 medical imaging 274
 mental imagery 356
 merged region 73
 message passing interface (MPI) 287
 metaheuristics 74
 Metropolis-Hastings algorithm 22
 microwave remote sensing 429
 minimum cut 75
 miss detect 428
 modified region adjacency graph 73
 morphological approach 246, 270, 318,
 320
 morphological image processing 321
 morphology 238
 most probably stationary pixels (MPSPs)
 143
 motion activity 400
 movement 78
 moving edge 142, 151
 moving object tracking 130
 moving picture experts group (MPEG) 141,
 189
 MPEG-4 standard 395
 MPEG-7 141, 189
 multi-level logistic (MLL) 213, 219
 multi-scale edge detection 8

N

navigation aid for visually impaired (NAVI)
 356
 NCut 77, 86, 90
 neighborhood 229
 nesting structure 233
 Neumann boundary 47, 56, 59, 69, 97
 neural network 10, 238
 neutral area 375

non-oversegmentation 73
non-parametric model 144

O

object segmentation 141
object-based video coding 141
objective criteria 426
objective evaluation 395
objective function 77
objective video segmentation 396
object's similarity metric 410
Odet's criteria (ODI) 373, 384
one-object, multiple-background clustering 315
optimal window 229
optimization 17, 22, 74, 85, 297
optimization algorithm 18
oscillatory correlation 94
output image 89, 424

P

pan-tilt-zoom (PTZ) 162
panchromatic image 429
parallel implementation strategy 270
Pareto front 376, 432
partial differential equation (PDE) 258
Parzen windows 24
pattern recognition 10
PDF 192, 255
pepper noise 295
performance metrics 424, 433
periphery 77
Perona-Malik equation 46, 49
perturbation detection rate (PDR) 432
Picard iteration 297, 299
pixel 4, 30, 51, 61, 73, 76, 81, 104, 127, 147, 169, 180, 215, 274, 305
point distribution model (PDM) 259
Power Shot G3 349
precise segmentation 260
preliminary object mask (POM) 142
primitive 254, 262
principal component analysis (PCA) 173, 259
probability density function (pdf) 192
probability theory 232

processing element (PE) 278
proposal probabilities 24
pruning 43

Q

quadratic index of fuzziness 357
quality measurement 374, 381

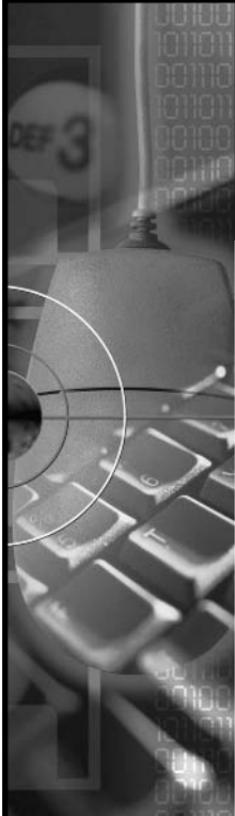
R

range of influence 233
receiver operating characteristic (ROC) 340, 375, 431
reference image 424
region adjacency graph (RAG) 73, 82, 277
region segmentation 17, 18
relative evaluation 411
relative objective evaluation 396
remote sensing 274
reversible jump 21
RGB color space 294
RGB image 295, 341, 344
root mean squared error (RMS) 370

S

satellite imaging 271
saturation effect 400
scene segmentation 202
scene transition graph 193
segment block 41
segmentation 1, 72, 130, 166
segmentation algorithm 8, 12, 72, 142
segmentation error (SE) 311
segmented image 341, 424
semantic scene 194
semantics 190
shadow lighting method 331
shape fidelity 398
shortest path 144, 152
shortest-half robust statistical method 149
shot boundary detection (SBD) 188, 190
shot color histogram (SCH) 202
signal-to-noise ratio (SNR) 281, 429
similarity of objects 410
simple image 75
simulated annealing 8, 74

- simulation 60
single board processing system (SBPS) 356
single metric 426
snake 165, 381
snake method 48, 163
space-time segmentation 166
Spatial accuracy 398
spatial feature 399, 404
spatial fuzzy c-mean clustering (SFCM) 293, 295
spatial perceptual information (SI) 400
spatial refinement 142
spatio-temporal chaos 94
spatiotemporal accuracy metrics 408
spectral angle (SAD) 274
stable content 406
stable content 413
standalone evaluation 396, 406, 410
standalone objective evaluation 395
stationary pixels 147
statistical mechanics method 18
statistically stationary area (SSA) 146, 157
stochastic diffusion 21
stochastic gradient descent 18
stochastic model 230
stochastic process 23
subjective criteria 426
subjective evaluation 395
subjective relative evaluation 397
subjective standalone evaluation 396
supervised evaluation 365
support vector machine (SVM) 433
SUSAN 8, 37
Swendsen-Wang method 75
synchronization 94
synchrony-de-synchrony dilemma 94
synthetic 75
systematic study 423
- T**
- tabu search 74
temporal accuracy 399, 408
temporal perceptual information (TI) 401
temporal stability 399
textural classification 228, 243
- textural segmentation 229
texture 229, 366
texture analysis 229
texture description 229
texture primitives 229
threshold setting 73
thresholding 10, 39, 201, 271, 294, 357
true negative (TN) 428
true positive (TP) 428
- U**
- ultrasound scanner 250
uniformly moving content 406, 414
user interface 319
- V**
- variational method 18
variogram 232
variography 229
vessel extraction 11
video analysis 188
video content 189
video content analysis (VCA) 433
video object plane (VOP) 158
video scene (v-scene) 194
video segmentation 394, 426
video semantics 117, 190
VisTex benchmark 230
voronoi tessellation 230
voxel 4, 257, 265
- W**
- W4 165
watershed 10, 73, 209
watershed algorithm 241
wavelet 10
wavelet modulus maxima 8
wavelet transform 241
weighted majority 433
- Z**
- Zhang and Mercereau's method (ZM) 310



Experience the latest full-text research in the fields of Information Science, Technology & Management

InfoSci-Online

InfoSci-Online is available to libraries to help keep students, faculty and researchers up-to-date with the latest research in the ever-growing field of information science, technology, and management.

The InfoSci-Online collection includes:

- Scholarly and scientific book chapters
- Peer-reviewed journal articles
- Comprehensive teaching cases
- Conference proceeding papers
- All entries have abstracts and citation information
- The full text of every entry is downloadable in .pdf format

Some topics covered:

- Business Management
- Computer Science
- Education Technologies
- Electronic Commerce
- Environmental IS
- Healthcare Information Systems
- Information Systems
- Library Science
- Multimedia Information Systems
- Public Information Systems
- Social Science and Technologies

“...The theoretical bent of many of the titles covered, and the ease of adding chapters to reading lists, makes it particularly good for institutions with strong information science curricula.”

— Issues in Science and Technology Librarianship

InfoSci-Online features:

- Easy-to-use
- 6,000+ full-text entries
- Aggregated
- Multi-user access



To receive your free 30-day trial access subscription contact:

Andrew Bundy

Email: abundy@idea-group.com • Phone: 717/533-8845 x29

Web Address: www.infosci-online.com

InfoSci-Online
Full Text • Cutting Edge • Easy Access

A PRODUCT OF  IDEA GROUP INC.

Publishers of Idea Group Publishing, Information Science Publishing, CyberTech Publishing, and IJM Press

infosci-online.com