To the Graduate Council:
I am submitting herewith a dissertation written by Yongsheng Pan entitled "Image Segmentation using PDE, Variational, Morphological and Probabilistic Methods". I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

J. Douglas Birdwell, Major Professor

Seddik M. Djouadi, Coadvisor

We have read this dissertation
and recommend its acceptance:

Hairong Qi, Committee Member

Tse-Wei Wang, Committee Member

Xiaobing Feng, Committee Member

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and
Dean of the
Graduate School

(Original signatures are on file with official student reconds.)

Image Segmentation using PDE, Variational, Morphological and Probabilistic Methods

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Yongsheng Pan
December 2007

Copyright © 2007 by Yongsheng Pan.
All rights reserved.

Dedication

This dissertation is dedicated to my wife, Chen Wang, and my family members, my father Baoqing Pan, my mother Suying Liu, my brother Fusheng Pan, my father-in-law Fenghe Wang, my mother in law Li-e Yan and my brother-in-law Zhenming Wang. Without their understanding, encouragement and support, this dissertation can not come true.

Acknowledgments

These five years 2003-2007 as a graduate student in ECE at UTK are absolutely the best time ever I spent in my life. The experience here benefited me a lot mentally, academically and financially. I would like to thank many people who have supported me in all these aspects.

First and foremost, I would like to thank my advisor, Dr. Douglas Birdwell for his innovative guidance, advice and persistent help. I have learned a lot from him in my research, including his solid background, his profound experience, his smart ideas, and his broad horizons. I have also been taught how to be a good engineer, a good person, a man of principles. It is usually hard for people to know such a distinguished man during their whole lives. I am so lucky to not only meet him, but to be his student for five years. I am sure what I have learned from him will continue to benefit me during the rest of my life.

Second, I would like to thank my coadvisor, Dr. Seddik Djouadi, who introduced me to the state-of-art area of curve evolution in 2004. His participation in our weekly meeting brings a lot of information to me on how to do good research and how to be a good researcher. I always appreciate his confidence in me.

I would further like to thank other committee members: Dr. Xiaobing Feng, Dr. Tse-Wei Wang and Dr. Hairong Qi. I really appreciate Dr. Feng for his outstanding advice on my research. The passion and efforts from Dr. Wang help to build a very nice environment for the research lab, which I have enjoyed so much. I have also learned a lot from the courses of Dr. Wang and Dr. Qi. The time, effort, valuable suggestions and constructive comments from all the committee members deserve my strong appreciation.

I would like to thank Shaurya Kumar and Bruce Johnson, whose efforts made our cooperations successful and beneficial. Within the Laboratory for Information Technologies, I owe many thanks to both students and staff. Specially, I thank Joe Carter for his nice treatment in his home. To my fellow graduate students, I express my sincerest gratitude for the many conversations that have had a tremendous impact on my research and myself as a person: Zushou Hu, Zhong Tang, Zeqian Shen, Miao Chen. Thank you so much.

Finally, I must express my appreciation to the many friends outside of my studies who have helped to relieve the sometimes stressful solitude of graduate school. Zhong Du, Yue Zheng, Yang Liu, Shaomeng Li, Wei Jiang, Lidan Miao, Mengwei Li, Wenjuan Zhang, etc. The list is too long to be writen all the names here. Your friendship through both good times and hard times have been a source of strength.

Abstract

The research in this dissertation has focused upon image segmentation and its related areas, using the techniques of partial differential equations, variational methods, mathematical morphological methods and probabilistic methods. An integrated segmentation method using both curve evolution and anisotropic diffusion is presented that utilizes both gradient and region information in images. A bottom-up image segmentation method is proposed to minimize the Mumford-Shah functional. Preferential image segmentation methods are presented that are based on the tree of shapes in mathematical morphologies and the Kullback-Leibler distance in information theory. A thorough evaluation of the morphological preferential image segmentation method is provided, and a web interface is described. A probabilistic model is presented that is based on particle filters for image segmentation.

These methods may be incorporated as components of an integrated image processed system. The system utilizes Internet Protocol (IP) cameras for data acquisition. It utilizes image databases to provide prior information and store image processing results. Image preprocessing, image segmentation and object recognition are integrated in one stage in the system, using various methods developed in several areas. Interactions between data acquisition, integrated image processing and image databases are handled smoothly. A framework of the integrated system is implemented using Perl, C++, MySQL and CGI.

The integrated system works for various applications such as video tracking, medical image processing and facial image processing. Experimental results on this applications are provided in the dissertation. Efficient computations such as multi-scale computing and parallel computing using graphic processors are also presented.

Contents

1 Introduction and Motivation **1**
1.1 Introduction to Image Processing and Computer Vision 2
1.2 Motivations 3
1.2.1 Integrated Image Processing Systems 3
1.2.2 Image Processing Methods 5
1.2.3 Image Processing Applications 6
1.3 Overview of the Dissertation 6

2 Literature Review **8**
2.1 Integrated Image Processing 8
2.1.1 Integration between Imaging and Image Processing - Computational Photography 8
2.1.2 Integration between Image Denoising and Image Segmentation - the Mumford-Shah Functional 9
2.1.3 Integration between Image Segmentation and Recognition 11
2.2 General Theoretical Methods for Image Processing 12
2.2.1 Partial Differential Equations 13
2.2.2 Wavelet Methods 20
2.2.3 Variational Methods 21
2.2.4 Stochastic Methods 21
2.3 Applications to Image Processing 22
2.3.1 Curve Evolution in Image Segmentation 22
2.3.2 Shape Representation and Recognition 25
2.3.3 Video Tracking 26
2.3.4 Numerical Implementations of Image Processing Algorithms 27
2.3.5 Parallel Image Processing using Graphic Processors 27
2.4 Summary 28

3 Image Segmentation using Curve Evolution **29**
3.1 Curve Evolution using both Gradient and Region Information 29
3.1.1 Model Description 29
3.1.2 Hierarchical Approach 30
3.1.3 Extension to Vector-Valued Images 31
3.1.4 Experimental Results 31
3.1.5 Conclusions and Future Research 33
3.2 Image Segmentation using Curve Evolution and Anisotropic Diffusion: An Integrated Approach 33

3.2.1 Curve Evolution Model . . . 35
3.2.2 Connections between Curve Evolution and Anisotropic Diffusion . . 36
3.2.3 Curve Evolution Model integrating Anisotropic Diffusion . . . 37
3.2.4 Implementation Issues . . . 37
3.2.5 Experimental Results . . . 38
3.2.6 Summary . . . 43
3.3 Bottom-Up Hierarchical Image Segmentation using Region Growing and the Mumford-Shah Model . . . 43
3.3.1 The Initialization Problem of The Chan-Vese Model . . . 43
3.3.2 Analysis of the Initialization Problem of the Chan-Vese Models . . . 45
3.3.3 Fast Curve Evolution Method Without Solving PDEs . . . 47
3.3.4 Image Segmentation Using Region Growing and the Mumford-Shah Functional . . . 49
3.3.5 Implementation Issues . . . 51
3.3.6 Experimental Results . . . 52
3.3.7 Conclusions and Future Work . . . 54
3.4 An Improvement of the Image Segmentation Method using Region Competition and the Mumford Shah Functional . . . 57
3.4.1 Experimental Results . . . 58
3.4.2 Summary . . . 58
3.5 Efficient Implementations of Image Segmentation . . . 58
3.5.1 Narrow Band Method . . . 61
3.5.2 Multi-scale Method . . . 61
3.5.3 Parallel Computing Method . . . 61
3.5.4 Experimental Results . . . 61
3.6 Summary . . . 64

4 Preferential Image Segmentation using Mathematical Morphologies 68
4.1 Introduction . . . 68
4.2 Background . . . 70
4.2.1 Image Representation using Connected Components of Sets of Finite Perimeters . . . 70
4.2.2 Image Representation using the Tree of Shapes . . . 75
4.2.3 Color and Geometry in Mathematical Morphologies . . . 76
4.2.4 Planar Curve Matching . . . 77
4.2.5 Scale Space Theory for the Tree of Shapes . . . 78
4.3 Preferential Image Segmentation . . . 78
4.4 Experimental Results . . . 81
4.5 Performance Evaluation of Preferential Image Segmentation using the Tree of Shapes . . . 88
4.5.1 Introduction to the Image Dataset . . . 88
4.5.2 Prior Selection . . . 92
4.5.3 Experimental Results . . . 93
4.6 Preferential Image Segmentation User Interface using an Image Database and the Web . . . 99
4.7 Summary and Future Research . . . 99

5 Image Segmentation by Matching Distributions **104**
5.1 Probabilistic Model for Curve Evolution using Particle Filtering 104
5.1.1 Introduction 104
5.1.2 The Geometric Active Contour Model 105
5.1.3 Probabilistic Curve Evolution Model 106
5.1.4 Particle Filters 108
5.1.5 Implementation 110
5.1.6 Experimental Results 112
5.1.7 Conclusion 112
5.2 Preferential Image Segmentation by Matching Histogram-based Distributions 112
5.3 Summary 115

6 Conclusions **117**
6.1 Image Processing using Graphic Processors 118
6.2 Applications using Preferential Image Segmentation 120
6.2.1 Video Capture from IP Cameras 120
6.2.2 Preferential Image Segmentation using IP Cameras for Object Tracking 121
6.2.3 Summary for Video Tracking 121
6.3 Summary 121

Bibliography **123**

Vita **135**

List of Figures

2.1 Curvature deformation of a rectangle under Eq. 2.8 14
2.2 Curvature deformation of an irregular concave curve under Eq. 2.8 15
2.3 Singularities may develop under the deformation of Eq. 2.13 16
2.4 Curvature deformation of an irregular concave curve under $\partial C/\partial t = (1+\kappa)N$. 17
2.5 Curvature deformation of an irregular concave curve under complicated speed function 18

3.1 Illustration of region information in the model 31
3.2 Illustration of region information in the model 32
3.3 Segmentation of multiple objects in an image using the proposed model . . 33
3.4 Segmentation of an image with triple junctions 34
3.5 Illustration of region information in the model 38
3.6 Segmentation of multiple objects in a noisy image 39
3.7 The boundary leakage problem 40
3.8 Segmentation of real images 40
3.9 Segmentation of real images 41
3.10 Segmentation of triple junctions in an image using an hierarchical approach 41
3.11 Solution of the stair-casing problem 42
3.12 Segmentation of an image with heavy noise 42
3.13 Example of the initialization problem, where $m_2/n_2 = 0.21$ is larger than $m_1/n_1 = 0.16$ 45
3.14 Example of the initialization problem, where $m_2/n_2 = 0.036$ is smaller than $m_1/n_1 = 0.065$ 46
3.15 Different initialization in the Chan-Vese model may generate different segmentation results 46
3.16 Coupling between curve evolution may enlarge initialization problems . . . 48
3.17 Comparison of the proposed fast curve evolution method to the classical method solving PDEs [105] [130] 49
3.18 Fast curve evolution of the Chan-Vese model for complicated cases 50
3.19 Comparison of the proposed method and the Chan-Vese model for images with multiple regions 53
3.20 More results of the proposed method for complicated images 53
3.21 Stability of the proposed method w.r.t. noise 53
3.22 Extension of the proposed method to color images 54
3.23 Effects of ν on segmentation results. 55
3.24 The ability of the proposed method for weak edges 55
3.25 Experimental results for real images 56
3.26 Comparison of the improved model with the previous model 59

3.27 Experimental results for real images 60
3.28 Illustration of narrow band method 62
3.29 Illustration of parallel computing for curve evolution 63
3.30 Multi-scale curve evolution 65
3.31 Multi-scale curve evolution on a large image 66
3.32 Parallel computing of curve evolution 66

4.1 Illustration of preferential image segmentation 69
4.2 Illustration of the tree of shapes 76
4.3 Illustration of tree of shapes for a color Image 77
4.4 Illustration of preferential image segmentation using the tree of shapes . . . 80
4.5 Preferential image segmentation of a car using the tree of shapes 83
4.6 More examples for performance image segmentation 84
4.7 Illustration of rotation invariance 85
4.8 Illustration of scale invariance 85
4.9 Illustration of scale invariance 86
4.10 Illustration of noise effects 86
4.11 Preferential image segmentation for image databases 87
4.12 Preferential image segmentation of faces using the tree of shapes 89
4.13 Examples of prior images in the training set 90
4.14 Examples of images to be segmented in the test set 91
4.15 Illustration of shape prior selection in training images 92
4.16 Examples of prior objects selected from the training images 94
4.17 Loose criteria for false alarm evaluation 95
4.18 Examples of segmentation results 96
4.19 Examples of segmentation results 97
4.20 Performance evaluation with respect to the distance threshold 98
4.21 Select and visualize images from a image database 100
4.22 Retrieve and visualize segmentation results 101
4.23 Upload, visualize and store the image to be segmented 102

5.1 Boundary detection using geometric curve evolution 106
5.2 Topological change of the probabilistic curve evolution 112
5.3 The role of region information in the probabilistic curve evolution 113
5.4 Segmentation of multiple regions with noise 113
5.5 Segmentation of real noisy images 113
5.6 Segmentation of a leave image based on prior color information 115

6.1 Image addition by GPGPU 119
6.2 Video capture from IP cameras using Java media framework 120
6.3 Preferential image segmentation for video tracking 122

Notation

BV	bounded variation
WBV	weakly bounded variation
$\mathcal{R}^N$	the set of real numbers of dimension N
$\mathcal{H}^{N-1}$	the Hausdorff measure of dimension $(N-1)$
κ	curvature
$\mathcal{N}$	inward unit normal to a curve
s	Euclidean arc length
ψ	level set of an evolving curve
ϕ_l	a decreasing function of image gradients
H	Heaviside function
div	divergence operator
max	maximum
min	minimum
α, β, ϵ, ν, γ	constant variables
∇	the gradient operator
∇^2	the Laplace operator
$\surd$	square root
$\cap$	set intersection
$\cup$	set union
$\subset$	subset
$\subset\subset$	compactly contained
$\subseteq$	proper subset
$\supset$	proper superset
$\supseteq$	superset
$\forall$	for all
$\exists$	there exists
$\cdot$	dot product
∞	infinity
$\sum$	sum
$\prod$	product
$\iint$	integration
$\in$	element of a set
∂	partial derivative

Chapter 1

Introduction and Motivation

This research has been motivated by a belief that the combination of image processing steps that have traditionally been treated separately, such as segmentation and recognition, can improve overall performance. Insights from optimization support this belief: The combination provides additional design flexibility without sacrificing the option of independent design solutions. Under these conditions the optimal combined design (under a suitable criterion) will do no worse than two independently designed components. Intuition has also played a role: For example, one should be able to segment desired objects better if one knows what he wants to find. The segmentation and recognition problems have been treated separately in most prior work because they are both difficult problems that can nevertheless be independently defined, and because the algorithms that yield solutions are computationally intensive. The processing capabilities of individual computer processors continues to expand, corresponding roughly to Moore's Law, and the capabilities of parallel computation have undergone significant evolution. Computational constraints are not as confining as they once were. It is time to examine the benefits of integrated image processing approaches.

Many research avenues are exposed when one considers the integration of image processing steps that have, for the most part, been treated separately in the published literature. Examples having multiple steps include the mechanical control of a camera to selectively acquire higher resolution within its field of view. This process can be driven by the content of acquired images, including their segmentation and recognition of content. A second example is the recognition and localization of three dimensional objects from multiple views of a scene, acquired using multiple camera platforms. A third example is consideration of the trade-off between time required to detect, localize, and identify an object, communications across bandwidth-constrained channels among a distributed group of collaborating sensor platforms, and energy utilization and management for platforms with limited resources. A fourth example is the coordination of platform motion control and image acquisition systems to detect and identify objects of interest.

One must begin with small, tractable problems and work toward the design of more complex systems such as those listed in the last paragraph. In this research, the goal has been the integration of object recognition, using prior information such as a database of images of known objects, and image segmentation. As progress toward this goal was made, other problems were formulated and solved, such as the use of anisotropic diffusion to selectively reduce the adverse effects of noise upon segmentation while maintaining detailed

segment boundary information (Section 3.2). Another example is the bottom-up segmentation method that has been developed, which approximates the solution of a variational problem but avoids its computational complexity (Section 3.3). A third example is the application of particle filtering to the segmentation problem, introducing a probabilistic approach used in nonlinear filtering (Section 5.1). The research culminates in the development of a preferential segmentation and object identification method that uses a tree of shapes representation of level sets and objects within an image to detect, localize, and identify objects in acquired images (Chapter 4).

The results that demonstrate the application of these combined approaches are varied. One test data set that was used consists of a collection of electronic parts and allows a controlled study of the adverse effects of lighting, shadows, occlusion, and background on the preferential segmentation method. Other test data sets included faces, leaves, and frames from a surveillance video. The method utilizes a database of prior objects and detects the presence and locations of those objects in other images.

Surveillance is an obvious application, such as tracking ships in a harbor, moving vehicles in a parking garage, or people in a mall or an airport. In all these cases it is frequently important to flag and track only specific ships, vehicles, or individuals. In medical images one wishes to locate specific features, such as organs or joints, or tissue types. These features may be tracked in video imagery, as has been done for studies of the beating heart. Automated testing systems look for poorly assembled mechanisms, such as portions of cars or electronic assemblies, poor connections (in, for example, solder inspection applications), or the incorrect use of components or use of damaged components. All these cases can be characterized by a database of images of prior objects and methods to preferentially segment newly acquired image data, leading to an understanding of the selected portions of scenes.

A high-level introduction to image processing and computer vision is provided by this chapter as a foundation for the research. The introduction is intended to motivate, from a reasonably generic perspective, the research directions that have been followed, and to provide a foundation for the information presented in future chapters. The discussion follows the brief introduction that has been provided up to this point: the integration of image processing stages, and the theoretical developments that support this introduction and are utilized in the present work are discussed, followed by a summary of applications of image processing.

1.1 Introduction to Image Processing and Computer Vision

Image processing and computer vision are two tightly coupled research areas. Their developments are motivated by recent developments in hardware (computers, digital cameras, graphics processing cards and FPGA), mathematics (partial differential equations, wavelets, stochastic methods, variational methods and information theory), information technologies and computational methods (data structures and algorithms, finite difference methods, finite volume methods and finite element methods). Both areas deal with digital images, and their focus is the development of technologies to allow computers to understand visual information. Image processing and computer vision systems that use **automatic** methods (methods that can operate successfully without human involvement) for complicated scenes are the central theme of this research. Such systems are candidates for inclusion in autonomous systems, such as unmanned surveillance, detection, and identification systems.

The topics in image processing can be roughly categorized as follows: imaging, image denoising, image restoration, image coding, image segmentation, image transformation, object representation and recognition, and content-based image retrieval. Typical tasks for computer vision are scene interpretation, object recognition, optical character recognition, egomotion, camera calibration, registration, feature extraction and video tracking.

The techniques of image processing and computer vision overlap the areas of vision theory, artificial intelligence, computer graphics, mathematical methods and statistics.

The focus of the dissertation is the development of mathematical methods and their applications in image segmentation and its related areas - shape representation and recognition, and video tracking. A current trend in image processing and computer vision is the introduction of well-established mathematical theories into these areas, including partial differential equations(PDEs) [10] [48] [127], calculus of variations [10], information theory [37], wavelet theory [41] [93], stochastic methods [90] [157], fractal geometry [51], functional analysis [123], differential geometry [19], optimization [15], control theory [128] and numerical analysis [65] [145]. This dissertation focuses on preferential image segmentation, shape representation and video tracking using mathematical methods (integrated PDE, variational, wavelet and stochastic methods), beginning with methods based upon curve evolution, anisotropic diffusion, and the Mumford-Shah model.

1.2 Motivations

The motivations of this dissertation can be grouped into three categories: development of a theory of integrated image processing systems, development of integrated image processing methods, and image processing applications. These three categories are discussed in the following subsections.

1.2.1 Integrated Image Processing Systems

Classical image processing is usually decomposed into several stages (in order of processing): imaging/image acquisition, image preprocessing (denoising/enhancement/restoration), image segmentation/feature extraction, and object recognition, which correspond to the topics shown in the previous section and are shown in the following figure.

$$\boxed{\text{Data Acquisition}} \longrightarrow \boxed{\text{Preprocessing}} \longrightarrow \boxed{\text{Segmentation}} \longrightarrow \boxed{\text{Object Recognition}}$$

Scheme 1.2.1 Components of Classical Image Processing Systems

An image is processed in the above system from one step to another. Every stage takes as input the output of the former block except the first block, which captures image information. The stages in Scheme 1.2.1 are formulated independently. This is a good point in some cases, because different methods can be developed for each stage without having to consider the other stages. It is up to the user to select different algorithms in every step, and the best choice depends upon the application. The system has considerable flexibility. The strategy of the design is to "Divide and Conquer".

The above system, however, may have serious problems in complicated cases. First, the performance of every step (except the first imaging step), depends on the output of the preceding step. If the preceding step does not work well, the later step is very likely to generate bad results. Second, prior information is not fully utilized in the system, especially

in the imaging step and the segmentation step. It is challenging to get good processing results without utilizing prior information about the image's or scene's content in complicated cases. Finally, the system is not fully automatic. If the result is not satisfactory, the user has to re-design the algorithms and try again. The feedback process in the scheme is manual.

Recent developments attempt to integrate these separate stages. For example, the Mumford-Shah model [100] integrates image denoising and image segmentation by minimizing the Mumford-Shah functional. The image parsing framework described in [148] unifies segmentation, detection and recognition using Markov Monte Carlo methods. Other methods have been proposed to integrate prior information in image processing; examples are described in [29] [53] [88]. These methods provide good starting points for the utilization of prior information in low-level image segmentation. However, prior information is not fully utilized in these methods. For example, the works in [29] and [88] only utilize prior shape information for image segmentation, and the method described in [53] utilizes prior gray density information from histograms (which loses spatial prior information) for video tracking.

These limitations provide motivation for a portion of this research. Methods and algorithms that support an integrated image processing system are proposed, which integrate image acquisition, image preprocessing, image segmentation and image recognition steps. The system utilizes prior information (color, texture, shape and motion) using a image database of prior images. The prior data focus attention on similar details in new images, and the integrated system generates image segments that are identified with objects in the prior database. The integrated system takes the following structure:

$$\boxed{\text{Image Acquisition}} \rightleftarrows \boxed{\text{Preprocessing/Segmentation/Recognition}} \rightleftarrows \boxed{\text{Image Database}}$$

Scheme 1.2.2 Components of Proposed Image Processing Systems

As shown in Scheme 1.2.2, image preprocessing, image segmentation and recognition work in an integral fashion in the proposed system. The image acquisition process captures new data (image or video), and the image database provides prior information which may be in the form of a library of known images. Both processes interact with the integral preprocessing/segmentation/recognition process. Both provide information for processing, and may also update their function based upon feedback from other processing results. For example, a video camera may be adjusted using the results to perform video tracking to make sure an identified moving object remains in the center of the captured scene. The image database may store new objects that are recognized with high confidence in the database of priors to provide more information for later application.

The proposed framework has two novel aspects: First, image preprocessing, image segmentation and image recognition methods are integrated. Second, this central processing step is integrated with image acquisition and image databases using bidirectional information exchange. The specific methods that will be employed are derived from mature areas such as the theory of PDEs, the calculus of variations, adaptive control, information theory and wavelet methods. An overview of how these methods are integrated is provided in the next subsection.

1.2.2 Image Processing Methods

Variational methods are good candidates for the integration of image preprocessing, image segmentation and image recognition, as shown by the work described in [100] and [148]. An energy functional is selected whose minimization leads to a locally optimal solution to a task such as object boundary detection. A solution strategy can be designed using a PDE whose steady-state solution corresponds to a locally optimal solution to the problem [10] [48] [127]. Some work [148] has done in the combination of preprocessing/segmentation/recognition tasks, demonstrating the feasibility of at least a portion of the proposed research.

With the invention of snakes [74] and geometric curve evolution methods [22] [92], PDE methods introduced dynamics into the segmentation of static images, and have become one of the most popular methods for segmentation. Since the segmentation of a static image is a geometric problem, the combination of PDEs and differential geometry generates a powerful tool in image analysis, with applications in image segmentation [100] and shape representation [132].

Wavelet methods [41] [93], which are related to the generalized Fourier transform, provide both spatial and frequency information about images. These methods are widely used in image segmentation [12], image compression [7] and many other areas.

Stochastic methods are based on Bayes' rule, and provide a good way to utilize prior information. Stochastic methods such as Markov random fields (MRF) [90] [157] have become very powerful tools for image segmentation. For example, MRF may integrate color and textured information into a perceptual model for good image segmentation results [28].

The book by T. Chan [26] demonstrates that the above areas are closely correlated: stochastic methods can be shown to be a variational problem; PDE methods are usually implementations of variational problems, and are closely related to wavelet methods. It seems challenging to develop a large framework to integrate all the four areas, but the relationships between them will lead us to select and combine appropriate methods from different areas for different applications.

The areas listed above may be insufficient in applications. In the proposed framework, the interaction between image acquisition and the preprocessing/segmentation/recognition process, however, requires methods to evaluate the performance, and control techniques to handle feedback among the processes. Information theory [37] or measure theory [50] may be helpful in performance evaluation. Adaptive control theory [128] may be used to dynamically adjust the parameters and even structures of a system. The interaction inside the preprocessing/segmentation/recognition process, on the other hand, will require the application and integration of pattern recognition [44] and learning [140] methods. Many techniques in the areas of information theory, control, pattern classification and learning may be categorized within the above four areas, and it may be worthwhile to utilize them in the framework.

Further extension of the integrated theories is possible. For example, the utilization of prior information in preferential image segmentation may benefit from knowledge representation techniques from the areas of artificial intelligence and expert systems. Both areas lie outside the scope of this research. It can be expected, however, that this dissertation may be helpful for further research in their directions.

1.2.3 Image Processing Applications

Many applications can be studied using the integrated systems and the integrated theories proposed above. These applications include not only classical image processing applications, but also computer vision problems.

A primary motivating application in this dissertation has been video surveillance. The framework works as follows. First, objects of interests are preferentially segmented and recognized from the images or video frames, using the prior information provided by an image database. The motions of these objects are tracked, and the camera may be dynamically adjusted. An alarm or a warning can be set off if irregular objects or abnormal motions are detected. The image database is updated throughout the process.

It is expected that such a system will be complicated and will require a lot of computation. Implementation methods, therefore, are very important. Three approaches are available: FPGA hardware enhancement, high performance parallel machines using the MPI [136] or PVM [137] libraries, and graphics processors (using the OpenGL shading language [121]). The latter two methods are investigated in this research.

1.3 Overview of the Dissertation

An overview of the dissertation is provided in this section. This overview corresponds to the three motivating areas described in section 1.2: image processing systems, image processing methods, and applications. The structure of the dissertation is provided at the end of this section.

An integrated image processing system has been developed. The framework of the system utilizes Perl [151] to coordinate the interactions between its components, including a web interface, an image database, image processing algorithms and efficient computation tools (Section 4.6). Images are acquired using IP cameras for the system (Section 6.2.1). The developed image processing methods include a new geometric snake model [109] (Section 3.1), an integrated framework for curve evolution and anisotropic diffusion [108] (Section 3.2), efficient bottom-up segmentation methods [111] [112] [107] [113] (Section 3.3), a preferential image segmentation method using mathematical morphologies and its performance evaluation (Chapter 4), and image segmentation methods based on matching distributions (Chapter 5). A detailed introduction to these methods is presented below. Efficient implementation algorithms for image segmentation (Section 3.5), including narrow-band methods, multi-scale methods and parallel methods, are incorporated into the system. A parallel implementation of image mixing using graphics processors (Section 6.1) shows the potential of graphics processors, which may be incorporated into the system.

The proposed methods can be categorized into four groups: PDE methods, variational methods, morphological methods, and probabilistic methods. The new geometric snake model developed in this research [109] utilizes both gradient and region information for image segmentation by minimization of an energy functional by solving PDE. This method works well in the presence of weak edges. An integrated framework for curve evolution and anisotropic diffusion [108] has been developed using PDEs and variational methods. This method outperforms [109] in some applications because it is more stable in the presence of noise. Efficient bottom-up segmentation methods [111] [112] [107] [113] were developed that minimize a piecewise constant Mumford-Shah functional; these are variational methods. A preferential image segmentation method integrates image segmentation and object

recognition using the tree of shapes [14], which is an approach to the mathematical description of morphology. A probabilistic curve evolution method using particle filters is derived in Chapter 5. This is a variational method that minimizes the Kullback-Leibler distance between image distributions for preferential image segmentation.

The characteristics of the above methods vary. The proposed PDE and variational methods [109] [108] [111] [112] [107] [113] are mathematically solid and numerically stable, but they are usually computationally intense. The probabilistic curve evolution method presented in Chapter 5 provides a new perspective of curve evolution in terms of a state space model, and utilizes particle filters for its implementation. This method has a straightforward extension to video tracking, but it is computationally intense. To the author's knowledge, the best method proposed in this dissertation is the preferential image segmentation method using the tree of shapes. This method rests upon a solid mathematical background, is invariant to similarity transformations, and is computationally efficient.

The proposed system and the methods therein may be applied to various applications, including medical image processing, facial image processing and video tracking. Experimental results are presented in Chapter 3, Section 4.5 and Section 6.2.

This dissertation is organized as follows. A literature review is provided in Chapter 2, covering prior work on integrated image processing systems, theoretical image processing methods and their applications. An overview of the details of curve evolution, anisotropic diffusion, and the Mumford-Shah model is provided. The proposed curve evolution methods are presented in Chapter 3. Most of the material presented in Chapter 3 has been published and includes the new geometric snake model which uses both gradient and region information [109] (Section 3.1), an integrated framework for curve evolution and anisotropic diffusion [108] (Section 3.2), and efficient bottom-up segmentation methods [111] [112] [107] [113] (Section 3.3). Preferential image segmentation methods using mathematical morphologies are proposed in Chapter 4. Chapter 4 also includes a thorough performance evaluation of the preferential methods using large image databases and a user interface to the implementation. Preferential image segmentation methods based on matching distributions are presented in Chapter 5. A summary of the work is provided in Chapter 6. The probabilistic curve evolution model using particle filters [110] is also presented in Chapter 6. As an example for future research, a parallel implementation of image mixing using graphics processors is shown in Section 6.1. The possibility of video tracking using the preferential segmentation methods and IP cameras is illustrated in Section 6.2.

Chapter 2

Literature Review

A review of the literature on image segmentation and related areas is provided in this chapter. The review includes four parts. The **first** part reviews the integration of image processing stages. Three topics are reviewed: integration between imaging and image processing, integration between image denoising and image segmentation and integration between image segmentation and recognition. The **second** part reviews general techniques in PDEs, wavelets, variational methods and stochastic methods. Details of the state-of-art PDE techniques for image segmentation and shape representation, namely, curve evolution, anisotropic diffusion and the Mumford-Shah Model, which are closely related to and broadly applied in this dissertation, are provided in this part. The **third** part reviews the applications to image processing using the techniques reviewed in the first two parts. The applications include preferential image segmentation, shape representation and recognition, video tracking, and their efficient implementations using parallel machines and graphic processors. Since these areas are broad and deep, only those methods and applications related to image processing (more specifically, image segmentation) and its applications are reviewed. The review is oriented to give an overall view of image processing techniques and application and directs the interested reader toward the literature for additional details.

2.1 Integrated Image Processing

Three aspects of the prior work toward integration between image processing stages are reviewed: integration between imaging and image processing, integration between image denoising and image segmentation and integration between image segmentation and recognition.

2.1.1 Integration between Imaging and Image Processing - Computational Photography

The integration between imaging and image processing introduces the area of computational photography, which is also called computational imaging. Image processing techniques may be applied to imaging using one single camera or camera arrays.

Computational cameras [32] [101] [102] have been developed by applying image processing techniques to the design of one camera. Image processing may be applied by inventing new optics to process the illumination before the light arrives at the detector (e.g. charge-coupled device (CCD)) of the camera. It may also be utilized after the objects are detected

by the detector and before the image is formulated. The new optics and the computation step provide more flexibility for imaging applications. Computational cameras may be designed to have larger dynamic range, wider spectrum, larger depth of field, higher signal-to-noise ratio and programmable illumination [101]. Complicated movements can be captured using the moment camera in [32]. Nayar *et al* [102] built a programmable imaging system which can be utilized for range imaging, feature detection and object recognition. The flexible cameras in the designed system have adjustable field of views and resolutions by utilizing a digital micro-mirror device (DMD).

An array of cameras [42] [89] utilizing image processing techniques may provide more powerful functionalities, such as 4D light field capture, light field rendering, multi-perspective panoramas, and synthetic aperture photography [89]. A captured scene in an image may be re-illuminated later in virtual cinematography [42], which is very useful in film making and other related areas.

The area of computational photography utilizes geometric image processing methods to improve the performance of imaging. To date it has not utilized prior information during imaging: No recognition methods have been utilized. If recognition methods are applied during imaging to utilize prior information, selective imaging or preferential imaging may be achieved. We may only need to capture objects of interests, thus reducing the challenges for future processing.

2.1.2 Integration between Image Denoising and Image Segmentation - the Mumford-Shah Functional

The Mumford-Shah model [100] is one of the most widely studied mathematical models for image processing. The Mumford-Shah model is a famous example of the current trend in image processing, the combination of image restoration/denoising and segmentation, which is suggested by neuroanatomical [73] and psychophysical [85] evidence. Instead of restoring (or denoising) an image first and then segmenting it, the Mumford-Shah model tries to achieve both tasks simultaneously in a variational framework.

The model treats an image as a function defined on a compact space, and applies techniques from differential geometry to acquire a piecewise smooth segmentation. The model attempts to find image segments with all of the following properties: (1) They approximate the original image, (2) they have small variations in regions, and (3) they require short boundaries between regions. The Mumford-Shah model shown in Eq. 2.1 produces piecewise smooth image segments while its well-posed special case shown in Eq. 2.2 generates piecewise constant segments.

The Mumford-Shah model is formulated as follows. Let I_0 be a function representing the image to be segmented and I be a differentiable function (except for region boundaries) representing the segmented image. Both I_0 and I are defined on a planar domain R. Let R_i be disjoint connected open subsets of R with piecewise smooth boundaries, and let Γ be the union of the portions of the boundaries of R_i inside R. Then the Mumford-Shah functional is defined as

$$E(I,\Gamma) = \mu^2 \iint_R (I - I_0)^2 dxdy + \iint_{R-\Gamma} \|\nabla I\|^2 dxdy + \nu|\Gamma|, \tag{2.1}$$

where $|\Gamma|$ represents the total length of Γ, and μ and ν are positive constants. The first term in Eq. 2.1 penalizes the integrated square error between original image I_0 and the segmented Image I; the second term in Eq. 2.1 penalizes the variation of I in each region;

the third term in Eq. 2.1 penalizes boundary length. The constants μ and ν define the relative weight according to each penalty term. All three terms work together to make the functional meaningful.

The segmentation I in the functional in Eq. 2.1 is generally required to be piecewise continuous. In most cases, a special case of Eq. 2.1, in which I is restricted to be piecewise constant, is applied. The special case of the Mumford-Shah functional, is

$$E(\Gamma) = \Sigma_i \iint_{R_i} (I_0 - \mathrm{mean_{R_i}(I_0)})^2 \mathrm{dxdy} + \nu|\Gamma|. \tag{2.2}$$

where $\mathrm{mean_{R_i}(I_0)}$ represents the average intensity of image I_0 in domain R_i.

Although the functional in Eq. 2.2 may have a global minimum, it is not convex and may have numerous local minima. This is one of the underlying reasons for the so-called initialization problem of the Chan-Vese models [27], meaning that the segmented images using the Chan-Vese models converge to solutions that depend upon initial conditions.

Several methods have been proposed for the implementation of the Mumford-Shah model, including as variational methods [10], an elliptic approximation method by Γ-convergence [6], and curve evolution methods [27] [55] [146] [150].

Among these methods, the curve evolution methods are mathematically well-founded as shown in [54] [58], and have stable implementations as in [130] [105]. Curve evolution methods drive one or more initialized curves toward the segment boundaries of a segmented image that minimize the Mumford-Shah functional. These curve evolution methods can automatically handle topological changes in the segment boundaries because they utilize a level set description of the boundary. The Mumford-Shah functional combines both region and gradient information in the image, so these curve evolution methods can be made robust to noise and weak edges.

The Chan-Vese models are curve evolution implementations of the well-posed Mumford Shah functional of Eq. 2.2. The bimodal Chan-Vese model [27] applies the functional Eq. 2.2 to bimodal images. The energy functional is a special case with $i = 2$. By means of curve evolution, bimodal images are segmented into two parts, the background and the foreground, which can correspond to images of objects.

The bimodal Chan-Vese model attempts to minimize the energy functional

$$\begin{aligned} F(c_1, c_2, C) = {} & \nu \cdot \mathrm{Length(C)} \\ & + \lambda_1 \iint_{\mathrm{inside(C)}} |I_0(x,y) - c_1|^2 dxdy \\ & + \lambda_2 \iint_{\mathrm{outside(C)}} |I_0(x,y) - c_2|^2 dxdy \end{aligned} \tag{2.3}$$

where I_0 is the original image, C is the evolving curve, and c_1 and c_2 are selected as the average values of pixels inside and outside C, respectively. The coefficients ν, λ_1 and λ_2 are positive constants that determine the relative weights of each component of the functional. Both λ_1 and λ_2 are usually taken as 1. These two parameters are neglected in the following derivations.

The Euler-Lagrange equation for minimizers can be shown to be given by

$$\frac{\partial \psi}{\partial t} = \delta_\epsilon(\psi)[\nu \cdot \kappa - (I_0 - c_1)^2 + (I_0 - c_2)^2], \tag{2.4}$$

where ψ is a level set representation of an evolving curve C_t at time t, which means $C_t = \{(x, y)|\psi(x, y, t) = 0\}$. c_1 and c_2 are the average values of pixels inside and outside C, respectively. κ represents the curvature of the evolving curve. $\delta_\epsilon(\psi) = \epsilon/(\pi(\epsilon^2 + \psi^2))$, and ϵ is a positive constant.

It can be seen from Eq. 2.4 that the evolution of the curve is affected by two terms. The curvature term κ regularizes the curve during evolution. The region term $-(I - c_1)^2 + (I - c_2)^2$ affects the motion of the curve. This term can be interpreted as a competition between the region inside the evolving curve and the region outside the curve.

The bimodal Chan-Vese model is directly applicable only for bimodal images. The multi-phase Chan-Vese model [150] has been proposed for complicated images. In this model, two or more coupled curves evolve simultaneously to segment images with multiple objects. Consider a four-phase Chan-Vese with energy functional defined by Eq. 2.2 and $i = 4$. In the implementation, two coupled curves ψ_1 and ψ_2 evolve according to coupled Euler-Lagrange equations.

Suppose the initial curves divide the image into four regions: $R_{00} = \{\psi_1 < 0, \psi_2 < 0\}$, $R_{10} = \{\psi_1 > 0, \psi_2 < 0\}$, $R_{01} = \{\psi_1 < 0, \psi_2 > 0\}$, $R_{11} = \{\psi_1 > 0, \psi_2 > 0\}$, as shown in Fig. 3.16 (a). Let c_{00}, c_{10}, c_{01}, and c_{11} be the average intensities inside R_{00}, R_{10}, R_{01}, R_{11}, respectively. The curve evolution follows two coupled PDEs [150]

$$\begin{aligned}\frac{\partial \psi_1}{\partial t} &= \delta_\epsilon(\psi_1)\{\nu\kappa_1 - ((I_0 - c_{11})^2 - (I_0 - c_{01})^2)H(\psi_2) \\ &\quad - ((I_0 - c_{10})^2 - (I_0 - c_{00})^2)(1 - H(\psi_2))\}\end{aligned} \tag{2.5}$$

$$\begin{aligned}\frac{\partial \psi_2}{\partial t} &= \delta_\epsilon(\psi_2)\{\nu\kappa_2 - ((I_0 - c_{11})^2 - (I_0 - c_{10})^2)H(\psi_1) \\ &\quad - ((I_0 - c_{01})^2 - (I_0 - c_{00})^2)(1 - H(\psi_1))\}\end{aligned} \tag{2.6}$$

where $\kappa_1 = \nabla \cdot (\frac{\nabla\psi_1}{|\nabla\psi_1|})$ and $\kappa_2 = \nabla \cdot (\frac{\nabla\psi_2}{|\nabla\psi_2|})$ are the curvatures of the evolving curves ψ_1 and ψ_2, and $H(\cdot)$ is the Heaviside function.

It can be seen from Eq. 2.5 that the evolution of ψ_1 determines a boundary comprised of two parts: the part between R_{00} and R_{10} where $\psi_2 < 0$, and the part between R_{01} and R_{11} where $\psi_2 > 0$. The first part evolves due to region competition between R_{00} and R_{10}. The evolution of the second part is driven by region competition between R_{01} and R_{11}. Similar observations can be made for ψ_2. In this manner, the multi-phase Chan-Vese model divides the image into several smaller regions and performs curve evolution based on competitions between these regions.

2.1.3 Integration between Image Segmentation and Recognition

Data-driven image segmentation methods usually utilize low-level image features such as color and texture to segment objects in an image. The methods in this category so far can only segment "cartoon"-like images. It is difficult to segment complex images captured from complicated scenes. The current focus of research is therefore the combination of unsupervised data-driven methods and model-based methods which utilize prior information. With the help of prior information, the performance of image segmentation methods may be improved.

Leventon *et al* utilize intensity and curvature priors [87] and statistical shape priors [88] for level set methods. A probabilistic model is constructed from training images for intensity priors, curvature priors [87] and shape priors [88]. Image segmentation is achieved by maximizing the constructed probability likelihood function which integrates prior information and image data.

Cremers *et al* [39] propose a variational model for recognition-driven image segmentation. The bi-modal Chan-Vese model is improved by adding another variational shape term to its energy functional. The modified energy functional takes the form

$$E_{\text{total}}(\phi) = \nu \cdot \text{Length}(\phi) + \lambda_1 \iint_{\text{inside}(\phi)} |\text{I}_0(\text{x},\text{y}) - \text{c}_1|^2 \text{dxdy} \tag{2.7}$$
$$+ \lambda_2 \iint_{\text{outside}(\phi)} |I_0(x,y) - c_2|^2 dxdy + \alpha \iint_{\Omega} (\phi(x,y) - \phi_0(x,y))^2 dxdy$$

where ϕ represents the evolving curve and ϕ_0 is the prior shape information. The first three terms on the right-hand side correspond to the classical Chan-Vese energy functional. The last term utilizes the prior shape information. The minimization of the last term will make the segmented objects ϕ carry a similar shape as ϕ_0. The ultimate image segments will minimize the classical Chan-Vese functional while keeping the shape close to ϕ_0 by introducing the last term weighted by parameter α in the energy functional. The model in [39] can be extended to work for multiple known shapes in a similarity-invariant formation, which means the segmentation results are invariant to similarity transformations like translation, scale and rotation. The model in [39] provides a typical example of the integration between image segmentation and recognition. More papers [12] [29] [40] [52] [53] [66] [76] [125] [148] are available that relate the integration between segmentation and recognition.

2.2 General Theoretical Methods for Image Processing

Image segmentation is one of the most important and challenging problems in image processing. Its objective is to subdivide images into their constituent regions or objects [57]. Objects of interest are segmented so post-processing can focus on these objects rather than whole images. The information to be processed and stored can therefore be greatly reduced. Image segmentation plays an important role in object recognition, content-based image retrieval and video surveillance.

Early image segmentation methods were quite intuitive and relatively simple. An image is usually processed pixel per pixel, using properties such as discontinuities and/or similarities. The Canny edge detector [18] and watershed methods [57] are good examples.

In the past twenty years, advanced mathematical theories and techniques have been introduced in image processing. Many outstanding image segmentation methods, image restoration methods and their combinations, come from partial differential equations (PDEs) [10] [48] [127], calculus of variations [10], wavelet theory [41] [93] and stochastic methods [90] [157]. These methods usually have solid theoretical foundations and practical applications. The methods rely upon a dynamic model of the segmentation or restoration process, which is driven by data from the image and evolves toward a desired result. An image may be processed at multiple scales and multiple resolutions. The image can be viewed as a function defined on a compact space. Curve evolution [104] [130], anisotropic diffusion [117] [154] and the Mumford-Shah model [100] are good representatives of these methods. These methods are described, with details and examples, in the following sections.

2.2.1 Partial Differential Equations

Curve evolution and anisotropic diffusion are typical partial differential equation methods in image processing. Curve evolution methods are widely used in image segmentation while anisotropic diffusion methods are general nonlinear filtering methods for image denoising. These methods are reviewed in the following two subsections.

Curve Evolution

Curve evolution methods [22] [23] [27] [55] [70] [74] [78] [81] [92] [115] [133] [146] [150] [160] [163] are widely used in image segmentation problems. These methods drive one or more initial curve(s), based on image gradient and/or region information, to the boundaries of objects in an image. These methods are derived using variational methods, and are implemented using finite difference approximations to PDEs and level sets [130] [105]. Theories of curve evolution and their implementation are provided in this section. Applications of curve evolution methods in image segmentation are demonstrated in section 2.3.1.

Theories and Implementations of Curve Evolution

The simplest curve evolution algorithm is the evolution of a simple closed curve (a closed curve with no self-intersections) driven by its curvature in its normal direction. Let $C = C(p,t)$ be a family of parameterized smooth closed planar curves, where t represents time and p the curve parameter. The parameter p takes values in $[0,1]$. Assume $C(0,t) = C(1,t)$ and $C_p(0,t) = C_p(1,t)$, restricting the family to closed curves. Let $\mathcal{N}$ represent the inward unit normal to the curve. Let κ be the Euclidean curvature and s the Euclidean arc length, which satisfies $ds = \|\partial C/\partial p\| dp$. Then the evolution of curve C can be represented by the model

$$\frac{\partial C}{\partial t} = \kappa \mathcal{N} \tag{2.8}$$

The evolving curve C at time t can also be implicitly represented by the zero level set of a function ψ of three variables, i.e.,

$$C = \{(x,y) : \psi(x,y,t) = 0\} \tag{2.9}$$

where (x,y) is a point in the image domain.

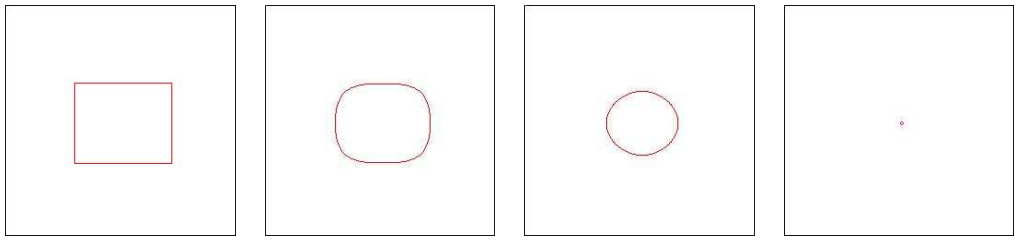
As will be shown later, this representation allows topological changes as the function ψ evolves, and this is a significant advantage. Using this representation, the curve evolution model in Eq. 2.8 can be formulated as

$$\frac{\partial \psi}{\partial t} = \mathrm{div}\left(\frac{\nabla \psi}{|\nabla \psi|}\right)|\nabla \psi| \tag{2.10}$$

where div represents the divergence operator, which is defined as $\mathrm{div}(\nabla \mathrm{f}) = \frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{f}}{\partial \mathrm{y}^2}$ for a general function f with two parameters x and y.

The curve evolution model in Eq. 2.8, which is also called curvature deformation, has many important properties. Let $C_0(s) = (x_0(s), y_0(s))$ be the initial simple closed curve. Then it can be proved [54] [80] that the coordinates of C satisfy

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial s^2} \quad \text{with} \quad \mathrm{x(s,0) = x_0(s)} \qquad \frac{\partial \mathrm{y}}{\partial \mathrm{t}} = \frac{\partial^2 \mathrm{y}}{\partial \mathrm{s}^2} \quad \text{with} \quad \mathrm{y(s,0) = y_0(s)} \tag{2.11}$$



(a) (b) (c) (d)

Figure 2.1: Curvature deformation of a rectangle under Eq. 2.8. (a) Original curve of a rectangle, which corresponds to the inner edge of the black rectangle. (b) Curve evolution after 200 iterations. (c) Curve evolution after 700 iterations. (d) Curve evolution after 1360 iterations.

This is the geometric heat equation. From Eq. 2.11, curve evolution under Eq. 2.8 can be seen as a nonlinear analog to Gaussian smoothing.

From the properties of Gaussian smoothing, it can be expected that a curve shrinks to a point during curve evolution. This has been proved using techniques from differential geometry [54] [58]. The author of [54] proved that the heat equation shrinks convex plane curves into points. It is further proved in [58] that any embedded curve shrinks to a 'circular' point under the geometric heat equation.

The properties of curve evolution have been explored in [54] and [58]. Let T be the unit tangent vector and N be the inward normal vector as before. Define $v = \sqrt{(\frac{\partial x}{\partial t})^2 + (\frac{\partial x}{\partial t})^2} = |\frac{\partial C}{\partial p}|$. The length of the evolving curve is $L = \int_0^1 v dp = \int_0^1 |\frac{\partial C}{\partial p}| dp$. Let θ be the angle between the tangent vector T and the x axis and A represent the area enclosed by the simple closed curve C. Then the properties in the following lemma, reproduced from [54], hold for the evolutions of both convex and non-convex curves:

Lemma 2.2.1. *Properties of curve evolution.*

- *The derivative of v with respect to t is $\frac{\partial v}{\partial t} = -\kappa T$.*
- $\frac{\partial L}{\partial t} = -\int \kappa^2 ds$.
- $\frac{\partial}{\partial t}\frac{\partial}{\partial s} = \frac{\partial}{\partial s}\frac{\partial}{\partial t} + \kappa^2 \frac{\partial}{\partial s}$.
- $\frac{\partial T}{\partial t} = \frac{\partial k}{\partial s} N$ *and* $\frac{\partial N}{\partial t} = -\frac{\partial k}{\partial s} T$.
- $\frac{\partial \theta}{\partial t} = \frac{\partial \kappa}{\partial s}$ *and* $\frac{\partial \theta}{\partial s} = \kappa$.
- $\frac{\partial \kappa}{\partial t} = \frac{\partial^2 \kappa}{\partial s^2} + \kappa^3$.
- $\frac{\partial A}{\partial t} = -2\pi$.

This lemma shows dynamic properties of curves with respect to time under the influence of their curvatures. These properties may represent the information contained in the original curve C_0. Therefore, these properties may be utilized to represent the shape of a simple closed curve, and a potential shape representation method may be derived from curve evolution.

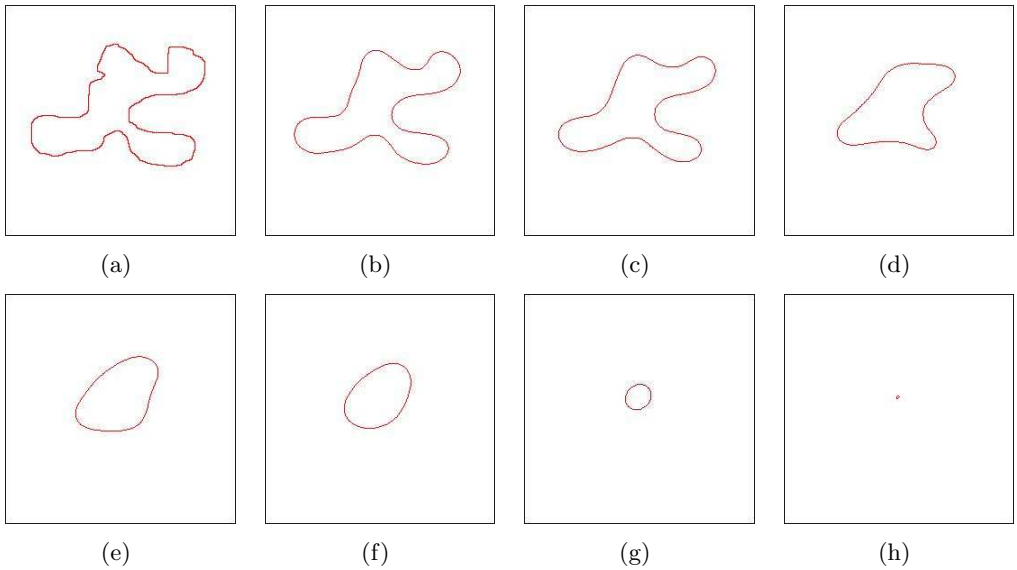


Figure 2.2: Curvature deformation of an irregular concave curve under Eq. 2.8. (a) Original curve of an irregular concave curve. (b) Curve evolution after 100 iterations. (c) Curve evolution after 200 iterations. (d) Curve evolution after 500 iterations. (e) Curve evolution after 800 iterations. (f) Curve evolution after 1000 iterations. (g) Curve evolution after 1500 iterations. (h) Curve evolution after 1614 iterations.

Fig. 2.1 and Fig. 2.2 show examples of curve evolution under curvature deformation Eq. 2.8. In Fig. 2.1, a rectangle, which is convex, shrinks to a point under curvature deformation Eq. 2.8. An irregular concave curve in Fig. 2.2 also shrinks to a 'circular' point under Eq. 2.8.

Things become more complicated when the evolution of a simple closed curve is driven by an external force (the gradient and/or region information in an image, for example). Consider the following curve evolution model

$$\frac{\partial C}{\partial t} = \upsilon \mathcal{N} \tag{2.12}$$

where υ is a constant that determines the speed of curve evolution. If the curve C is represented using a level set function ψ, as in Eq. 2.9, the curve evolution model becomes

$$\frac{\partial \psi}{\partial t} = \upsilon |\nabla \psi| \tag{2.13}$$

Singularities, as shown in Fig. 2.3(b), may develop [80] [130] [105] [104] when a curve is evolved according to Eq. 2.13. A classical solution can not be found in general in this case. A weak solution can be found using Hamilton-Jacobi equations for the problem defined by Eq. 2.12 [130]. Hyperbolic conservation laws and an entropy condition are utilized here.

Fig. 2.3 illustrates the development of singularities and their weak solutions. Fig. 2.3(a) represents the original image which shrinks under the deformation of Eq. 2.13. Singularities (or shocks) develop after several iterations of curve evolution, as can be seen from Fig. 2.3(b). Fig. 2.3(b) (the red evolved curve) looks very different from the desirable curve without singularities, which is shown in Fig. 2.3(c).

(a) (b) (c) (d)

Figure 2.3: Singularities may develop under the deformation of Eq. 2.13. (a) Original curve which shrinks in its normal direction. (b) Singularities develop after several iterations. (c) The desirable curve evolution result, which can be acquired using the Hamilton-Jacobi equations and the Entropy condition. (d) Information loss in curve evolution. The original curve can't be acquired if evolving in the opposite direction without special processing.

A stable implementation of the evolution model Eq. 2.12 is proposed in [105] [130] to avoid the development of singularities. The key idea is the utilization of an *entropy condition* [105] [130]: "Imagine the boundary curve as a source for a propagating fire flame, then the expanding flame satisfies the requirement that once a point in the domain is ignited by the expanding front, it stays burnt". The desirable result in Fig. 2.3(c) can be obtained using this condition.

The work from Osher and Sethian [105] [130] gives a finite difference algorithm to solve Eq. 2.12:

$$\psi_{k+1} = \psi_k + \triangle t\{\max(\upsilon, 0)\nabla^+ + \min(\upsilon, 0)\nabla^-\} \tag{2.14}$$

where

$$\nabla^+ = \{\max(\mathrm{D}^{-\mathrm{x}}, 0)^2 + \min(\mathrm{D}^{+\mathrm{x}}, 0)^2 + \max(\mathrm{D}^{-\mathrm{y}}, 0)^2 + \min(\mathrm{D}^{+\mathrm{y}}, 0)^2\}^{1/2} \tag{2.15}$$

and

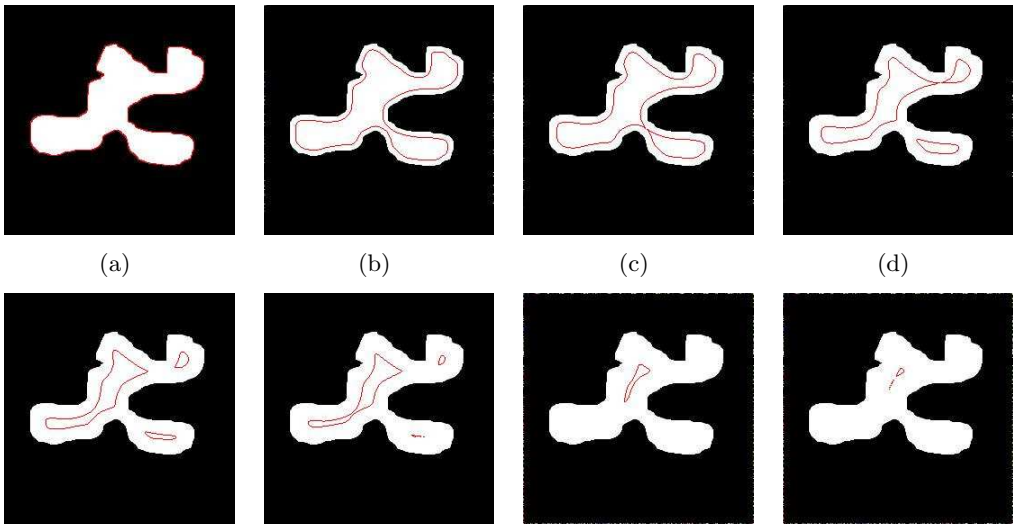
$$\nabla^- = \{\max(\mathrm{D}^{+\mathrm{x}}, 0)^2 + \min(\mathrm{D}^{-\mathrm{x}}, 0)^2 + \max(\mathrm{D}^{+\mathrm{y}}, 0)^2 + \min(\mathrm{D}^{-\mathrm{y}}, 0)^2\}^{1/2} \tag{2.16}$$

in which D^{-x}, D^{+x}, D^{-y} and D^{+y} are the backward and the forward finite differences of ψ in the x and y directions, respectively.

As can be expected, curve evolution under the entropy condition mentioned above loses information and therefore is an irreversible process. This can be seen from Fig. 2.3(d). If the result in Fig. 2.3(c) evolves in the opposite direction in its normal direction, it can not evolve backward into the original curve as in Fig. 2.3(a) without additional information.

The models of Eq. 2.8 and Eq. 2.12 are usually combined in applications of curve evolution, i.e., the deformation of the curve is affected by both its curvature and an external force. The curvature term keeps the evolving curve smooth at all times, while the external force (gradient and/or region information from an image in image segmentation) drives the curve to the correct position.

Fig. 2.4 shows the deformation of the irregular concave curve under $\partial C/\partial t = (1+\kappa)N$. In this figure, the concave curve evolves under the combination of a constant speed and the curvature term. Most models shown so far, however, examine the properties of a simple closed curve. It has been shown in [3] that the properties of the shapes of more complicated objects (e.g., an object with a hole) can be examined using curve evolution by



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.4: Curvature deformation of an irregular concave curve under $\partial C/\partial t = (1+\kappa)N$. (a) Original curve of an irregular concave curve. (b) Curve evolution after 30 iterations. (c) Curve evolution after 45 iterations. (d) Curve evolution after 77 iterations. (e) Curve evolution after 100 iterations. (f) Curve evolution after 118 iterations. (g) Curve evolution after 160 iterations. (h) Curve evolution after 181 iterations.

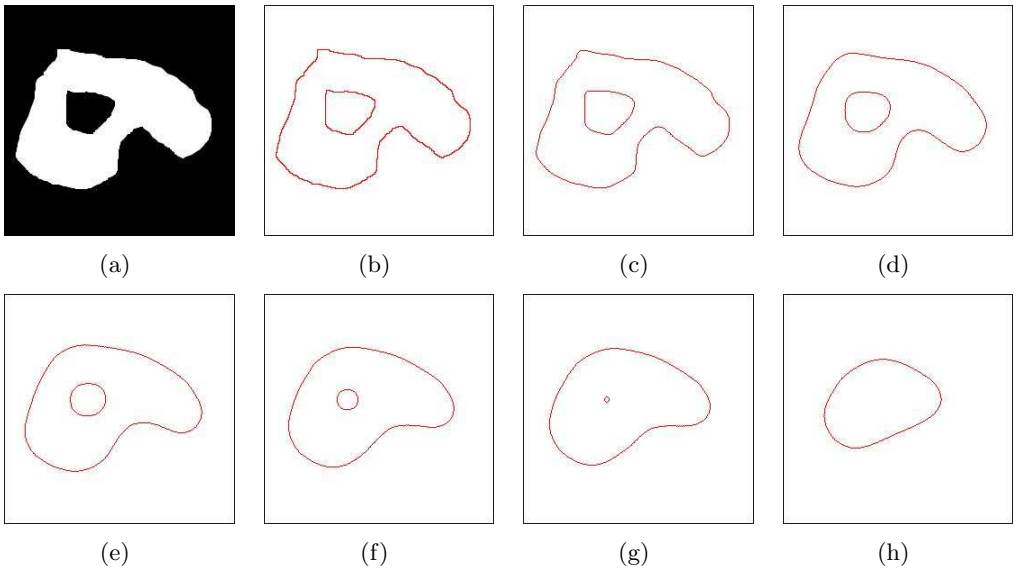


Figure 2.5: Curvature deformation of an irregular concave curve under $\partial C/\partial t = f(\kappa)N$, where the speed function $f(\kappa)$ is defined in Eq. 2.17. (a) A white object with a hole inside. (b) Original curve from the shape of a complicated object. (c) Curve evolution after 10 iterations. (d) Curve evolution after 100 iterations. (e) Curve evolution after 300 iterations. (f) Curve evolution after 500 iterations. (g) Curve evolution after 600 iterations. (h) Curve evolution after 1300 iterations.

carefully designing a speed function. Fig. 2.5 shows the curve evolution of an object under $\partial C/\partial t = f(\kappa)N$ where $f(\kappa)$ depends on the sign of curvature κ and takes the following formation

$$f(\kappa) = \begin{cases} 1+\kappa & \text{if } \kappa \text{ is larger than } 0 \\ -1+\kappa & \text{if } \kappa \text{ is smaller than or equal to } 0 \end{cases} \tag{2.17}$$

It can be seen from Fig. 2.5 that the outside curve and the inside curve of the white object in the image evolve differently according to the function $f(\kappa)$. This provides a way to study the shapes of complicated objects by means of curve evolution.

Anisotropic Diffusion

Noise is an important issue for active contour models, as its presence can substantially degrade performance. For models using region information, image denoising is usually an independent step. Gaussian filters can be used to smooth the image. For those models using gradient information, a Gaussian smoothing method is selected to reduce the effects of noise when calculating the gradients. In this process, however, the boundaries are also smoothed, which makes it difficult to find the boundary locations and encourages boundary leakage. Anisotropic diffusion [117] is a good solution for this denoising problem. It prefers intra-region smoothing to inter-region smoothing, and is thus able to smooth the noise while keeping the boundaries from being smoothed. Anisotropic diffusion models are introduced below.

The application of Gaussian filtering to an image I is equivalent to diffusing this image using the heat equation [82] [158]

$$\frac{\partial I}{\partial t} = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2.18}$$

This generates a family of images with different scales.

Anisotropic diffusion was first introduced by Perona and Malik [117]. A family of smoothed images $I(t)$ are generated from an initial image I_0 by solving the partial differential equation

$$I_t = \mathrm{div}(\mathrm{g}(|\nabla \mathrm{I}|)\nabla \mathrm{I}) = \mathrm{g}(|\nabla \mathrm{I}|)\nabla^2 \mathrm{I} + \nabla \mathrm{g} \cdot \nabla \mathrm{I} \tag{2.19}$$

where $g(|\nabla I|)$ is designed to preferably smooth pixels inside a region rather than pixels near the boundary. $g(|\nabla I|)$ is usually selected as:

$$g(|\nabla I|) = \exp(-(|\nabla \mathrm{I}|/\mathrm{K})^2) \tag{2.20}$$

or

$$g(|\nabla I|) = \frac{1}{1 + (|\nabla I|/K)^2} \tag{2.21}$$

where K is a constant that is tuned for a particular application.

You *et al.* [162] showed that the Perona-Malik anisotropic diffusion is equivalent to minimizing the following functional using a gradient descent method:

$$E(I) = \iint_{\Omega} \frac{\nabla I}{1 + (|\nabla I|/K)^2} dxdy \tag{2.22}$$

This paper also shows that the Perona-Malik anisotropic diffusion method is ill-posed in the sense that images close to each other are likely to diverge during the diffusion process. The method can also form density steps near boundaries, termed stair-casing [156].

A theoretical analysis of the Perona-Malik method has shown that a weak solution is not guaranteed to exist [77]. In practice, however, this method usually performs very well. Weickert [155] and You [161] have examined the differences between continuous and discrete anisotropic diffusion. They show that discrete anisotropic diffusion is well-posed even if its continuous counterpart is ill-posed. The performance of discrete anisotropic diffusion, however, depends on finite difference schemes and grid sizes.

Several methods have been proposed to solve these problems. Alvarez *et al.* [4] proposed a well-posed method to selectively smooth the pixels in the image in its tangential direction

$$I_t = g(G * |\nabla I|)|\nabla I| \mathrm{div}\left(\frac{\nabla \mathrm{I}}{|\nabla \mathrm{I}|}\right) \tag{2.23}$$

where G is a Gaussian kernel and $G * |\nabla I|$ is a smoothed approximation of ∇I with reduced noise. This method, together with the Perona-Malik method, makes use of local information only, which provides poor estimation of the gradient and its direction. Global information can be used to improve the methods, such as the Gabor method in [20]. Several models have been proposed recently to improve the diffusion performance, such as [30] [131] [142] [146].

Shah [131] proposed a common framework for curve evolution, segmentation and anisotropic diffusion, in which images are segmented by minimizing the following energy functional:

$$E(I, \upsilon) = \iint_\Omega \alpha \frac{|\nabla I|}{1 + \alpha\rho|\nabla I|^2} dxdy \tag{2.24}$$

$$+ \iint_\Omega \{\beta|I - I_0| + \frac{\rho}{2}|\nabla I|^2 + \frac{\upsilon^2}{2\rho}\} dxdy \tag{2.25}$$

where an edge function $\upsilon = 2\alpha\rho|\nabla I|^2/(1 + 2\alpha\rho|\nabla I|^2)$ is used to measure the existence of an edge. α and ρ are positive constants. The model is implemented using coupled PDE's. It can smooth images with strong noise and handle triple junctions automatically. The anisotropic diffusion method in this continuous model, however, is ill-posed.

Chen [30] also proposed a model based on energy minimization, whose energy functional is:

$$E(I) = \iint_\Omega \frac{|\nabla I|}{1 + |\nabla G_\sigma * I|^2/K} dxdy \tag{2.26}$$

$$+ \iint_\Omega \beta(I - I_0)^2 dxdy \tag{2.27}$$

In this model the existence and uniqueness of a solution can be guaranteed. But experimental results using this model demonstrate that stair-casing problems still exist. The combination of curve evolution and anisotropic diffusion may provide a solution for this problem, as is shown in Section 3.2.

2.2.2 Wavelet Methods

In many applications, given a signal $f(t)$, one is interested in its frequency content locally in time, i.e., the time-frequency localization. Wavelet methods [41] [93] are very powerful tools for this problem. Wavelet methods generalize the windowed Fourier transform to achieve time-frequency localization. The wavelet transform of a signal $f(t)$ takes the form

$$(Tf)(a, b) = |a|^{-1/2} \int f(t)\psi(\frac{t - b}{a})dt \tag{2.28}$$

where a and b are parameters representing scale and translation respectively, $\psi(t)$ is a function called the "mother wavelet" which satisfies $\int \psi(t)dt = 0$. By proper design of the mother wavelet $\psi(t)$ and choice of parameters a and b, time-frequency localization can be achieved in different applications.

A classical choice of the mother wavelet $\psi(t)$ is the Haar function, which is defined by

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \le x < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

The Haar function can be used to construct orthonormal wavelet bases for multi-resolution analysis. Define $\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n)$. Then as proved in [41], $\psi_{m,n}$ is orthonormal. Furthermore, any L^2 function f can be approximated, up to arbitrary small precision, by a finite linear combination of the function $\psi_{m,n}$. However, the Haar function does not have good time-frequency localization [41].

The wavelet transform has two types: a continuous wavelet transform and a discrete transform, whose details can be found in several well-known books [41] [57] [93]. The Matlab wavelet toolbox is a very useful tool to apply wavelet methods [68]. In image segmentation, wavelet methods are usually applied to extract features from images. The combination of wavelet methods and level set methods [12] may provide good segmentation results.

2.2.3 Variational Methods

Variational methods formulate the problem to be solved as the minimization of an energy functional. The key idea in variational methods is to design an energy functional whose minimization is well-posed and will achieve the objective. The partial differential methods introduced in Section 2.2.1 and 2.3.1, the Mumford-Shah model [100] in Section 2.1.2 and the integrated methods in Section 2.1.3 all have corresponding variational methods. Other variational methods [10] [11] [16] [83] [122] have been proposed for image processing. One of the typical variational methods is the total variation [122] method for image denoising.

The total variation method [122] removes noise in an image u_0 by minimizing a L_1 norm-based energy functional

$$F(u) = \int_\Omega \sqrt{{u_x}^2 + {u_y}^2}\ dxdy \tag{2.30}$$

subject to two constraints:

$$\int_\Omega u\ dxdy = \int_\Omega u_0\ dxdy \tag{2.31}$$

and

$$\int_\Omega (u - u_0)^2\ dxdy = \sigma^2 \tag{2.32}$$

where σ is a constant representing a prior information that the standard variation of the noise in u_0 is σ. The Euler-Lagrange equation for the energy functional is

$$\frac{\partial}{\partial x}(\frac{u_x}{\sqrt{{u_x}^2 + {u_y}^2}}) + \frac{\partial}{\partial y}(\frac{u_y}{\sqrt{{u_x}^2 + {u_y}^2}}) - \lambda_1 - \lambda_2(u - u_0) = 0 \quad in\ \Omega \tag{2.33}$$

with $\frac{\partial u}{\partial n} = 0$ on the boundary $\partial\Omega$ of Ω. λ_1 and λ_2 are usually constants. The Euler-Lagrange equation (2.33) may be solved using the gradient descent method by means of a parabolic differential equation, where λ_1 is set to 0 and λ_2 is adjusted adaptively as in [122]. A limitation is that the method in [122] works only for gray images. Blomgren *et al* [16] generalized it to color images by proposing color total variational (Color TV) methods.

Variational methods provide a very useful mathematical tool for integrating various methods, as the energy functional (2.7) shown in Section 2.1.3. However, the functional (2.7) only integrates prior shape information with low-level features. In the proposed system, more prior information such as color and texture will be integrated for better segmentation results.

2.2.4 Stochastic Methods

Recent novel applications of stochastic methods in image processing can be roughly categorized in three classes. The first class contains classical methods like Markov random fields [90] [157] (applied in image analysis). The second class includes methods based on

stochastic differential equations [149], such as stochastic active contours [8] [72]. The third class contains particle filtering methods [9] [43] which are based on Bayesian tracking and can be utilized in video tracking. Details on particle filtering are provided in Section 5.1.

Stochastic methods have a deep connection with PDE methods. Stochastic methods describe systems microscopically while PDE methods describe systems macroscopically. But both of them may be applied to solve the same variational problem, as shown in L.C. Evens' lecture notes [49]. One example is the stochastic approximation of curve shortening flow using particle systems in [8]. Another example is stochastic active contours proposed in [72], which introduce shape optimization and stochastic differential equations to the level set framework. Stochastic active contours constructed this way are shown to be less sensitive to initialization problems. The connection gives a hint for the choice of appropriate methods for different applications.

2.3 Applications to Image Processing

2.3.1 Curve Evolution in Image Segmentation

Based on the theories and implementation of curve evolution methods in the previous subsection, many models [22] [23] [27] [55] [70] [74] [78] [81] [92] [115] [133] [146] [150] [160] [163] have been proposed for the application of curve evolution in image segmentation problems. These methods introduce dynamics into image segmentation. They evolve one or more curves in an image, subject to constraints derived from the image and the evolving curve(s), to detect objects in the image by stopping the curve(s) at boundaries between objects. These models are usually implemented using variational methods, with partial differential equation solution techniques and level set methods.

Several symbols are defined to describe the models mentioned above. Let $C = C(p,t)$ be a family of smooth closed planar curves, where t represents the time and p the curve parameter. The parameter p takes values in $[0,1]$. Assume $C(0,t) = C(1,t)$ and $C_p(0,t) = C_p(1,t)$, restricting the family to closed curves. Let $\mathcal{N}$ represent the inward unit normal, κ the Euclidean curvature and s the Euclidean arc length, which satisfies $ds = \|\partial C/\partial p\| dp$. By selecting different functionals, several minimizing flows are defined. The basic curve evolution 2.8 corresponds to the minimization of the following length functional:

$$L(t) = \int_0^1 \left| \frac{\partial C}{\partial p} \right| dp \tag{2.34}$$

Caselles *et al.* [22] and Malladi *et al.* [92] proposed a geometric model of active contours, which evolves curves in a Eulerian formulation and is implemented via level set algorithms [130]. More precisely, their model is

$$\frac{\partial \psi}{\partial t} = \phi(x,y)\mathrm{div}\left(\frac{\nabla \psi}{|\nabla \psi|}\right) + \nu|\nabla \psi| \tag{2.35}$$

where ψ is the level set representation of the evolving curve C; $C = \{(x,y)|\psi(x,y) = 0\}$. ϕ is designed to stop the curve at the edges of objects and usually takes the form $\phi = 1/(1 + |\nabla G * I|^2)$, where G represents the Gaussian filter. This model, compared to the classic parametric snake model put forward by Kass *et al.* [74], automatically handles topological changes and detect multiple objects simultaneously.

Kichenassamy et al. [78] proposed minimization of the energy functional

$$L_\phi(t) = \int_0^1 \left| \frac{\partial C}{\partial p} \right| \phi dp \tag{2.36}$$

where ϕ is defined as above. The minimization of this functional leads to the flow

$$C_t = \{\phi\kappa - \bigtriangledown\phi \cdot \mathcal{N}\}\mathcal{N} \tag{2.37}$$

The doublet in the right side of the equation, together with the weight function ϕ, attracts the curve to features of interest. When ϕ equals 1, the flow shrinks any simple closed curves into a point. The minimization of this energy functional has the level set representation:

$$\frac{\partial \psi}{\partial t} = \phi(x,y)\left(\upsilon + \epsilon \cdot \mathrm{div}\left(\frac{\nabla\psi}{|\nabla\psi|}\right)\right)|\nabla\psi| + \nabla\phi \cdot \nabla\psi \tag{2.38}$$

Geodesic active contours are proposed by Caselles *et al* in [23]. Curve evolution is formulated as the computation of geodesics of curves in a Riemannian space whose metric is defined by the image content in this model. The model minimizes the functional

$$E_{GAC}(C) = \oint_C g(C(s))ds \tag{2.39}$$

where g is a function of the evolving curve and s represents the arc length. The level set formulation of the geodesic active contours is

$$\frac{\partial \psi}{\partial t} = |\nabla\psi|\mathrm{div}\left(\mathrm{g(I)}\frac{\nabla\psi}{|\nabla\psi|}\right) = \mathrm{g(I)}|\nabla\mathrm{u}|\mathrm{div}\left(\frac{\nabla\psi}{|\nabla\psi|}\right) + \nabla\mathrm{g(I)} \cdot \nabla\psi \tag{2.40}$$

where I represents the image to be segmented.

Gradient information is used in these models to stop a curve at boundaries of objects, by weighting the evolution speed using a monotone decreasing function of image gradients. Although this works in many cases, the method allows boundary leakage when the image gradient is small. Siddiqi *et al.* [135] provide mathematical derivations of length and area minimizing flows and combines gradient and area information for shape segmentation. Their methods have good performance but still need improvement, because boundary leakage still occurs.

Yezzi *et al.* [160] propose a fully global gradient flow for image segmentation using coupled curve evolution equations. This model provides very good performance for bi-modal and tri-modal images. However, this model requires *a priori* knowledge of the number of regions in the image. Furthermore, when multiple objects (more than 3) exist in the image, the implementation of this method becomes very complicated, because $n-1$ curves are needed for n objects in an image and complex coupling terms are required.

In the papers by Chan and Vese [27], region information is utilized for image segmentation. In their methods, the segmentation problem, as shown in Eq. 2.3, is formulated as the minimization of an energy functional simplified from the Mumford-Shah model [100]. This model is implemented using level set methods and provides very good performance for bimodal images. Vese *et al.* later extended the model to multi-modal images [150]. In their model, however, gradient information is not utilized. Tsai *et al.* [146] implemented the

Mumford-Shah model [100] using curve evolution methods and extended it to image noise reduction, interpolation and magnification.

Paragios *et al* [115] put forward a model of coupled geometric active regions to make use of both gradient and region information in a statistical framework. In this model, each region in the image and each pixel in a region are assumed to be independently distributed. Each region is modeled using a Gaussian probability distribution function. In the first step, the number of regions and the parameters for each region are estimated using the image histogram and the minimum description length method. Then, each region is segmented using a specific level set function. For this multi-phase model, techniques are proposed to make sure that no points are segmented to more than one region and no points are left over.

Xie *et al.* [159] proposed a region-aided geometric snake model, which also makes use of the combination of gradient and region information. The method resolves the boundary leakage problem by using the region information. In the first step of its implementation, a region growing method is applied on the image. The capture range of the results is then extended by using a gradient vector diffusion method. In the last step, the image is segmented using an improved curve evolution model. However, it seems difficult to deal with textured images by applying this method together with the tensor measure.

Kimmel [81] presents a general model that incorporates the alignment term, the geodesic active contour model [23] and the minimal variance criterion [27]. The energy functional to be minimized takes the following form:

$$\begin{aligned} E(C, c_1, c_2) = &- \oint_C |\langle \nabla I, \vec{n} \rangle| ds + \alpha \oint_C g(C(s)) ds \\ &+ \beta \frac{1}{2} \left(\iint_{\text{inside(C)}} (I - c_1)^2 dxdy + \iint_{\text{outside(C)}} (I - c_2)^2 dxdy \right) \end{aligned} \tag{2.41}$$

This functional utilizes both gradient and region information for image segmentation.

Recent developments in active contours include Sobolev active contours [138] [94] [139] and stochastic active contours [72] [149]. Sobolev active contours [139] utilize the Sololev norm [94] instead of the classical H^0 (or L^2) norm to make curve evolution more stable to local variations. Sobolev active contours show impressive results in image segmentation and video tracking [138]. Stochastic active contours [72] introduce stochastic differential equations [149] and shape optimization methods into the level set framework to reduce the effects of the initialization problem.

Curve evolution methods still have problems. First, most have initialization problems, which means different initial curves result in different segmentations for the same image. This is a consequence of the lack of convexity of the energy functional. Second, these methods have difficulty with complicated images having multiple junctions. Top-down hierarchical methods [146] or multiple coupled evolving curves [150] are used to segment multiple junctions. Both top-down hierarchical methods and coupled curve evolution are time-consuming. Techniques are usually required to ensure that no pixels in the image are unsegmented or included in multiple segments.

Several methods are proposed to solve these problems, such as those described [112] and [55]. Gao and Bui [55] illustrated the initialization problem for Chan-Vese models, and proposed a solution using a decoupled multi-phase Chan-Vese model that reduces the computational load. The method in [55] is hierarchically top-down, as in Tsai *et al* [146], so multiple segmentation passes are required for complex images. We provided a mathematical analysis for the initialization problem of the bimodal Chan-Vese model in [112], which

proposed a bottom-up hierarchical method to minimize the well-posed Mumford-Shah functional. The authors also proposed a fast curve evolution method for the bimodal Chan-Vese model. The methods in [112] are efficient, robust in the presence of noise and are able to handle complicated images.

Some preliminary results on the improvement of the application of the Mumford-Shah model have been established. A mathematical analysis for the initialization problem of the Chan-Vese models is provided in Chapter 3. Region competition, which includes region growing as a special case, is incorporated. The improved method keeps the advantages reported in [112] and outperforms the previous method for images with weak edges. The improved method bears some similarities with those reported by Tek and Kimia [143] and by Zhu and Yuille [163], but no *a priori* information or human interaction is necessary. Please refer to Section 3.3 and Section 3.4 for more details.

It can be seen from the methods introduced above that a key issue in curve evolution is how to utilize both gradient and region information in an image. A new geometric snake model has been developed [109], which uses both the gradient and region information for image segmentation. This model is able to segment objects where weak edges exist and does not exhibit boundary leakage. A hierarchical scheme for multi-modal images is also proposed, which helps to segment multi-modal images without prior knowledge of the number of regions. The methods handle color and textured images, and complicated cases such as images with triple junctions, are correctly segmented using this model. Details are provided in Section 3.1.

2.3.2 Shape Representation and Recognition

Shape representation and recognition based on deformation have been two of the most active areas recently since it has been proved in [54] [58] that the heat equation shrinks embedded (i.e. simple and closed) plane curves to "round" points. Theoretical research includes the construction of shape spaces [75] [132], the statistical analysis of shapes [84], the signatures of shapes [17] and the design of shape representation methods [96].

The shape theory introduced in the book [75] by Kendall *et al* explains the properties of shape spaces such as the topological structure, the homology groups, geodesics, the Riemannian structure etc. Sharon and Mumford [132] proposed to represent a two-dimensional shape (i.e., simple, closed 2D curves) by a diffeomorphism of the unit circle to itself (i.e., a differentiable and invertible, periodic function) via conformal mapping. The computations of shapes from diffeomorphisms and vice versa have been put forward, along with a Riemannian norm. The computation of geodesics, however, has not been established, yet.

Calabi *et al* [17] utilize signature curves for shape representation. The signature curves parameterized by (κ, κ_s) are invariant to Euclidean transforms where κ is the Euclidean curvature and κ_s is the first derivative of κ with respect to the arc length s. Similarly, the signature curves parameterized by (κ, κ_s) are invariant to affine transforms where κ is the affine curvature and κ_s is the first derivative of κ with respect to the arc length s. Invariance to Euclidean (or affine) transformations is important because shapes or objects in images need to be recognized regardless of their placement or orientation in the image. Numerically invariant computations of these signature curves are also proposed in this paper. The application of the signature curves in shape recognition and classification, however, is not straightforward and still needs work.

Mokhtarian *et al* [96] proposed the use of Gaussian smoothing, which is equivalent to geometric heat equation (Eq. 2.8) $\partial C/\partial t = \kappa N$, for shape representation. A curvature

scale space (spatial parameter p vs. variance of Gaussian filter σ) is constructed for a simple closed curve $C(p), 0 \leq p \leq 1$, using the points whose curvature is zero $\kappa(p, \sigma) = 0$ during Gaussian smoothing. Gaussian smoothing stops when there is no point with zero curvature, when the shape is very smooth. This idea has been adopted in the MPEG-7 standard for shape representation. An illustration of the idea is available online at **http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html**. However, as can be seen from the illustration, the curve in the end is not a circular point, and the curve then still contains different information for different initial curves. In this sense, shape representation may not be accurate.

Kim *et al* [79] [80] proposed to utilize curve evolution $\partial C/\partial t = (\beta_0 + \beta_1 \kappa)N$ to represent shapes. However, the application of this method to real applications such as content-based image retrieval, is not straightforward. The method proposed in [3] generalized the idea in [79] [80] so that the curve evolution $\partial C/\partial t = f(\kappa)N$ may be driven by a general function $f(\kappa)$ of constants and curvatures κ, as shown in Fig. 2.5. Shape representation based on this curve evolution may represent an object with holes inside.

Applications of shape representation and recognition include image segmentation based on shape priors [87] [88], shape classification [3], recognition [129] and reconstruction [56].

2.3.3 Video Tracking

Video tracking is another important application of image processing algorithms. It has received more attention for security applications such as video surveillances after the attacks of 9/11/2001.

Video tracking is oriented in tracking the motion of one or more objects in a scene. Most video tracking methods assume initial locations of objects in the first video frame, which come from the results of image segmentation, are known.

Video tracking may be formulated as parameter estimation and tracking in some cases, in which the shapes of the objects are to be represented as several parameters. Particle filters [9] [43], which are based on the framework of Bayesian tracking, are widely applied in these cases to estimate and update the parameters, thus tracking the moving objects.

The shapes of objects are important for video tracking in many cases, especially when the objects are under deformation while moving. The objects may split or merge in some cases, i.e., the number of objects may change. Compared to other methods such as Kalman snakes [118] [144], video tracking methods utilizing level set methods [116] are advantageous for these cases, since level set methods are able to handle topological changes of objects automatically. Mean-shift methods [34] are also very popular in this application.

The deformation of areas in an object may need to be tracked, for example, in the case of facial expression modeling. Active appearance models [35] are helpful in these applications. But these models need the prior information of the face and the point-wise correspondence of feature points in the prior face image and the image to be tracked.

Tracking using prior information has received a lot of attention recently, such as in [53]. Prior information of objects to be tracked, in combination of motion information between video frames, may provide better tracking results and more robustness to the algorithm. However, only histogram information for a gray image is utilized in [53]. Color, texture and shape prior information may be utilized for better performance, which will be one of the main topics in this dissertation.

2.3.4 Numerical Implementations of Image Processing Algorithms

Numerical implementations of image processing algorithms using softwares based on PDEs can be categorized as three parts. The first part includes classical PDE numerical solutions such as finite difference methods (FDM), finite volume methods (FVM), finite element methods (FEM) and multi-grid methods. The methods in this category are described in standard texts such as the book written by Heath [65]. The second part contains the numerical implementation of the weak solution of curve evolution [105] and its fast implementation methods such as narrow-band methods [31] and multi-scale methods [116]. Details of the methods and their experimental results in this category are provided in Section 2.2.1 and Section 3.5. The third part contains parallel implementation of algorithms based on the message passing interface (MPI) [136] and the parallel virtual machine (PVM) [137] schemes. Parallel computation may also be achieved by using graphics cards, which will be introduced in the Section 2.3.5. An introduction to MPI and PVM is provided below.

MPI stands for Message Passing Interface [136]. It is a standard for developing parallel programs by means of message passing interface among computational nodes. Parallel computing is achieved and coordinated by passing messages. An implementation of MPI is available in the public domain at the anonymous ftb site info.mcs.anl.gov in pub/mpi. MPI has become an industrial standard and is supported by most vendors of parallel machines.

PVM stands for Parallel Virtual Machine [137]. It is a scheme developed by Oak Ridge National Library (ORNL) for parallel computing in a heterogeneous network of computers. Parallel computation is achieved by building a virtual machine. A daemon runs on every computer to construct the virtual machine and coordinate the computation. It is also available in the public domain and can be acquired from the site netlib2.cs.utk.edu. PVM has been supported by several vendors of parallel machines, such as Cray Research, IBM, Convex, Intel, SGI, and DEC. Although PVM is not so popular as MPI, it is at present more fault-tolerant.

2.3.5 Parallel Image Processing using Graphic Processors

Graphic processors, also known as Graphic Processing Unit or GPUs, provide the most powerful graphics hardware for the price. General purpose computation using graphic processors (called GPGPU for "General Purpose" GPU) has been emerging as a new area of research. Graphic processors have three advantages. First, they are powerful and inexpensive. For example, the NVIDIA GeForce 6600 GT ($127 at amazon.com in 2005) has a memory bandwidth of 16.0GB/sec, a polygon fill rate of 4.0 billion texels/sec, is able to draw 375 million vertices per second and a memory data rate as 1000MHz. Second, the performance of graphics cards is improving quickly, roughly doubling every six months [106]. Almost one year later, the NVIDIA Geforce 6800 advertised a 35.2GB/sec of memory bandwidth, and the NVIDIA Geforce 7950 advertised 76.8GB/sec. Finally, graphic processors are highly parallelized.

Early graphic processors had only fixed-function pipelines, and few options were left for programmers. Recent graphic processors have programmable processing units that support vectorized floating-point operations using IEEE single precision. High level languages like the OpenGL Shading Language (GLSL) [121] [134] have recently emerged to support the programmability of the vertex and fragment pipelines, allowing general-purpose computation.

General-purpose computation using graphics processors usually utilizes the computer graphics pipelines [2] [152]. Grid-based computations are solved by first transferring the data to be processed into texture memory. The GPU then performs computations by rendering and rasterizing graphics primitives that access the texture. A fragment program (also called the kernel) uses texture coordinates, which identify the locations of the data and their neighbors, to read data from texture memory, perform computations, and write the result back to texture memory. More details about the procedure and its generalizations can be found in [86]. The procedure can be applied in matrix computation, differential equations, curve evolution, anisotropic diffusion, video tracking and other applications in computer graphics and computer vision. A thorough review on the computational methods, developing tools and applications is provided in [106].

Although many researchers have worked in the areas of general-purpose computation using graphic processors, the potential of graphic processors has not been thoroughly explored. A deep understanding of computer graphics [2] [152] is necessary to take advantage of the parallel structure of graphic processors and develop efficient programs. An example algorithm is implemented in this dissertation using graphic processors to demonstrate their capability, performance and potential.

2.4 Summary

A review of the literature has been provided in this chapter. The review first shows the current status of integrated image processing, namely the integration between imaging and image preprocessing, the integration between image preprocessing and image segmentation, and the integration between image segmentation and object recognition. The techniques of PDE, calculus of variations, wavelets and stochastic methods are reviewed next and their potential integrations are discussed. Applications of the proposed integrated system and the potentially integrated methods are introduced at last. The review represents the current progress of the research related to this research, thus giving a hint of the necessity and feasibility of the proposed problems, whose solutions will be presented in the next several chapters.

Chapter 3

Image Segmentation using Curve Evolution

New curve evolution methods are presented in this chapter. Most of the material presented in this chapter has been published and includes a new geometric snake model which uses both gradient and region information [109] (in Section 3.1), an integrated framework of curve evolution and anisotropic diffusion [108] (in Section 3.2), and an efficient bottom-up segmentation methods [111] [112] [107] [113] (in Section 3.3). The methods presented in this chapter belongs to the image preprocessing stage and the image segmentation stage in the proposed scheme 1.2.2 in Chapter 1. All of them are formulated as minimizations of energy functionals, and they are solved using partial differential equations.

3.1 Curve Evolution using both Gradient and Region Information

In this section, a new geometric snake model is developed [109], based upon techniques of curve evolution and the utilization of gradient information and region information simultaneously. This model successfully solves the boundary leakage problem. With the help of a hierarchical approach, it can handle complicated cases, such as triple junctions. Furthermore, it supports vector-valued images and can be easily extended to handle color and textured images. Experimental results demonstrate the model's power in image segmentation.

3.1.1 Model Description

The geometric snake model proposed in [135] makes use of the gradient and area information by using a combination of weighted length and area minimizing flow. It has the following form:

$$C_t = \alpha\{\phi_l\kappa - \bigtriangledown\phi_l \cdot \mathcal{N}\}\mathcal{N} + \{\phi_l + \frac{1}{2}\langle C, \bigtriangledown\phi_l\rangle\}\mathcal{N} \tag{3.1}$$

In this model, the doublets and the weight function are designed to stop the evolving curve at the boundaries of interest. The weight function ϕ_l is usually selected as $\phi_l = 1/\{1 + |\bigtriangledown G_\sigma * I|^n\}$. Note that this is a function of gradient and is independent of the evolving curve.

However, as pointed out in [27], the minimal length of a closed curve and the minimal area inside it are correlated by the isoperimetric inequality, i.e.

$$\mathrm{Area(inside((C)))} \leq \mathrm{c} \cdot (\mathrm{Length((C)))}^{\mathrm{N}/(\mathrm{N}-1)}$$

where $N > 1$ is an arbitrary dimension, $N = 2$ for planar images, and c is a constant depending only on N.

This correlation shows that the minimization of the curve length and that of the area inside the curve are somewhat equivalent. Region information is not fully utilized in this way. This may explain why the method in [135], although it improves upon prior models, still exhibits boundary leakage.

To make use of region information in our proposed model, different weight functions are chosen for the length minimizing flow and area minimizing flow. The weight function ϕ_l for length minimization is selected as before to make use of gradient information. The weight function ϕ_r for region minimization, similar to [27], is chosen to minimize the following region functional:

$$R_{\phi_r}(t) = \iint \phi_r dxdy = \int_{\mathrm{inside(C)}} |I(x,y) - c_1|^2 dxdy + \int_{\mathrm{outside(C)}} |I(x,y) - c_2|^2 dxdy$$

where c_1 and c_2 are the average intensities inside and outside the evolving curve C, i.e., $c_1 = \frac{\int_{\mathrm{inside(C)}} I(x,y)dxdy}{\int_{\mathrm{inside(C)}} dxdy}$ and $c_2 = \frac{\int_{\mathrm{outside(C)}} I(x,y)dxdy}{\int_{\mathrm{outside(C)}} dxdy}$.

Here, ϕ_r is selected as a function of the evolving curve to utilize the region information. The level set function can be directly acquired using the Mumford-Shah functional (see [27] for details). Thus the complete level set equation is obtained:

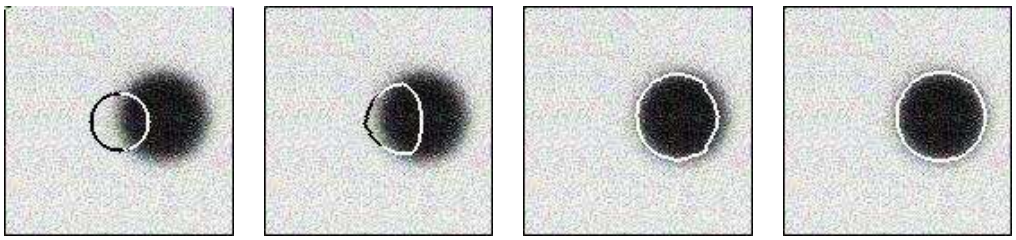
$$\psi_t = \alpha\phi_l(v + \epsilon\kappa)| \bigtriangledown \psi| + (1-\alpha)\delta_\beta(\psi)[(I - c_2)^2 - (I - c_1)^2] \tag{3.2}$$

The first term in the right part of Eq. 3.2 is a combination of constant motion and curvature motion. In this term, v is the inflationary term, which attracts the curve in one direction: either expanding or shrinking; κ represents the curvature, which keeps the curve smooth when evolving. The doublets are not used in this model. The second term in the right part of Eq. 3.2 makes use of the region information, which helps to stop the curve at the boundaries of interest. In this term, $\delta_\beta(x) = \beta/(\pi(x^2 + \beta^2))$ acts like a delta function. α, β and ϵ are positive constant coefficients. α lies between 0 and 1 and determines the weights of the gradient and the region information. When α is set to 1, the model changes to the length and area minimizing flow, as in [135]. When it is set to 0, it becomes similar to the model in [27]. α is usually set to 0.5. For those images where weak edges exist, α is set to be less than 0.5 to increase the weight of the region information.

This proposed model combines gradient and region information. It can be seen to be an extension of the method in [135] by utilizing the region information. It can also be seen as the refinement of the method in [27] by introducing the gradient information.

3.1.2 Hierarchical Approach

The way to utilize the region information shown above is particularly suitable for a bi-model image. To deal with images containing multiple objects, a hierarchical approach is proposed, which is similar to as the one in [146]. In this approach, an image is segmented



(a) (b) (c) (d)

Figure 3.1: Illustration of region information in the model. (a) Original curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final results of curve evolution.

into two sub-images, and then the method is applied on these sub-images respectively. The procedure is implemented iteratively until each region contains only one object.

3.1.3 Extension to Vector-Valued Images

The proposed method is easy to extend to vector-valued images. For a vector-valued image $I : \mathcal{R}^2 \rightarrow \mathcal{R}^m$, the length weight function is selected as:

$$\phi_l = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{1 + |\bigtriangledown G_\sigma * I_i|^n}$$

Accordingly, the region weight function is selected so that the level set function takes the following form:

$$\psi_t = \alpha \phi_l (v + \epsilon\kappa) | \bigtriangledown \psi| + (1 - \alpha)\delta_\beta(\psi) \frac{1}{m} \left[\sum_{i=1}^{m} (I_i - c_{2i})^2 - (I_i - c_{1i})^2 \right]$$

For color images, three channels of information can be directly used in the above formulae. For textured images, preprocessing techniques, such as wavelet or Gabor transforms, can be performed, and the resulting vector-valued images are applied to this model.

3.1.4 Experimental Results

Experimental results from the proposed model are given in this section. The proposed model is implemented using level set methods [130]. In the implementation, the values of the region term is normalized to lie between -1 and 1 so that they are comparable with that of the gradient.

Fig. 3.1 illustrates the role of the region term in Eq. 3.2. The initial curve is set to be a circle in the center of the original image in Fig. 3.1(a). The parameters v and ϵ are set to expand the curve. With the effects of the region information, however, the left part of the curve shrinks and the right part expands, as can be seen clearly from Fig. 3.1(b) and Fig. 3.1(c). The curve finally stops at the correct boundary in Fig. 3.1(d).

In Fig. 3.2, comparisons between the proposed method and the one in [135] are given. The original image is generated by smoothing an ideal circle using a 13-by-13 Gaussian filter. In this figure, Fig. 3.2(a)- Fig. 3.2(d) show the results of our model with α set to be 0.1. Our method correctly segment out the object in the image. In comparison, the results

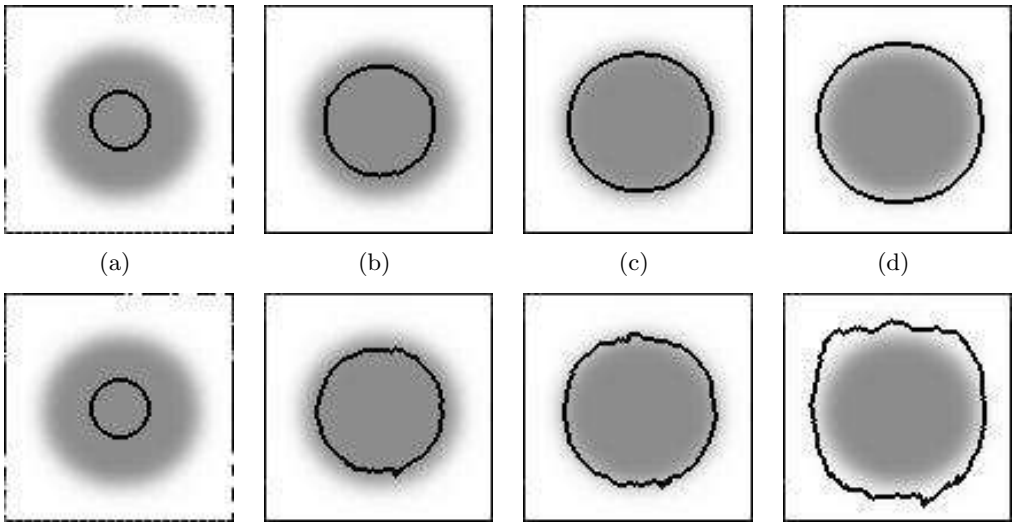
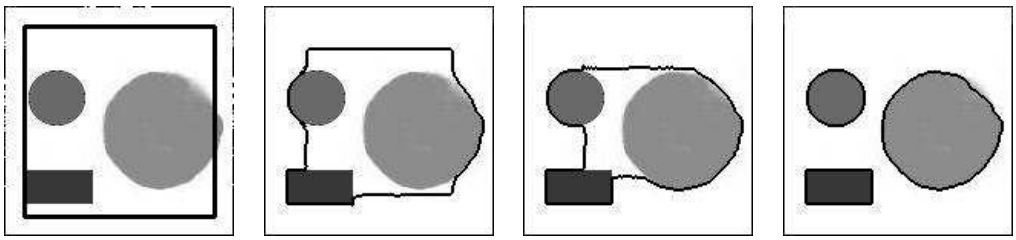


Figure 3.2: Illustration of region information in the model. (a) Original curve. (b) Intermediate results of curve evolution using the proposed model. (c) Intermediate results of curve evolution using the proposed model. (d) Final results of curve evolution using the proposed model. (e) Original curve. (f) Intermediate results of curve evolution using [135]. (g) Intermediate results of curve evolution using [135]. (h) Final results of curve evolution using [135].



(a) (b) (c) (d)

Figure 3.3: Segmentation of multiple objects in an image using the proposed model. (a) An image with an initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final results of curve evolution.

of the method in [135], Fig. 3.2(e)- Fig. 3.2(h), shows that the boundary leakage happens. This example shows that our model performs very well for weak-edge images and improves upon prior methods.

The results of Fig. 3.3 show that our model(with $\alpha = 0.2$) works well for multi-model images. One point for this case is that our model prefers the image boundary for the initial curve.

In Fig. 3.4, a color image with a triple junction is processed. Using the hierarchical approach, three parts in the image are segmented sequentially; see Fig. 3.4(d), Fig. 3.4(g) and Fig. 3.4(h) for details. This example shows that the proposed method can deal with complicated cases successfully.

3.1.5 Conclusions and Future Research

In this section, a geometric snake model is proposed, which makes use of both the gradient and the region information for curve evolution. It successfully solves the boundary leakage problem. Complicated cases, such as multi-model images, color images and images with triple junctions, can be successfully processed. Experimental results shows that the model can be a powerful tool for image segmentation. Extensions to textured images can also be performed; this is a topic for future research.

3.2 Image Segmentation using Curve Evolution and Anisotropic Diffusion: An Integrated Approach

In this section, a new model [108] is proposed for image segmentation that integrates the curve evolution and anisotropic diffusion methods. The curve evolution method, utilizing both gradient and region information, segments an image into multiple regions. During the evolution of the curve, anisotropic diffusion is adaptively applied to the image to remove noise while preserving boundary information. Coupled partial differential equations (PDE's) are used to implement the method. Experimental results show that the proposed model is successful for complex images with high noise.

The proposed curve evolution model, which makes use of both gradient and region information in the image, will be introduced first in this section. The connections between curve evolution and anisotropic diffusion are then examined. The integration of curve

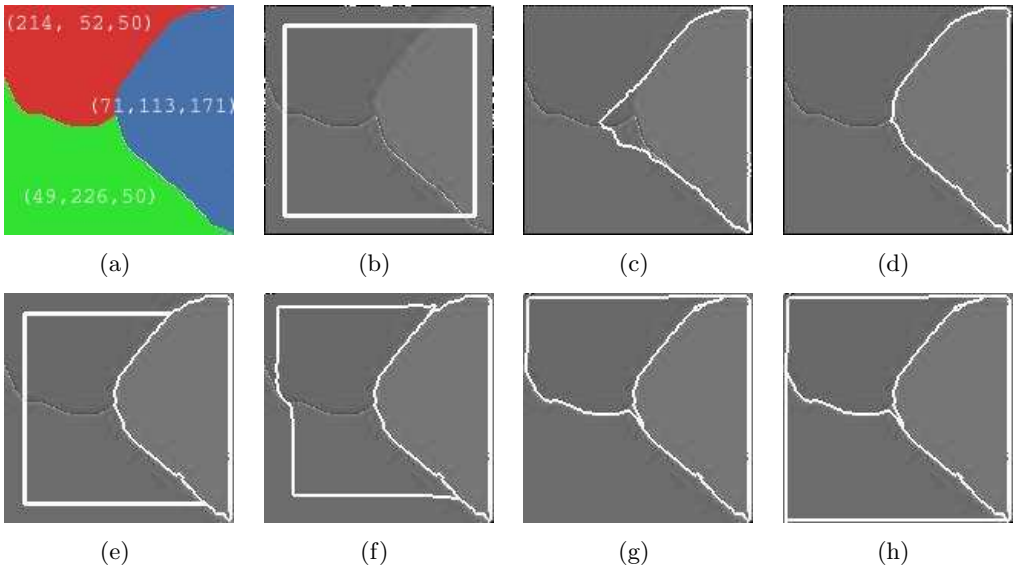


Figure 3.4: Segmentation of an image with triple junctions. (a) A color image with triple junctions. (b) Initial curve. (c) Intermediate results of curve evolution. (d) Segmentation results of the blue region. (e) Initial curve of the second iteration. (f) Intermediate results of curve evolution. (g) Segmentation results of the red region. (h) Final results of curve evolution.

evolution and anisotropic diffusion is shown next. Implementation issues, experimental results and a summary are presented sequentially.

3.2.1 Curve Evolution Model

The proposed model begins with an energy functional that integrates gradient and region information using the form

$$\begin{aligned} E(C) = \alpha \cdot \int_0^1 \left| \frac{\partial C}{\partial p} \right| \phi dp \\ + (1-\alpha)\lambda \iint_{\text{inside(c)}} |I(x,y) - c_1|^2 dxdy \\ + (1-\alpha)\lambda \iint_{\text{outside(c)}} |I(x,y) - c_2|^2 dxdy \end{aligned} \tag{3.3}$$

where c_1 and c_2 are the average intensities inside and outside the evolving curve C, i.e., $c_1 = \frac{\int_{\text{inside(C)}} I(x,y)dxdy}{\int_{\text{inside(C)}} dxdy}$ and $c_2 = \frac{\int_{\text{outside(C)}} I(x,y)dxdy}{\int_{\text{outside(C)}} dxdy}$.

The minimization of (3.3) leads to the following active contour model

$$\psi_t = \alpha\phi_l(\upsilon + \epsilon\kappa)|\nabla\psi| + \nabla\phi \cdot \nabla\psi + (1-\alpha)\lambda\delta_\beta(\psi)[(I-c_2)^2 - (I-c_1)^2] \tag{3.4}$$

In both equations, the weight function ϕ_l is selected as $1/(1+|\nabla I|^2)$. For more details about the derivation, refer to [78] [81] [135].

The first term in the right hand side of Eq. 3.4 is a combination of constant motion and curvature motion. In this term, υ is the inflationary term, which attracts the curve in one direction, either expanding or shrinking; κ represents the curvature, which keeps the curve smooth when evolving. The third term makes use of the region information, which helps to stop the curve at the boundaries of interest. In this term, $\delta_\beta(x) = \beta/(\pi(x^2+\beta^2))$ acts like an approximate delta function, as in [27]. α, β and ϵ are positive constant coefficients. λ is a normalization constant to make the values of gradient and region information comparable. α lies between 0 and 1 and determines the weights of the gradient and the region information. α is usually set to 0.5. For those images where weak edges exist, α is set to be less than 0.5 to increase the weight of the region information; 0.2 is usually a good choice for α in such cases.

When α is set to 1, only gradient information is used, and the model is the same as model (2.36) in [78]. When the function ϕ is chosen to 1 everywhere and υ is set to 0, (3.4) becomes the model (2.3) in [27]. Another point worth mentioning is that the minimization of the energy functional (3.3), according to the proofs in [81], will lead to the following active contour model

$$\psi_t = \alpha\phi_l(\upsilon + \epsilon\kappa)|\nabla\psi| + \nabla\phi \cdot \nabla\psi + (1-\alpha)\lambda\delta_\beta(\psi)[(I-c_2)^2 - (I-c_1)^2]|\nabla\psi| \tag{3.5}$$

Note that the region information, in (3.5), influences the speed of the evolving curve in its normal direction. Our model in (3.4) constructs a region force field to guide the curve evolution. The region force field bears some similarity to the region force diffusion model in [159], but differs in that our region force field varies with the evolving curve.

Utilization of region information as shown above is particularly suitable for a bimodal image. To deal with images containing multiple objects, a hierarchical approach is proposed,

which is similar to the one in [146]. In this approach, an image is segmented into two sub-images, and then the method is recursively applied on these sub-images. The recursion continues until each region contains only one object. In this way, the number of regions in the image need not to be known *a priori* [160] nor estimated in the segmentation [115]. We only need to detect whether each of the sub-images contains more than one object, which may be accomplished using the histogram of the sub-image.

The proposed method is easy to extend to vector-valued images. For a vector-valued image $I : \mathcal{R}^2 \to \mathcal{R}^m$, the length weight function is selected as:

$$\phi_l = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{1 + |\nabla G_\sigma * I_i|^n} \tag{3.6}$$

Accordingly, the region weight function is selected so that the level set function takes the following form:

$$\begin{aligned} \psi_t &= \alpha \phi_l (\upsilon + \epsilon\kappa)|\nabla\psi| + \nabla\phi \cdot \nabla\psi \\ &\quad + (1-\alpha)\delta_\beta(\psi)\frac{1}{m}\left[\sum_{i=1}^{m}(I_i - c_{2i})^2 - (I_i - c_{1i})^2\right] \end{aligned} \tag{3.7}$$

For color images, three channels of information can be directly used in the above formulas. For textured images, preprocessing techniques, such as wavelet or Gabor transforms, can be performed, and the resulting vector-valued images are used.

3.2.2 Connections between Curve Evolution and Anisotropic Diffusion

Connections between curve evolution methods and anisotropic diffusion methods are examined before introducing anisotropic diffusion to the proposed model.

For the curve evolution model in (2.38), the term $\mathrm{div}(\frac{\nabla\psi}{|\nabla\psi|})$ corresponds to the curvature of the evolving curve. It is used to keep the evolving curve smooth. The constant ϵ is usually chosen to be much smaller than υ. This term can then be neglected in the following analysis, without affecting the conclusion. Then (2.38) becomes

$$\psi_t = \upsilon\phi(x,y)|\nabla\psi| \tag{3.8}$$

Since $\phi(x,y)$ is related to the image only and independent of the evolving curve, the above equation is equivalent to the minimization of the following functional via Green's theorem [135] (set $\upsilon = 1$ here):

$$E(I) = \iint_\Omega \phi(x,y)dxdy = \iint_\Omega \frac{1}{1 + |\nabla G_\sigma * I|}dxdy \tag{3.9}$$

Comparing the energy functional of curve evolution (3.9), with that of anisotropic diffusion (2.22), (2.24) and (2.26), we find that they have structural similarities.

This similarity shows the relationship between curve evolution and anisotropic diffusion, which gives a clue to their combination. Curve evolution methods (2.3), (2.41), (3.3) minimize the energy functional by evolving a simple closed curve, which corresponds to the zero level set of ψ. The proposed method makes use of global information and smoothing to avoid stair-casing. Anisotropic diffusion methods (2.19), (2.23), on the other hand, minimize the energy functional by smoothing the whole image, which can be viewed as a set of curves

corresponding to multiple level sets. The proposed method utilizes the local information and reduce the effects of noise. Thus, the combination of these methods should solve the above problems and provide good performance. The method of coupled PDE's [119] is a good candidate for the combination.

3.2.3 Curve Evolution Model integrating Anisotropic Diffusion

Anisotropic diffusion methods are utilized to reduce the effects of noise on (3.4). Since both gradient and region information are used in the model, anisotropic diffusion methods are separately applied for each of them.

For gradient information, the Perona-Malik method (2.19) is applied, in which $g(|\nabla I|) = \frac{1}{1+(|\nabla I|/K)^2}$. Backward diffusion in the method (2.19) strengthens the edge gradient, which helps to stop the curve at object boundaries. Region information in the curve evolution model, on the other hand, helps to reduce stair-casing. Since curve evolution is used to minimize the energy functional, only closed contours exist in the final result for closed initial contour, and stair-casing problems are avoided. During curve evolution, only the gradient information near the evolving curve is used. Perona-Malik anisotropic diffusion, therefore, is applied to a narrow band close to the evolving curve.

For region information, the method (2.23) in [4] is applied to the regions away from the evolving curve. Since this method is well-posed and no backward diffusion is introduced, noise in the image will be effectively smoothed, and region information will be fully utilized.

The proposed model is described by the coupled PDE's:

$$I_t = f(\psi)\mathrm{div}\{\mathrm{g}(|\nabla \mathrm{I}|)\nabla \mathrm{I}\} + (1 - \mathrm{f}(\psi))\mathrm{g}(|\nabla \mathrm{I}|)|\nabla \mathrm{I}|\mathrm{div}\frac{\nabla \mathrm{I}}{|\nabla \mathrm{I}|} \tag{3.10}$$

$$\psi_t = \alpha\phi_l(\upsilon + \epsilon\kappa)|\bigtriangledown \psi| + (1-\alpha)\delta_\beta(\psi)[(I - c_2)^2 - (I - c_1)^2] \tag{3.11}$$

where $f(\psi)$ acts as an indicator function and takes the following form:

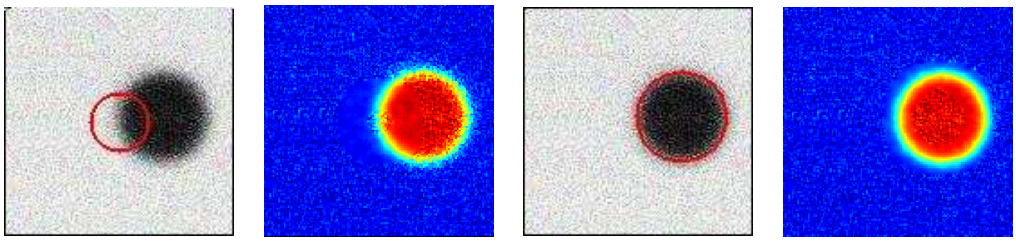
$$f(\psi) = \begin{cases} 1 & \text{if the pixel is close to } \psi \\ 0 & \text{if the pixel is far away from } \psi \end{cases}$$

In the proposed model, curve evolution methods help anisotropic diffusion methods by reducing stair-casing, while anisotropic diffusion methods help the curve evolution methods by reducing noise without smoothing object boundaries. This concept is implemented by introducing the coupling between the two PDE's (3.10) and (3.11).

3.2.4 Implementation Issues

The proposed model is implemented using level set methods [130] [105]. The narrow-band methods in [130] are utilized to reduce the computational load, and the fast marching methods are applied for re-initialization. The narrow-band is defined to contain those pixels which are no more than 6 pixels away from the evolving curve. In the implementation, the speed of the evolving curve υ is set to make the curve shrink at speed 1. The coefficient of the curvature ϵ is set to be 0.5.

For anisotropic diffusion, the Perona-Malik method (2.19) is applied to the pixels in the narrow-band, and the selective diffusion method (2.23) is applied elsewhere in the



(a) (b) (c) (d)

Figure 3.5: Region information in the model drives the curve, which is initially set to shrink, to evolve to correct position ($\alpha = 0.2$, 44 iterations, CPU = 6.93s). (a) Image with initial contour, size = 128 * 128. (b) The region field at the start. (c) Final segmentation result. (d) The region field in the end.

image. Each of the diffusion methods is applied for 20 iterations during initialization and re-initialization. The threshold K in the Perona-Malik method is set to be 40 unless specifically mentioned.

Another implementation issue is the choice for the value of the normalization parameter λ in (3.11) for specific images. To reduce the labor necessary to empirically choose the value, a heuristic method is utilized in the implementation. Since the maximal value of the gradient function ϕ is 1.0, λ is chosen as the maximal absolute value in the region force field. In this way, the largest absolute value of the region term is also 1.0, making it comparable to the gradient information. Experimental results shows that this implementation achieves very good experimental results.

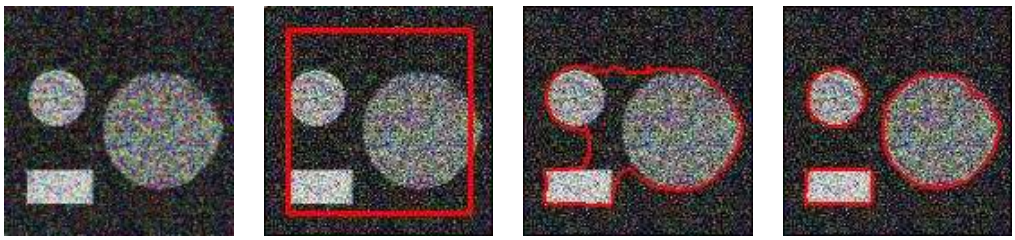
In this approach, the parameter λ is no longer a constant and varies with the evolving curve in each iteration. Thus, the implementation changes to some extent the energy functional to be minimized (3.3). Convergence problems may be introduced. Adaptive control techniques [128] may be utilized here, which is a topic for future research.

3.2.5 Experimental Results

Experimental results from the proposed model are given in this section. The program is implemented on a computer which has two Intel(R) Xeon^{TM} 2.4GHz CPUs , 2G bytes RAM, and runs the Red Hat Linux operating system. The CPU times given in this section are the sums of system CPU times and user CPU times. The system CPU time is usually very small, typically 0.01 - 0.03 second.

Fig. 3.5 illustrates the role of region information in the proposed model. The initial curve in Fig. 3.5(a) is set as a small circle in its center. The curve is set to shrink, but the region information forces the curve to move and expand to the correct boundary of the object, as can be seen in Fig. 3.5(c). The final result is very good, even though the image is very noisy. The region information also helps to stop the evolving curve at the correct position and avoid the boundary leakage problem. Fig. 3.5(b) and Fig. 3.5(d) show the region force field at the start and end of the curve evolution. Note that no big differences exist between Fig. 3.5(b) and Fig. 3.5(d).

Fig. 3.6 shows the segmentation of multiple objects in an image. Note that the original image Fig. 3.6(a) is much noisier than Fig. 3.5(a). Fig. 3.6(b), Fig. 3.6(c) and Fig. 3.6(d) show the evolving curve for the proposed model. The irregularity in the boundary of the big circle occurs in Fig. 3.6(d) because the corresponding area is more similar to the



(a) (b) (c) (d)

Figure 3.6: Segmentation of multiple objects in a noisy image. Multiple objects can be successfully segmented using the proposed model even in a very noisy image (size = 150 * 150) (α = 0.1, 348 iterations, CPU = 20.06s).. (a) Original image, size = 128 * 128. (b) Initial Contour. (c) Intermediate results during curve evolution. (d) Final results of curve evolution.

background than the circle. The objects are successfully segmented in spite of the noise. This experiment shows that anisotropic diffusion methods used by the proposed model work very well to reduce the effects of noise.

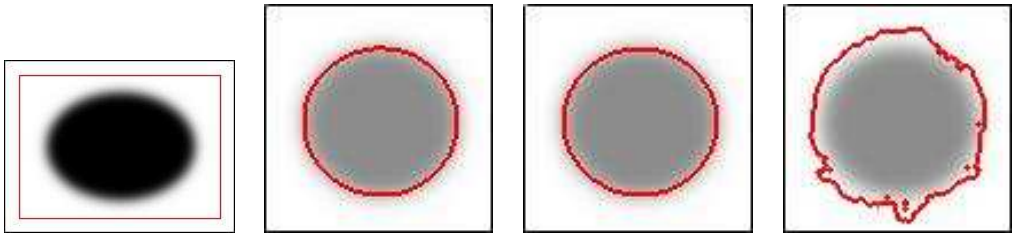
In Fig. 3.7, comparisons between the proposed method and the methods of [78] and [27] are given when α is set to be 0.1. The original image Fig. 3.7(a) is generated by smoothing an ideal circle using a 13-by-13 Gaussian filter. It can be seen that our method (Fig. 3.7(b)) and the method in [27] (Fig. 3.7(c)) correctly segment out the object in the image. The results from Fig. 3.7(b) and Fig. 3.7(c) show no big differences. In comparison, the method in [78] (Fig. 3.7(d)) allows the boundary leakage. This example shows that our model performs very well for images with weak edges.

Fig. 3.8 shows the segmentation results of a real image with initial contour Fig. 3.8(a) and Fig. 3.8(e). Fig. 3.8(b) - Fig. 3.8(d) show the results of the proposed method, in which the rock is successfully segmented. Fig. 3.8(f) - Fig. 3.8(h) show those of the method in [27]. The evolving curve, driven mostly by the region information, doesn't stop at the boundaries of the rock. It can be clearly seen that in this case our results in Fig. 3.8(d) are much better than Fig. 3.8(h) produced by the method of [8]. Fig. 3.9 shows more experimental results on real images.

The results in Fig. 3.10 show the segmentation of a triple junction in an image, in which a hierarchical approach is utilized. For the original image Fig. 3.10(a), the first iteration of the model separates the red region from the green and blue regions. The second iteration divides the green region and the blue region. The triple junction is successfully segmented by these two iterations.

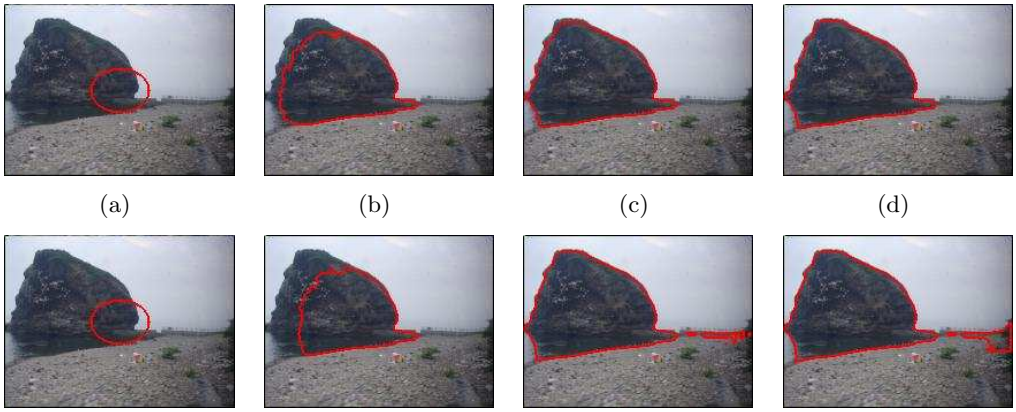
In Fig. 3.11, the susceptibility of the model to the stair-casing problem is tested. For the original image Fig. 3.11(a), which is the same as Fig. 3.5(a), Perona-Malik diffusion is performed with the threshold $K = 10$. Stair-casing occurs in Fig. 3.11(b) as expected. Fig. 3.11(c) shows the magnification of a portion in Fig. 3.11(b), which shows stair-casing. Fig. 3.11(d) - Fig. 3.11(g) show the results of the proposed model. The final segmentation result in Fig. 3.11(g) is good. Fig. 3.11(h) also magnifies part of Fig. 3.11(g). It can be seen that stair-casing is not present, and the image is segmented into two regions.

In Fig. 3.12, the proposed method is applied to an image with heavy noise, as can be seen in the original image (a). (b) and (c) show intermediate segmentation results. (d) contains the final segmentation results. From these results, we can see that our model achieves acceptable results in the presence of high noise.



(a) (b) (c) (d)

Figure 3.7: The boundary leakage problem. Segmentation of multiple objects in a noisy image. Multiple objects can be successfully segmented using the proposed model even in a very noisy image (size = 150 * 150) (α = 0.1, 348 iterations, CPU = 20.06s).. (a) Image containing weak edges with initial contour, size = 100 * 100. (b) Results using the proposed method (α = 0.1, 451 iterations, CPU = 5.14s). (c) Results using the method in [27] (α = 0.1, 51 iterations, CPU = 0.84s). (d) Results using the method in [78], where boundary leakage occurs (α = 1.0, 1137 iterations, 13.32s).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.8: Segmentation of real images. (a) original images with initial contour, size = 200 * 150. (b) Intermediate results using the proposed method. (c) Intermediate results using the proposed method. (d) Segmentation results using the proposed method (α = 0.2, 334 iterations, CPU = 57.99s). (e) original images with initial contour, size = 200 * 150. (f) Intermediate results using the method in [27]. (g) Segmentation results using the method in [27]. (h) Final results using the method in [27](α = 0.1, 934 iterations, CPU = 84.13s).

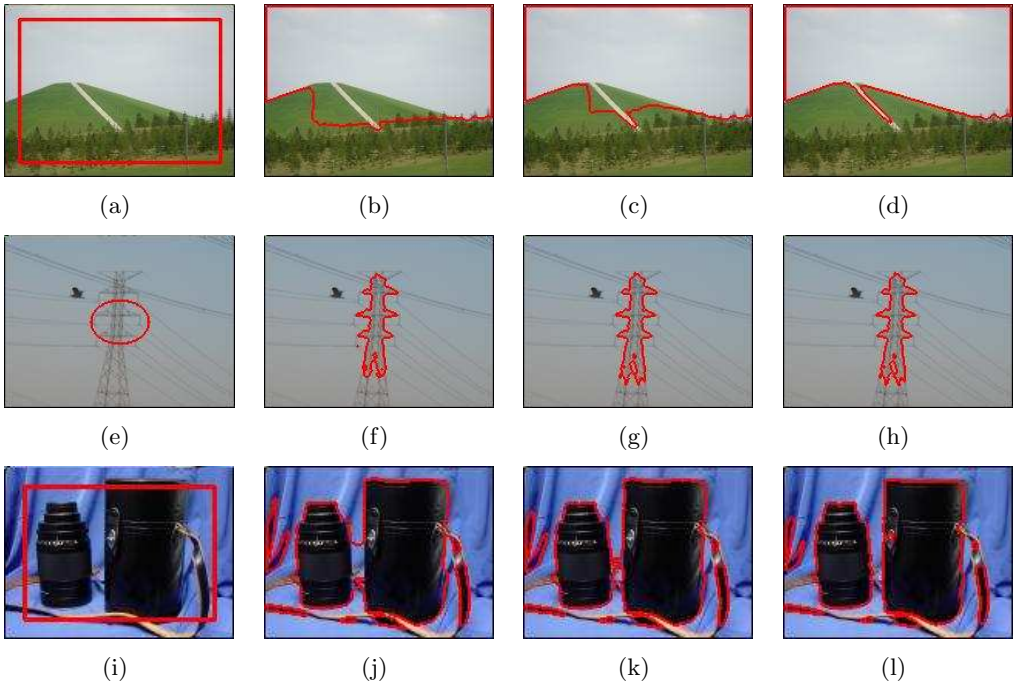


Figure 3.9: Segmentation of real images. (a) Original image with initial contour, size = 200 * 150. (b) Intermediate results of (a). (c) Intermediate results of (a). (d) Segmentation results of (a) (α = 0.2, 680 iterations, CPU = 52.53s). (e) Original image with initial contour, size = 200 * 150. (f) Intermediate results of (e). (g) Segmentation results of (e). (h) Final results of (e) (α = 0.1, 115 iterations, CPU = 20.56s). (i) Original images with initial contour, size = 200 * 150. (j) Intermediate results of (i). (k) Intermediate results of (i). (l) Segmentation results of (i) (α = 0.2, 451 iterations, CPU = 25.87s).

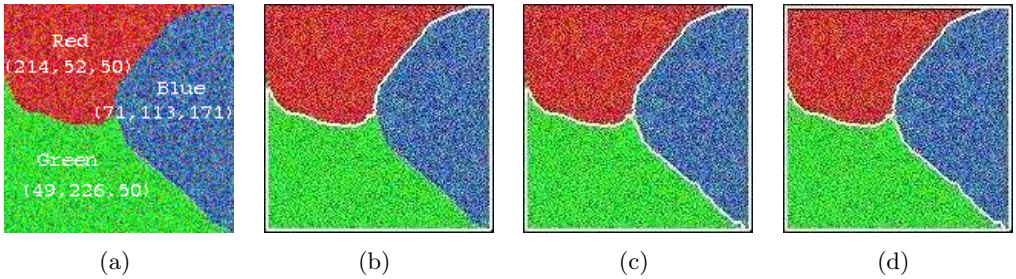


Figure 3.10: Segmentation of triple junctions in an image using an hierarchical approach. (a) Original image, size = 150 * 150. (b) Segmentation results after first iteration. (c) Segmentation results after second iteration. (d) Final segmentation results.

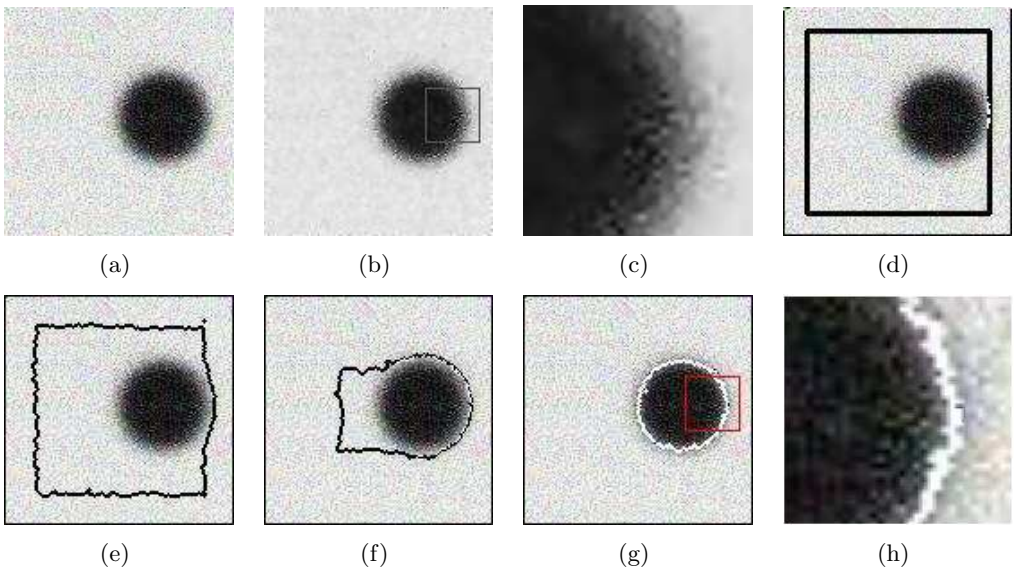


Figure 3.11: Solution of the stair-casing problem. (a) Original image, size = 182 * 182. (b) Stair-casing problem by PM method ($K = 10$). (c) Magnified portion in (b). (d) Initial curve using the proposed method. (e) Intermediate results using the proposed method. (f) Intermediate results using the proposed method. (g) Segmentation results using the proposed method. (h) Magnified portion of (g).

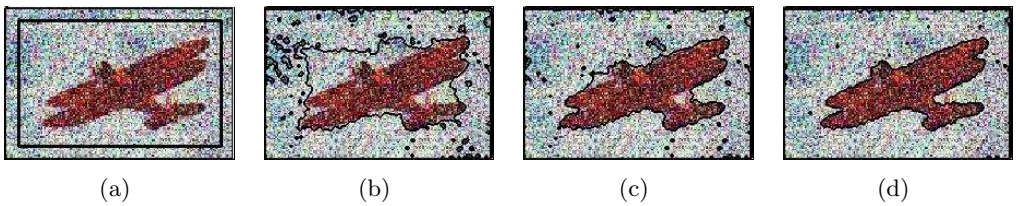


Figure 3.12: Segmentation of an image with heavy noise(size = 211 * 141, $\alpha = 0.2$, 289 iterations, CPU = 193.88s). (a) Original image. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.

3.2.6 Summary

A new image segmentation method is proposed in this section, which integrates curve evolution and anisotropic diffusion. The curve evolution method, utilizing both the region and gradient information, solves the boundary leakage problem. It can segment complicated images with the hierarchical approach. Anisotropic diffusion methods reduce the effects of noise on both the gradient and the region information. Experimental results show that the model works very well.

3.3 Bottom-Up Hierarchical Image Segmentation using Region Growing and the Mumford-Shah Model

The contributions of this section (references [111]- [113]) may be divided into three categories. First, a mathematical analysis of the initialization problem [55] in the Chan-Vese model [27] is provided. It shows that the initialization problem relies upon its top-down hierarchy to utilize the global region information in the image. Second, an efficient curve evolution method is proposed for the Chan-Vese model. Although it still has initialization problem, this method works very fast for images without strong noise. Finally, and most importantly, an efficient image segmentation method is proposed, based on region growing, region competition, and the Mumford Shah functional [100]. This algorithm is able to automatically and efficiently segment objects in complicated images. By means of bottom-up techniques, the method solves the initialization problem in the Chan-Vese model and works very well for images with multiple junctions. It also works very well for color images and can be easily extended to textured images. Experimental results show that the proposed method is robust in the presence of strong noise.

3.3.1 The Initialization Problem of The Chan-Vese Model

In this section, a mathematical analysis is provided for the initialization problem of the Chan-Vese model. This analysis acts as the foundation for the methods proposed in Section 3.3.3 and Section 3.3.4. The Chan-Vese model is introduced first, followed by the analysis for its initialization problem.

The Chan-Vese Model

The Chan-Vese model [27] [150] is the curve evolution implementation of a special case of the Mumford-Shah model [100]. The bimodal Chan-Vese model [27] tries to minimize the following energy functional:

$$\begin{aligned} F(c_1, c_2, C) = \mu \cdot \text{Length(C)} \\ + \lambda_1 \iint_{\text{inside(C)}} |I(x,y) - c_1|^2 dxdy \\ + \lambda_2 \iint_{\text{outside(C)}} |I(x,y) - c_2|^2 dxdy \end{aligned} \tag{3.12}$$

where I is the original image, C is the evolving curve, and c_1 and c_2 are selected as the average values of pixels inside and outside C, respectively. μ, λ_1 and λ_2 are positive constants.

Both λ_1 and λ_2 are usually taken as 1. These two parameters, therefore, are neglected in the following derivations.

The energy functional (3.12) is minimized by solving the following PDE:

$$\psi_t = \delta_\epsilon(\psi)[\mu \cdot \kappa - (I - c_1)^2 + (I - c_2)^2] \tag{3.13}$$

where ψ is the level set representation of the evolving curve C at time t, which means that $C = \{(x, y)|\psi(x, y, t) = 0\}$. κ represents the curvature of the evolving curve. $\delta_\epsilon(\psi) = \epsilon/(\pi(\epsilon^2 + \psi^2))$ and ϵ is a positive constant.

It can be clearly seen from (3.13) that the evolution of the curve is affected by two terms. The curvature term κ regularizes the curve and makes it smooth during evolution. The region term $-(I - c_1)^2 + (I - c_2)^2$ affects the motion of the curve. In fact, the initialization of the curve affects curve evolution through this term.

The Initialization Problem of The Bimodal Chan-Vese Model

Consider a piecewise constant bimodal image. Suppose there are n_1 pixels in the background of the image, among which $m_1(0 \leq m_1 \leq n_1)$ pixels lie inside the initial curve. Suppose there are n_2 pixels in the foreground of the image, among which $m_2(0 \leq m_2 \leq n_2)$ lie inside the initial curve. All the pixels in the background (foreground) take $u_1(u_2)$ as their intensity values. Obviously, $m_1 + m_2 > 0$ for all initializations.

Then the average intensity inside the evolving curve is

$$c_1 = (m_1 u_1 + m_2 u_2)/(m_1 + m_2) \tag{3.14}$$

and the average intensity outside the evolving curve is

$$c_2 = ((n_1 - m_1)u_1 + (n_2 - m_2)u_2)/((n_1 - m_1) + (n_2 - m_2)) \tag{3.15}$$

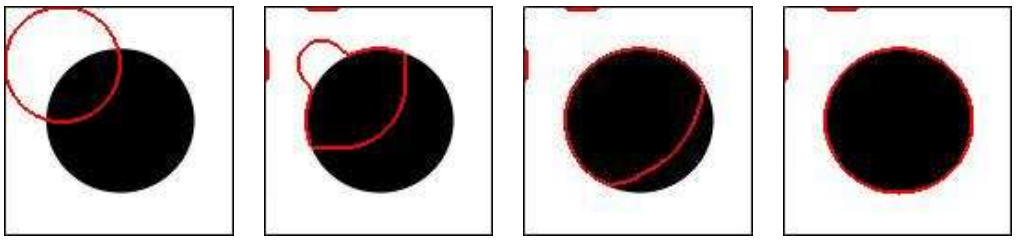
Therefore, the region terms $-(I - c_1)^2 + (I - c_2)^2$ for points on the evolving curve in the foreground and the background are,

$$(u_2 - c_2)^2 - (u_2 - c_1)^2 = K_0 K_2 (m_2 n_1 - m_1 n_2)(u_1 - u_2)^2 \tag{3.16}$$

$$(u_1 - c_2)^2 - (u_1 - c_1)^2 = -K_0 K_1 (m_2 n_1 - m_1 n_2)(u_1 - u_2)^2 \tag{3.17}$$

where $K_0 = 1/\{(n_1 - m_1 + n_2 - m_2)(m_1 + m_2)\}$, $K_1 = (n_2 - m_2)/(n_1 - m_1 + n_2 - m_2) + m_2/(m_1 + m_2)$ and $K_2 = (n_1 - m_1)/(n_1 - m_1 + n_2 - m_2) + m_1/(m_1 + m_2)$, respectively. K_0, K_1 and K_2 are positive for any initialization.

From (3.16) and (3.17), it can be seen that the region term for points on the foreground part of the evolving curve is opposite in sign to that on the background part of the evolving curve. Therefore, if one part of the evolving curve expands, the remaining part will have to shrink and vice versa. Without loss of generality, set $\psi > 0$ inside the evolving curve and $\psi < 0$ outside the evolving curve. Then three cases may happen for different initializations. First, if $m_2 n_1 - m_1 n_2 > 0$, i.e., $m_2/n_2 > m_1/n_1$, then the foreground part of the curve expands, and the background part shrinks. The curve evolves into the foreground and segments the object from the foreground, as shown in Fig. 3.13. Secondly, if $m_2 n_1 - m_1 n_2 < 0$, i.e., $m_2/n_2 < m_1/n_1$, then the foreground part of the curve shrinks and the background part expands. The curve evolves into the background and segment the object from the background, as shown in Fig. 3.14,. Finally, if $m_2 n_1 - m_1 n_2 \approx 0$, then the influence of the



(a) (b) (c) (d)

Figure 3.13: Example of the initialization problem, where $m_2/n_2 = 0.21$ is larger than $m_1/n_1 = 0.16$. (a) A bimodal image with initial curve. (b) Intermediate results of curve evolution after 10 iterations. (c) Intermediate results of curve evolution after 60 iterations. (d) Final segmentation results after 234 iterations.

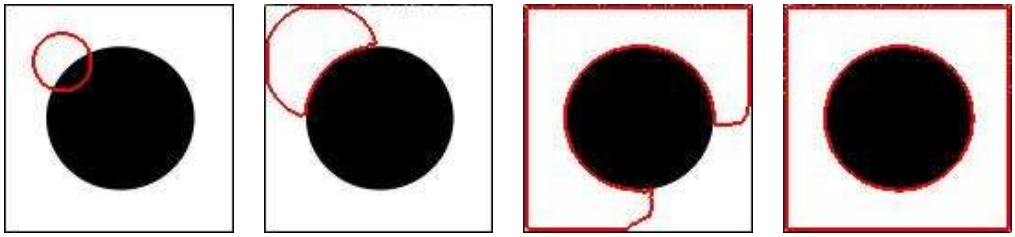
region term on the curve evolution is small at first. The curve is expected to evolve very slowly and may segment nothing.

It can be seen from the above analysis that initialization greatly affects curve evolution in the Chan-Vese model. The results from Fig. 3.13 and Fig. 3.14 also illustrate this point explicitly. Although curve evolution in Fig. 3.13 takes less time to segment the object than in Fig. 3.14, the same segmentation results are achieved. In complicated cases, however, different initializations may generate different segmentation results, as shown in Fig. 3.15. Both initializations in Fig. 3.15(a) and Fig. 3.15(c) satisfy $m_2/n_2 > m_1/n_1$, making the foreground part of the evolving curve expand and the background part shrink. Since the upper object is not included in the initialization in Fig. 3.15(c), it is not segmented for that initialization. This suggests that every object in the image should have at least one pixel included in the initial curve for good segmentation. For the Chan-Vese model, therefore, good choices for the initialization would be the boundary or multiple bubbles. Initialization can be a even bigger problem for multi-modal images for the Chan-Vese model. This holds even for the multi-phase Chan-Vese model [150], as shown in [55]. In the worst case, the Chan-Vese model can fail to segment any object if the total influence of the region information on the initial curve is zero, as is shown in Fig. 3.19 (b1)(b2).

The Chan-Vese model uses a top-down hierarchical method for segmentation, and that global region information is utilized only on the evolving curve in the model. The initialization problem is a consequence of this. A bottom-up hierarchical method, which makes use of local information, may solve the initialization problem and deal with complicated images. A new image segmentation method, which is based on region growing and the Mumford-Shah model, is proposed later in this section. Before the introduction of the new method, however, we propose a very fast curve evolution method for the Chan-Vese model, based on the analysis in this section.

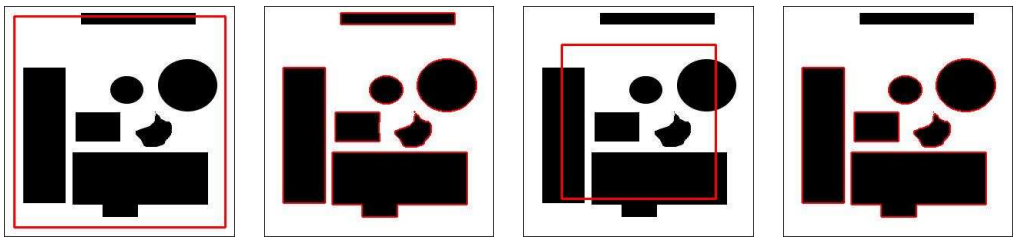
3.3.2 Analysis of the Initialization Problem of the Chan-Vese Models

In Subsection 3.3.1, the initialization problem is analyzed for the bimodal Chan-Vese model. In this subsection, a more comprehensive analysis is provided for the initialization problem of the Chan-Vese models. Based on the background information in Subsection 3.3.1, three observations can be made about the initialization problem.



(a) (b) (c) (d)

Figure 3.14: Example of the initialization problem, where $m_2/n_2 = 0.036$ is smaller than $m_1/n_1 = 0.065$. (a) A bimodal image with initial curve. (b) Intermediate results of curve evolution after 10 iterations. (c) Intermediate results of curve evolution after 600 iterations. (d) Final segmentation results after 1182 iterations.



(a) (b) (c) (d)

Figure 3.15: Different initialization in the Chan-Vese model may generate different segmentation results. (a) A multi-modal image with initial curve. (b) Segmentation results of (a) after 2600 iterations. (c) A multi-modal image with another initial curve. (d) Segmentation results of (c) after 913 iterations.

First, initialization determines which local minimum of the energy functional (2.2) is achieved. Initialization in the Chan-Vese models provides the starting point for the minimization of the energy functional. Since the energy functional may have multiple local minima, and the Euler-Lagrange method is a gradient-descent method, a local minimum may be reached, for example, if the initial value is chosen to be closer to one of the local minima than the global minimum. Fig. 3.15 illustrates this fact. The initialization in Fig. 3.15(a) causes the global minimum of (2.2) to be found where all the objects are segmented. The initialization in Fig. 3.15(c), however, is closer to a local minimum of (2.2) than to the global minimum, and one object is not segmented as a result in Fig. 3.15(d) . The authors of [112] have also illustrated that the solutions to the Chan-Vese model may go through different intermediate states for different initializations even if they achieve the same local minimum.

Second, the way the Chan-Vese model utilizes region information creates the initialization problem. In the Chan-Vese model, information from different regions are competing to evolve the curve. If initialization causes the total influence of multiple regions on the curve to be zero, nothing is segmented. Although the multi-phase Chan-Vese model introduces computation between multiple small regions, initialization is still a problem without prior information about the image.

Third, coupling between evolving curves may magnify the effects of initialization and introduce more computational load. Fig. 3.16 illustrates this fact. For the initialization in Fig. 3.16(b), curve evolution based on only region information can not reach a local minimum, as shown in Fig. 3.16(d). It can be seen from Fig. 3.16(e) - Fig. 3.16(h), however, that every part of the evolving curve converges to a local minimum. Although the curvature term may finally drive the curves to a local minimum, the coupling introduces extra computation and makes the segmentation time-consuming. Therefore, it is useful to decouple the evolving curves in the multi-phase Chan-Vese model for better segmentation results [55].

3.3.3 Fast Curve Evolution Method Without Solving PDEs

The mathematical analysis of the Chan-Vese model provides the concept for a fast curve evolution method that does not require solving PDEs. The curve evolution method is based on the way the Chan-Vese model utilizes region information. Observing Eq. 3.16 and Eq. 3.17, we can see the following: First, the region term $-(I - c_1)^2 + (I - c_2)^2$ of the foreground has the opposite sign to the region term of the background for any initialization. Second, for any point strictly inside the foreground or the background, its region term will has the same sign as the terms for its neighboring points. Third, only the boundary points will have neighboring points with region terms different in sign.

These observations hold for all bimodal images without strong noise. This information can be used to construct an efficient evolution of the initial curve. A list of points, instead of a narrow band of points, is utilized to represent the evolving curve. Without loss of generality, suppose the points inside the initial curve are set to have positive ψ value. For any point in the list, if this point and all its neighboring points have positive region terms, then at that point the curve will expand according to Eq. 3.13. We only need to remove this point from the list, and add to the list those neighboring points that have negative ψ values. Correspondingly, if a point on the list and all its neighboring points have negative terms, then the curve will shrink at that point. We need to remove this point from the list

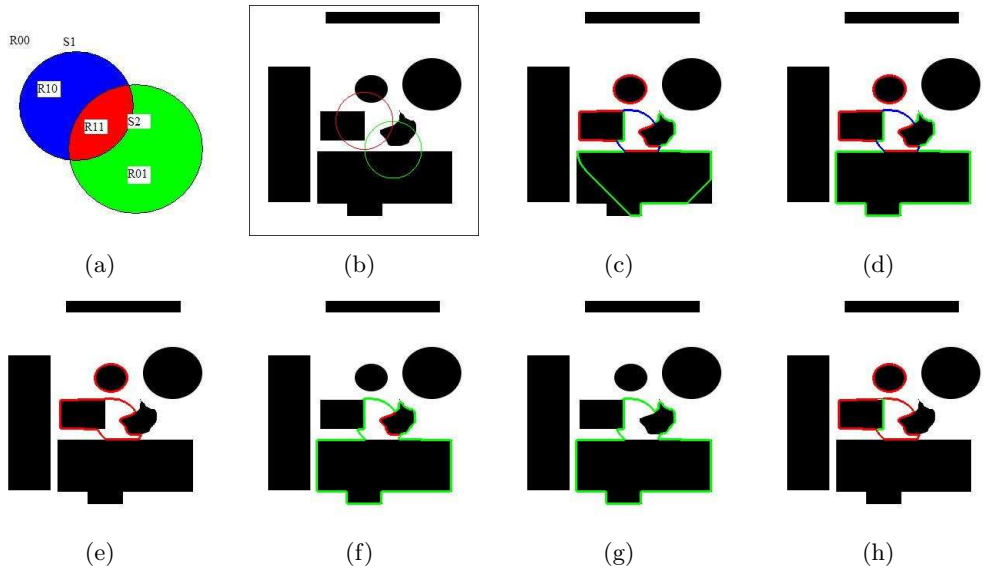
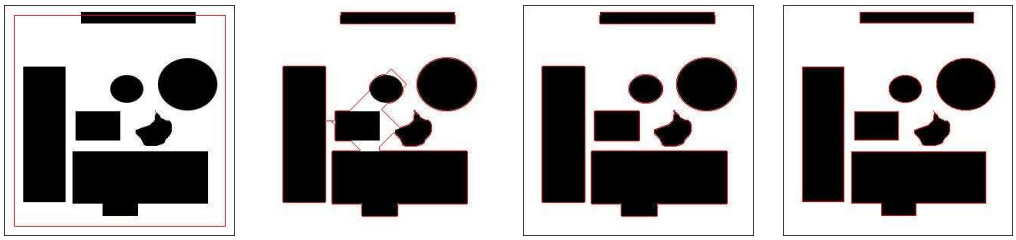


Figure 3.16: Coupling between curve evolution may enlarge initialization problems. (a) Multi-phase Chan-Vese model. (b) Initialization. (c) Intermediate results after 12 iterations. (d) Segmentation results. (e) The positions of one part of the evolving curves. (f) The positions of one part of the evolving curves. (g) The positions of one part of the evolving curves. (h) The positions of one part of the evolving curves.



(a) (b) (c) (d)

Figure 3.17: Comparison of the proposed fast curve evolution method to the classical method solving PDEs [105] [130]. The new method provides a 23 times speed-up. (a) An image (size - 300 * 300) with initial curve. (b) Intermediate result using the proposed method. (c) Segmentation result of the proposed method, CPU = 0.51s. (d) Segmentation result of the Chan-Vese model by solving PDEs, CPU = 11.98s.

and add to the list those neighboring points that have positive ψ values. Otherwise, the boundary has been reached, and nothing needs to be done.

The proposed method is very efficient compared to the classical method solving PDEs [105] [130], as can be seen from Fig. 3.17. For the images of Fig. 3.17, the proposed method is 23 times faster and achieves the same segmentation results. Figs. 3.17(a) - Fig. 3.17(c) demonstrate that the proposed method is able to automatically handle topological changes.

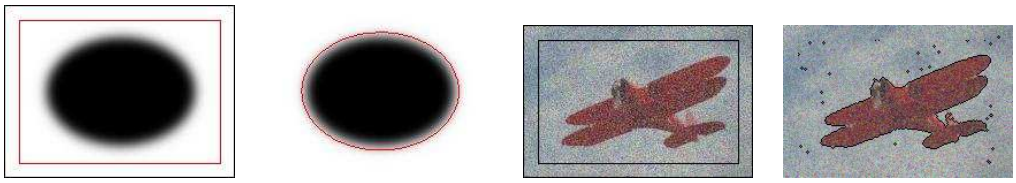
Fig. 3.18 shows segmentation results using the proposed method for more complicated images. Fig. 3.18(a) and Fig. 3.18(b) show that the method works well for images with weak edges. The results in Fig. 3.18(c) and Fig. 3.18(d) demonstrate the ability of the method to handle noise. One point worth mentioning is that region information has to be updated after every iteration to reduce the effects of noise for Fig. 3.18(c) and Fig. 3.18(d); this is not required in the image of Fig. 3.17.

Up to this point, the regularization term in Eq. 3.13 has not been used in the proposed method. As proved in [54] [58] [80], curve evolution based on curvature $\partial C/\partial t = \kappa N$ is equivalent to a nonlinear analog to Gaussian smoothing. Thus Gaussian smoothing can be applied after each iteration of curve evolution to smooth the curve. Experimental results will be provided to illustrate this idea in the next section.

Although the proposed method provides high efficiency for curve evolution, problems remain. First, this method makes use of only the sign of the region information, so it is sensitive to strong noise. If noise is strong enough to change the signs of the region terms, the proposed algorithm will fail. Second, it is still sensitive to initialization. Third, it is difficult to deal with complicated images, such as triple junctions. A new segmentation algorithm will be provided in the next section to solve these problems.

3.3.4 Image Segmentation Using Region Growing and the Mumford-Shah Functional

In this section, a new image segmentation method will be proposed based on region growing, region competition and the Mumford-Shah model. First, an explanation of the Chan-Vese models [27] [150] is provided based on the concept of region competition. Second, the proposed segmentation method will be shown in details. Third, the proposed method will be extended to color images.



(a) (b) (c) (d)

Figure 3.18: Fast curve evolution of the Chan-Vese model for complicated cases. (a) An image (size - 200 * 150) with weak edges and the initial curve. (b) Segmentation result of (a), CPU = 0.28s. (c) A noisy image (size - 200 * 133) with the initial curve. (d) Segmentation result of (c), CPU = 0.3s.

Region Competition in the Chan-Vese Models

The Chan-Vese models [27] [150] minimize an energy functional by evolving an initialized curve. Curve evolution, therefore, can be seen as the result of competition between the foreground and the background. This idea is similar to the method proposed in [163]. Curve evolution stops when the competition reaches a balance.

Consider the bimodal case in Section 3.3.1, and let $\psi > 0$ inside the evolving curve. The curve evolves according to the competition between the foreground region and the background region. When the evolving curve reaches the boundary of the object, $m_1 = 0$, $m_2 = n_2$, $c_1 = u_2$, and $c_2 = u_1$. The region terms calculated using the boundary points in the foreground are $(u_2 - c_2)^2 - (u_2 - c_1)^2 = (u_2 - u_1)^2 > 0$, and the region terms for the boundary points in the background are $(u_1 - c_2)^2 - (u_1 - c_1)^2 = -(u_2 - u_1)^2 < 0$. These region terms are equal in magnitude and opposite in sign. The competition is balanced, and the evolving curve stops at the boundary of the object.

Image Segmentation Based on Region Growing and The Mumford-Shah Functional

The proposed image segmentation method is designed to minimize the well-posed case of the Mumford-Shah functional [100] using bottom-up region growing and region competition. The energy functional takes the form:

$$E(\Gamma) = \sum_i \iint_{R_i} (I - c_i)^2 dxdy + \nu \cdot \Gamma \tag{3.18}$$

where Γ represents the length of object boundaries, I represents the image to be segmented, c_i represents the average intensity of the ith region R_i and ν is a constant parameter.

As mentioned in [100], the energy functional (3.18) tends to segment images into piecewise constant regions, which gives an opportunity for minimization using region growing. The proposed method works as follows: At the beginning of image segmentation, every pixel in the image is taken as one region. A region is merged with a neighboring region if this action will decrease the energy functional (3.18). In this way, the neighboring regions of the current region are competing with each other to reduce the energy functional. After two regions are merged, the intensity of each pixel in the merged region will be the average intensity of the region. The process is repeated until no region merging occurs. A mathematical description of region merging is given below.

Consider two neighboring regions Ω_1 and Ω_2 in the image. Suppose Ω_1 and Ω_2 contain n_1 and n_2 pixels respectively, with c_1 and c_2 as their average intensities. These two regions have Γ pixels in common. If these two regions are merged, the average intensity would be $c = (n_1c_1 + n_2c_2)/(n_1 + n_2)$. The energy functionals before and after region merging are

$$E_{prev} = \iint_{\Omega_1} (I - c_1)^2 dxdy + \iint_{\Omega_2} (I - c_2)^2 dxdy + \nu \cdot \Gamma \tag{3.19}$$

$$E_{after} = \iint_{\Omega_1} (I - c)^2 dxdy + \iint_{\Omega_2} (I - c)^2 dxdy \tag{3.20}$$

. The energy difference, therefore, is

$$\begin{aligned} E_{after} - E_{prev} = 2V_1(c_1 - c) + 2V_2(c_2 - c) + n_1(c^2 - c_1^2) \\ + n_2(c^2 - c_2^2) - \nu \cdot \Gamma \end{aligned} \tag{3.21}$$

where $V_1 = \iint_{\Omega_1} I dxdy$ and $V_2 = \iint_{\Omega_2} I dxdy$. Region merging is performed only if the energy difference is smaller than zero.

During region growing, irregular boundaries may be generated, especially in images with strong noise. Regularization for region boundaries, therefore, is necessary. Gaussian smoothing is utilized for regularization, as in the method of Section 3.3.3. Gaussian smoothing is performed only when the region becomes larger than a specified threshold.

In the proposed method, region information is used in a manner that is similar to the Chan-Vese model [27], but initialization problem can be avoided and complicated cases such as multiple regions and triple junctions can be automatically handled, because of the bottom-up hierarchical approach of the proposed method.

Extension to Color Images

The proposed method can be extended in a straightforward manner to color images. Several color models, such as the one used in [163], can be chosen for the extension. The RGB color model is utilized here for simplicity. The energy functional is taken as the summation of energy functionals for each of three channels of the image, and has the form

$$E(\Gamma) = \sum_{j=1}^{3} \sum_{i} \iint_{R_i} (I_j - c_{ij})^2 dxdy + \nu \cdot \Gamma \tag{3.22}$$

where j represents the index of the color channel.

The implementation of the proposed method for color images is similar to that for intensity images. The only difference lies in the calculation of the region information, which is straightforward.

3.3.5 Implementation Issues

The key issue for the implementation of the proposed method is how to select an appropriate value for ν for a specified image. Intuitively, ν should be large enough to suppress noise and small enough not to merge regions separated by edges with high gradients. We show that the value of ν should also be related to the size of competing regions.

Consider two extreme cases for the region growing problem section 3.3.4. First, if $V_1 = n_1c_1$, $V_2 = n_2c_2$ and $n_1 = n_2 = n$, then $c = (c_1 + c_2)/2$, and

$$E_{after} - E_{prev} = 1.5 \cdot n(c_1 - c_2)^2 - \nu \cdot \Gamma \tag{3.23}$$

If, on the other hand, $V_1 = n_1c_1$, $V_2 = n_2c_2$ and $n_1 >> n_2$, then $c \approx c_1$, and

$$E_{after} - E_{prev} \approx n_2(c_1 - c_2)^2 - \nu \cdot \Gamma \tag{3.24}$$

In both cases, the energy change in region growing is highly related to the size of competing regions (n in Eq. 3.23 or n_2 in Eq. 3.24). It is very difficult to select the proper ν value if many regions of different sizes are competing at the same time.

Since the proposed method assigns a region to each pixel at the beginning, the above problem can be solved by restricting the maximum size of a region after each iteration. For example, the largest region after the first traverse of the whole image is set to 2 pixels. The image is then segmented into numerous regions containing two pixels after the first traverse. The largest region after the second traverse of the image is then set to 4 pixels. Now almost all competing regions contain 2 pixels and the image will be segmented into regions of 4 pixels, and so on. In this way, most competing regions have the same size, and one appropriate value for ν may be enough for the segmentation of a specified image.

Suppose the variance of noise in an intensity image is σ^2, and the gradient of the region boundaries is expected to be g_0 ($g_0 > \sigma$); then the value of ν can be selected such that $\sigma^2 < \nu < g_0^2$. For color images, the value of ν can be selected such that $3\sigma^2 < \nu < 3g_0^2$. It is usually acceptable to choose $\nu = 1000$ for intensity images and $\nu = 3000$ for color images.

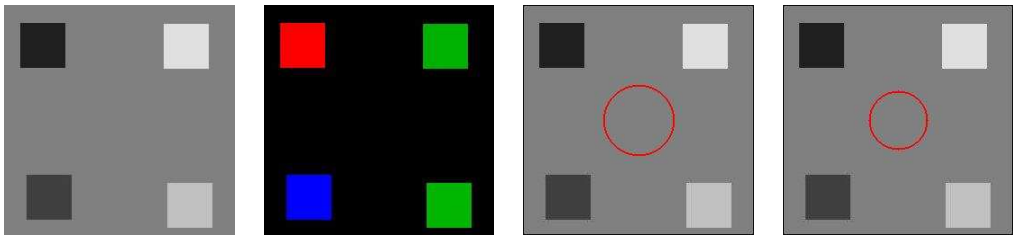
3.3.6 Experimental Results

Experimental results from the proposed method are shown in this section. The proposed method is implemented on a computer which has two Intel(R) Pentium(R) 3.2GHz CPUs, 2G bytes RAM, and runs the Red Hat Enterprise Linux operating system. The CPU times given in this section are the sums of system CPU times and user CPU times. The system CPU time is usually very small, typically 0.01 - 0.08 seconds.

Fig. 3.19 represents the comparison of the proposed method and the Chan-Vese model. Fig. 3.19(a) shows the image to be segmented. The image contains one background region (intensity 128) and 4 foreground regions (intensity 32, 64, 192, 224 counterclockwise) of equal size. The segmentation result in Fig. 3.19(b) shows that the proposed is very efficient (1.9s). The Chan-Vese model fails for the initialization shown in Fig. 3.19(c). Since the effects of the region information on the curve are zero, the curve evolves very slowly, driven by the curvature. After more than 13 seconds, the initial curve evolves into Fig. 3.19(d) and will shrink to a point in the end.

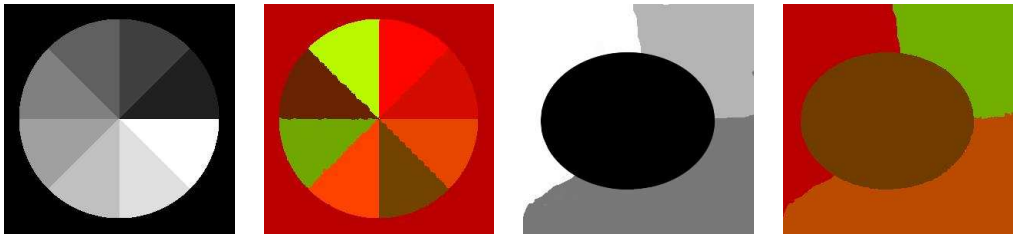
Fig. 3.20 demonstrates the ability of the proposed method to deal with images with multiple junctions. The initialization problem can happen for the Chan-Vese model, as shown in [55]. The proposed method generates very good segmentation results with high efficiency.

Fig. 3.21 shows the stability of the proposed method w.r.t. noise. It can be seen that the proposed method works very well for images with strong noise. The results also show that a larger ν is required, and the segmentation process becomes longer for images with stronger noise.



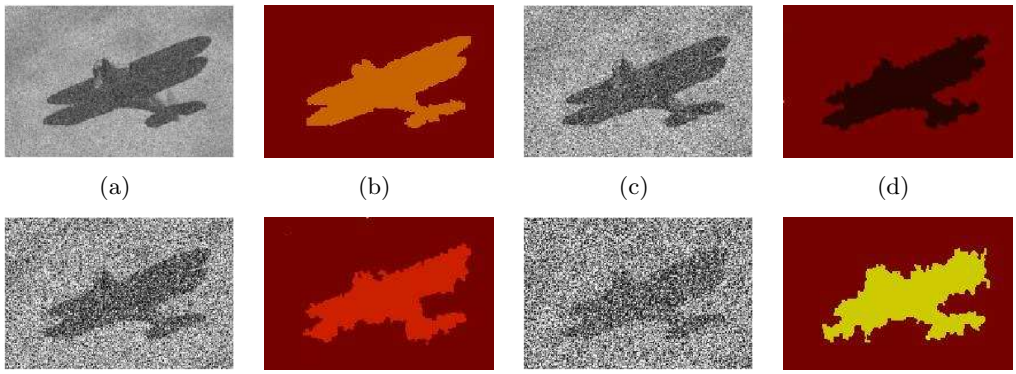
(a) (b) (c) (d)

Figure 3.19: Comparison of the proposed method and the Chan-Vese model for images with multiple regions. (a) A gray image (size - 256 * 256) with average intensity 128. (b) Segmentation result of the proposed method, ν=1000, CPU =1.9s. (c) A gray image with the initialized curve.. (d) Segmentation result using the Chan-Vese model, CPU = 13.03s.



(a) (b) (c) (d)

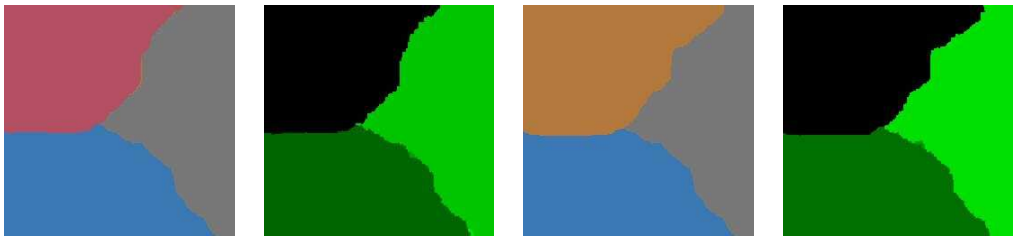
Figure 3.20: More results of the proposed method for complicated images. (a) A biscuit image (size - 300 * 300). (b) Segmentation result of (a), ν=300, CPU =4.08s.. (c) An image with multiple junctions (size - 300 * 300). (d) Segmentation result of (c), ν = 2000, CPU = 5.02s.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.21: Stability of the proposed method w.r.t. noise. (a) An image with very weak noise (size - 200 * 133). (b) Segmentation result of (a), ν=1000, CPU =1.08s. (c) An image with weak noise (size - 200 * 133). (d) Segmentation result of (c), ν = 2000, CPU = 5.52s. (e) An image with strong noise (size - 200 * 133). (f) Segmentation result of (e), ν = 3000, CPU = 8.85s. (g) An image with very strong noise (size - 200 * 133). (h) Segmentation result of (g), ν = 4000, CPU = 7.25s.



(a) (b) (c) (d)

Figure 3.22: Extension of the proposed method to color images. (a) A color image (size - 256 * 256). (b) Segmentation result of (a), ν=1050, CPU =2.19s. (c) A color image (size - 256 * 256). (d) Segmentation result of (c), $\nu = 1200$, CPU = 2.05s.

In Fig. 3.22, the extension of the proposed method to color images is tested. Images in Fig. 3.22(a) and Fig. 3.22(c) are designed to have the same intensity so that they can not be segmented just using intensity. By means of the color information, the proposed method successfully segments objects with different colors.

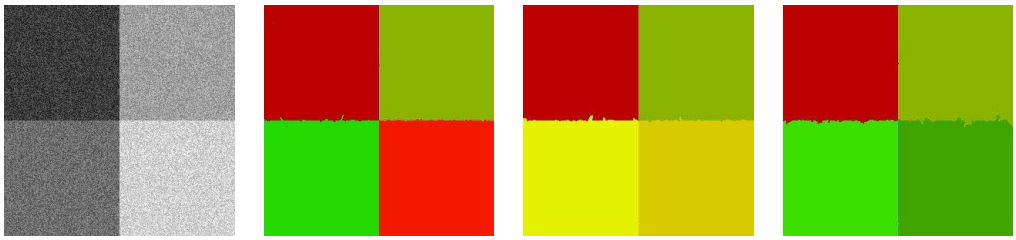
Fig. 3.23 shows the effects of ν on the segmentation results. The image has four regions. The pixels are randomly chosen and independent, with Gaussian distribution $N(60, 40^2)$, $N(110, 40^2)$, $N(160, 40^2)$, and $N(210, 40^2)$. By the discussion in Section 3.3.5, the choice of ν should satisfy $40^2 < \nu < 50^2$. The results in Fig. 3.23 show that the proposed method works for a wider range of ν. From the results in Fig. 3.23 (b)(c)(d), it can be seen that the segmentation time becomes shorter with increasing ν, while at the same time the object boundaries become coarser. This can be explained by the fact that regions are more likely to be merged with larger ν values. In practice, a compromise has to be made between efficiency and accuracy.

Fig. 3.24 illustrates the utility of the proposed method for images with weak edges. From the results in Fig. 3.24(b) - Fig. 3.24(d), we can see that the proposed method over-segments the image Fig. 3.24(a). This can be explained by the fact that gradient information is utilized in the proposed region growing method to control the segmentation process. This also shows that the proposed method uses more local information than global information. Post-processing may be necessary for images with weak edges.

Experimental results for complicated real images are provided in Fig. 3.25. Gaussian smoothing is not utilized here since these images are of good quality. It can be seen that the proposed method is very efficient, even for complex images. As in Fig. 3.24, post-processing may be necessary for better results.

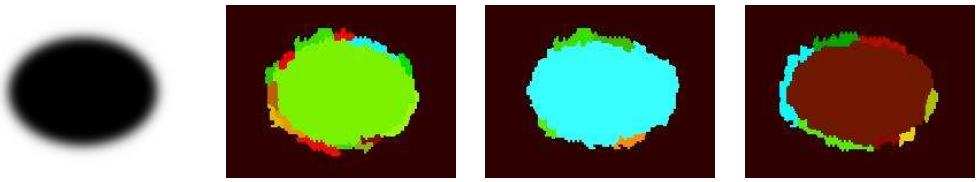
3.3.7 Conclusions and Future Work

In this section, a mathematical analysis of the initialization problem of the Chan-Vese model is provided. This analysis shows that the initialization problem is caused by the top-down manner in which region information is used. A new bottom-up image segmentation method is proposed to solve this problem. It is based on region growing, region competition, and the Mumford-Shah functional. This method works very well for complicated images. It is very efficient, easy to implement and robust to noise. Experimental results show this method is able to quickly segment complex images. A fast curve evolution method is also proposed for the Chan-Vese model. This method does not need to solve PDEs and works very well for images without strong noise.



(a) (b) (c) (d)

Figure 3.23: Effects of ν on segmentation results. (a) A gray image with different distributions (300 * 300). (b) Segmentation of (a), $\nu = 800$, CPU = 24.79s. (c) Segmentation of (a), $\nu = 1200$, CPU = 10.29s. (d) Segmentation of (a), $\nu = 2000$, CPU = 3.91s.



(a) (b) (c) (d)

Figure 3.24: The ability of the proposed method for weak edges. (a) A gray image with weak edges (200 * 150). (b) Segmentation of (a), $\nu = 5000$, CPU = 1.34s. (c) Segmentation of (a), $\nu = 10000$, CPU = 1.11s. (d) Segmentation of (a), $\nu = 20000$, CPU = 1.12s.

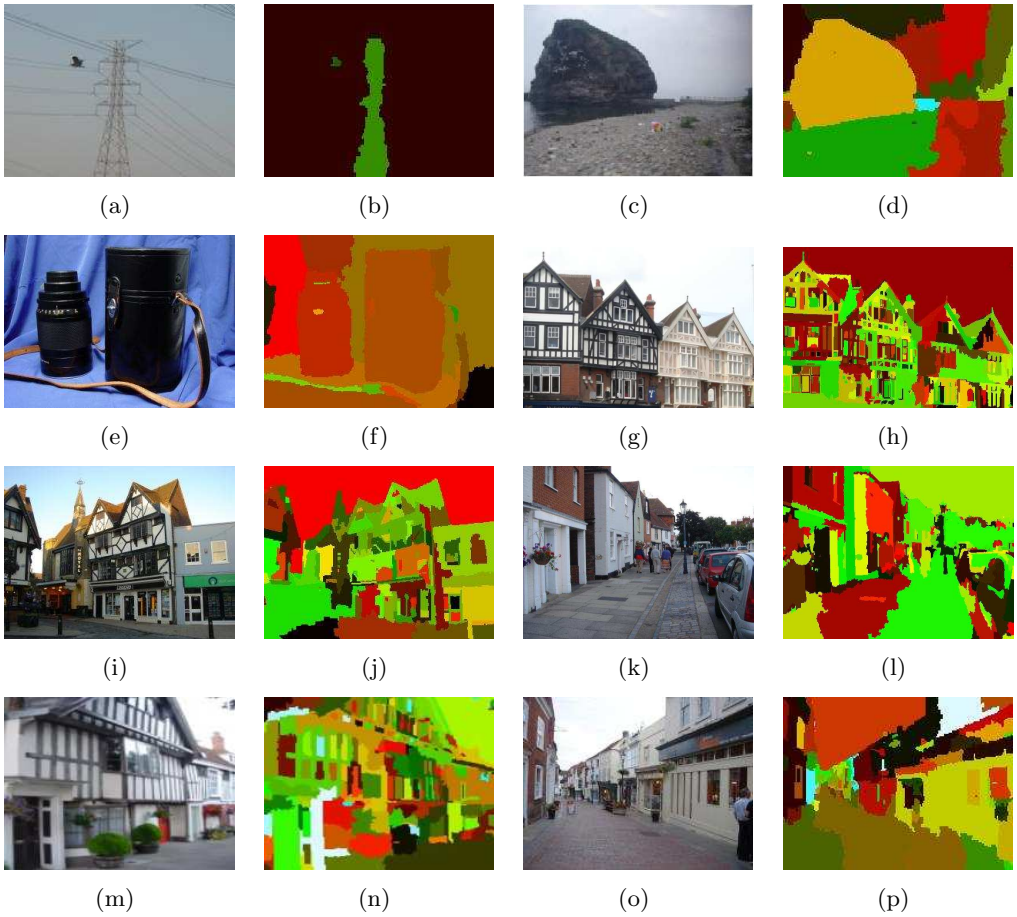


Figure 3.25: Experimental results for real images. (a) An original real image (200 * 150). (b) Segmentation of (a), $\nu = 1500$, CPU = 1.09s. (c) An original real image (200 * 150). (d) Segmentation of (c), $\nu = 2500$, CPU = 1.06s. (e) An original real image (300 * 225). (f) Segmentation of (e), $\nu = 15000$, CPU = 2.66s. (g) An original real image (400 * 278). (h) Segmentation of (g), $\nu = 3000$, CPU = 8.04s. (i) An original real image (300 * 250). (j) Segmentation of (i), $\nu = 5000$, CPU = 3.33s. (k) An original real image (500 * 375). (l) Segmentation of (k), $\nu = 3000$, CPU = 9.47s. (m) An original real image (200 * 150). (n) Segmentation of (m), $\nu = 3000$, CPU = 1.08s. (o) An original real image (200 * 150). (p) Segmentation of (o), $\nu = 3000$, CPU = 1.04s.

The following four topics are suggested for future research. First, region growing alone may not be enough for the minimization of the Mumford-Shah functional. Boundaries between large regions need to be evolved to minimize the functional, as in [163]. This will generalize the proposed method and make region growing a special case. Since the image has been segmented into several regions, multiple junctions have to be processed if using curve evolution for boundary evolution, as in [95]. Second, the proposed method sometimes over-segments images because it is hierarchically bottom-up. Post-processing may be necessary in some cases. The proposed method may also be combined with the top-down method to utilize global information. Third, segmentation results of the proposed method are sensitive to the ν parameter, which is characteristic for the Mumford-Shah model. Better results may be generated if the value of ν is dynamically chosen. Adaptive control techniques [128] are good candidates for this problem. Finally, since the computational load of the proposed method increases exponentially with image size, fast implementation techniques, such as multi-scale methods [115] and parallel methods, may be utilized to increase the speed of the proposed method.

3.4 An Improvement of the Image Segmentation Method using Region Competition and the Mumford Shah Functional

In the bottom-up hierarchical image segmentation method shown above, the well-posed Mumford-Shah functional (2.2) is minimized while avoiding the initialization problem and reducing the computational load. The method utilizes region growing to minimize the energy functional. At the beginning, each pixel in the image is taken as a region. In this way, no curves are initialized and the initialization problem is avoided. During image segmentation, two neighboring regions are merged so long as the action will decrease the energy functional. Since the well-posed Mumford-Shah model tends to segment images into piecewise constant regions, the pixels in the merged region are set to be the average intensity in the region. Such a process is repeated until no merges of regions can decrease energy. This method is shown to be efficient, robust in the presence of strong noise, and capable of handling complicated images.

The above method, however, can be improved. Region growing is not enough for energy minimization in some cases. Although the merges of two regions may increase the energy functional, moving their common boundaries may decrease the energy functional. Thus, region competition on the boundaries between neighboring regions can be beneficial. In fact, region growing can be taken as a special case of the region competition. This idea is similar to the ideas [143] [163] and the Chan-Vese models presented in [27] [150]. But there are differences. First, region competition in the present case results in changes between regions, not the evolution of curves. Second, region competition is performed only when region growing can not decrease the energy functional, lowering the computational cost.

Several methods may be applied for region competition, such as in [27] and [163]. The bimodal Chan-Vese model [27] is a good choice because the geometric curve evolution in [27] can handle topological changes automatically. But [27] may be time-consuming. In the implementation, therefore, a simplified method of [27] as proposed in [112] is applied. The method evolves the boundary based on the signs of the region term $-(I - c_1)^2 + (I - c_2)^2$

and ψ, and regularizes the boundary using Gaussian smoothing. Since it does not solve PDEs, it is very efficient.

The improved method can be seen as an extension of the multi-phase Chan-Vese model when numerous initial curves are introduced so that every pixel in the image is taken as a region. It can also be viewed as a method to select the starting point for energy minimization to try to reach the the global minimum.

3.4.1 Experimental Results

Experimental results using the improved image segmentation method are shown in this section. The proposed method is implemented on a computer which has two Intel(R) Pentium(R) 3.2GHz CPUs, 2G bytes RAM, and runs the Red Hat Enterprise Linux operating system. The CPU times given in this section are the sums of system CPU times and user CPU times. The system CPU time is usually very small, typically 0.01 - 0.08 second.

Fig. 3.26 shows the comparison between the improved method and the method reported in [112]. For images with weak edges Fig. 3.26(a) and Fig. 3.26(f), segmentation results of the improved method (Fig. 3.26(c), Fig. 3.26(e) and Fig. 3.26(h)) are much better than the results of the previous method. The boundaries between regions are more regularized and the results are closer to the original image. It can also be seen that both methods are very efficient.

Segmentation results for more complicated images are shown in Fig. 3.27. Fig. 3.27(b) illustrates that the method is stable in the presence of large noise in Fig. 3.27(a). Fig. 3.27(d), Fig. 3.27(f), Fig. 3.27(h) and Fig. 3.27(p) are the segmentation results of Fig. 3.27(c), Fig. 3.27(e), Fig. 3.27(g) and Fig. 3.27(o), respectively. These results demonstrate that the proposed method is able to handle complicated images and is very efficient. Fig. 3.27(j), Fig. 3.27(k), Fig. 3.27(m) and Fig. 3.27(n) show the influence of the parameter ν. With ν increasing, fewer objects are segmented. The proper selection of ν for a specific image is a topic under research.

3.4.2 Summary

This section generalizes the results reported in [112]. A more comprehensive analysis is provided for the Chan-Vese models, and an improved image segmentation method is proposed using region competition and the Mumford-Shah functional. The proposed method is shown to be efficient, robust in the presence of noise, and able to handle complicated images. Furthermore, it generates better results than the previous method for images with weak edges with only a minor increase in computational cost. Suggested topics for future research include the dynamic selection of the parameter ν and the parallel implementation of the proposed method.

3.5 Efficient Implementations of Image Segmentation

In this section, fast implementation methods of curve evolution are introduced. Narrow band method and multi-scale method [116] are briefly introduced first. Parallel computing based on MPI will be illustrated next.

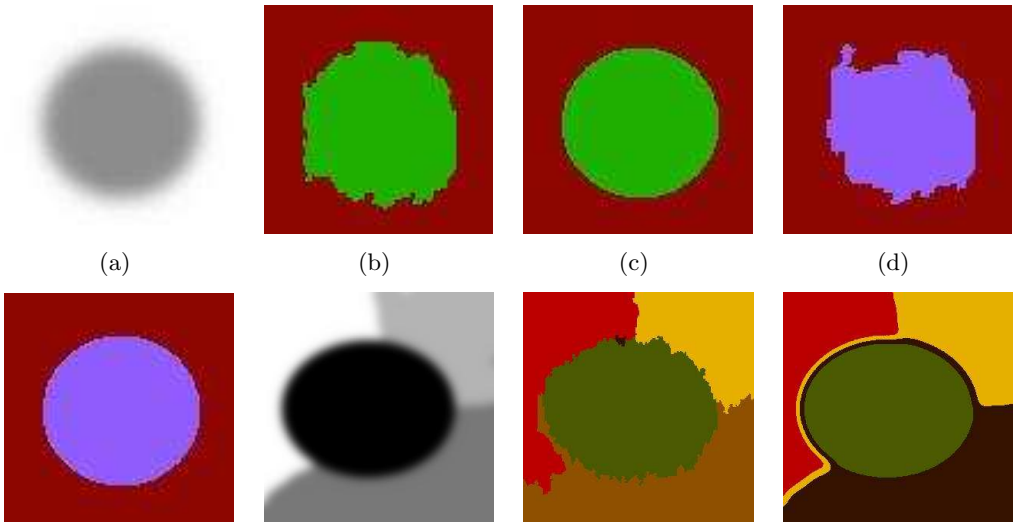


Figure 3.26: Comparison of the improved model with the previous model. (a) Original image (100 * 100). (b) Segmentation of (a) using the previous model, $\nu = 5000$, CPU = 0.27s. (c) Segmentation of (a) using the improved model, $\nu = 5000$, CPU = 0.33s. (d) Segmentation of (a) using the previous model, $\nu = 12000$, CPU = 0.26s. (e) Segmentation of (a) using the improved model, $\nu = 12000$, CPU = 0.27s. (f) Original image (300 * 300). (g) Segmentation of (f) using the previous model, $\nu = 18000$, CPU = 2.69s. (h) Segmentation of (f) using the improved model, $\nu = 18000$, CPU = 3.52s.

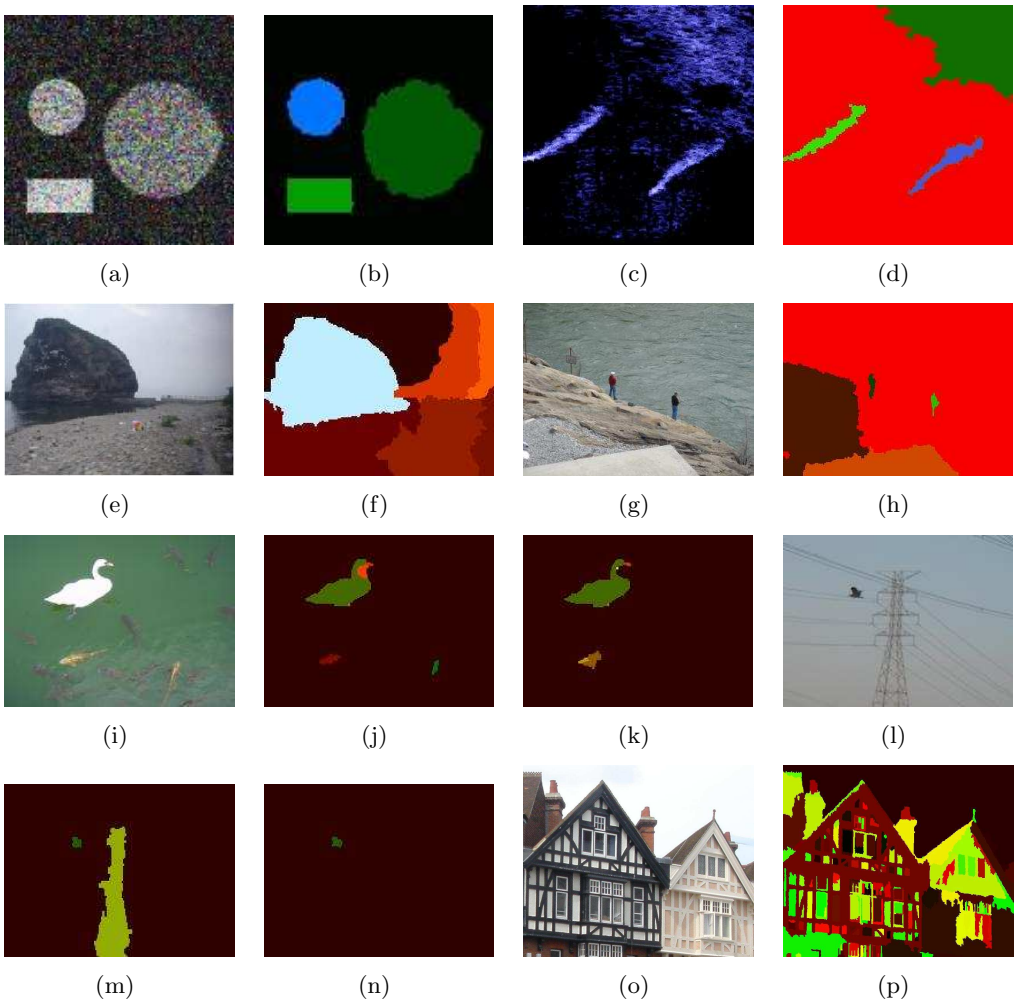


Figure 3.27: Experimental results for real images. (a) Original image (128 * 128). (b) Segmentation of (a), $\nu = 1200$, CPU = 0.82s. (c) Original image (255 * 266). (d) Segmentation of (c), $\nu = 7000$, CPU = 4.61s. (e) Original image (200 * 150). (f) Segmentation of (e), $\nu = 2500$, CPU = 2.23s. (g) Original image (300 * 225). (h) Segmentation of (g), $\nu = 10000$, CPU = 2.64s. (i) Original image (200 * 150). (j) Segmentation of (i), $\nu = 600$, CPU = 1.82s. (k) Segmentation of (i), $\nu = 700$, CPU = 1.96s. (l) Original image (200 * 150). (m) Segmentation of (l), $\nu = 600$, CPU = 0.93s. (n) Segmentation of (l), $\nu = 1500$, CPU = 0.91s. (o) Original image (698 * 581). (p) Segmentation of (o), $\nu = 3000$, CPU = 27.16s.

3.5.1 Narrow Band Method

Level set methods embed a one-dimensional curve in a two-dimensional image domain. Curve evolution is then transformed to the update of the values in the image domain. Since the evolution of the curve itself is of interests, it is only necessary to update those points near the zero level set of the image. A narrow band of points near the evolving curve, instead of the whole image, are updated in each iteration. Once the evolving curve reaches the boundary of the narrow band, a re-initialization step will be made and a new narrow band will be generated. Such a process iterates until the curve evolution stops. Fig. 3.28 illustrates the idea of the narrow band method.

Such a process greatly reduces the computational load of curve evolution in update, velocity extension and the determination of time steps using CFL conditions [130]. It requires $O(kN)$ operations instead of the original $O(N^2)$ operations in one iteration in two dimensions, where N represents the length of the image and k is the number of points in the narrow band.

The above narrow band methods was introduced by Chopp [31] and has been widely utilized. See [130] and the references therein for more details.

3.5.2 Multi-scale Method

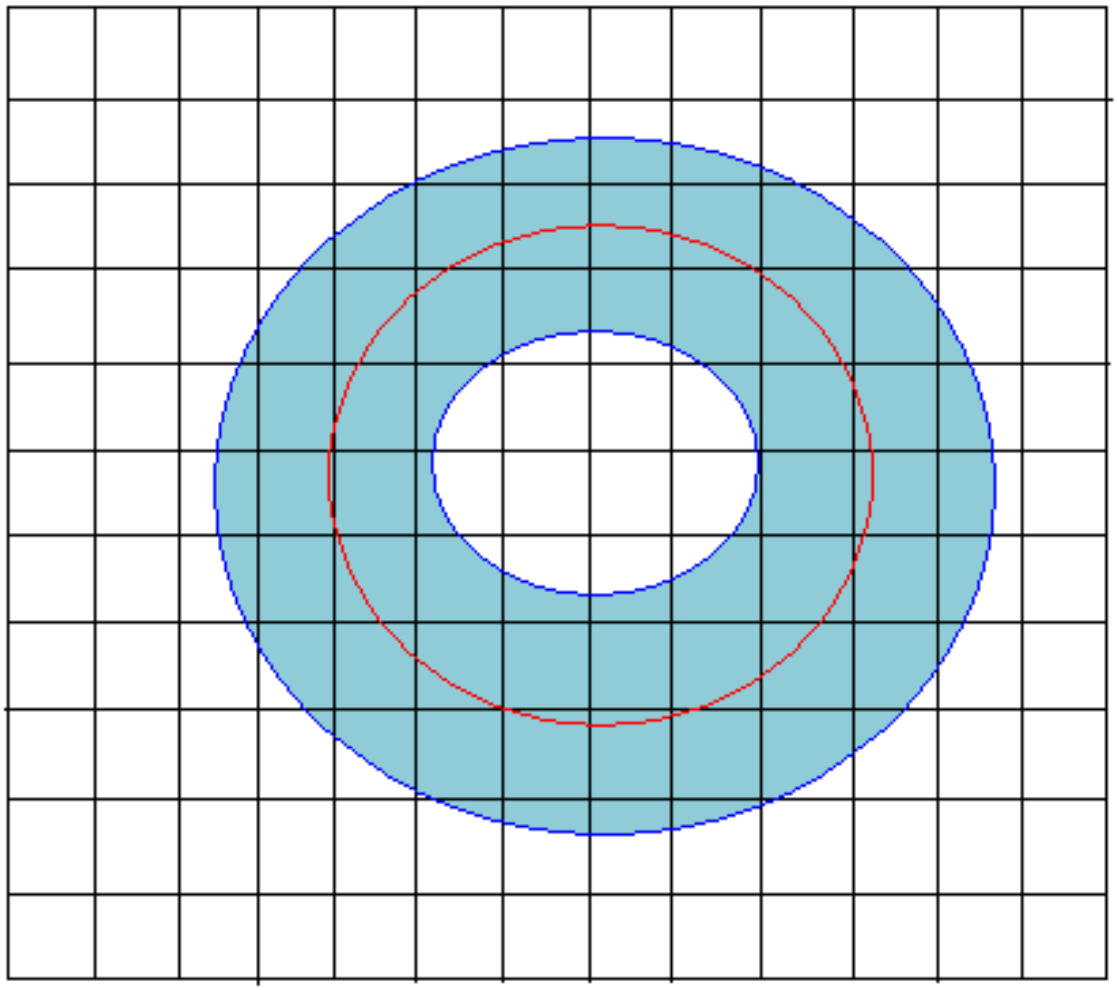
Multi-scale method is another way to greatly reduce the computational load of curve evolution. It was first proposed by Paragios [116]. The main idea is to perform curve evolution in different scale spaces, which are subsets of the original image by interpolation. Then the curve evolution results in a smaller scale will be extrapolated to a larger scale as its initial solution for curve evolution. Such a process iterates until the original image is successfully segmented.

3.5.3 Parallel Computing Method

During curve evolution, the update of each point in the image is only related to its four neighborhoods. This characteristic provides a good way for parallel computing. Instead of performing curve evolution in a serial computer, we can make use of parallel machines for the update of the level sets, with each parallel node responsible for the points in a specific portion of the narrow band. Since most of the work in curve evolution lies in the update of the level sets, parallel computing will improve the speed of curve evolution. Fig. 3.29 demonstrates one solution of load balancing during the implementation of parallel computing.

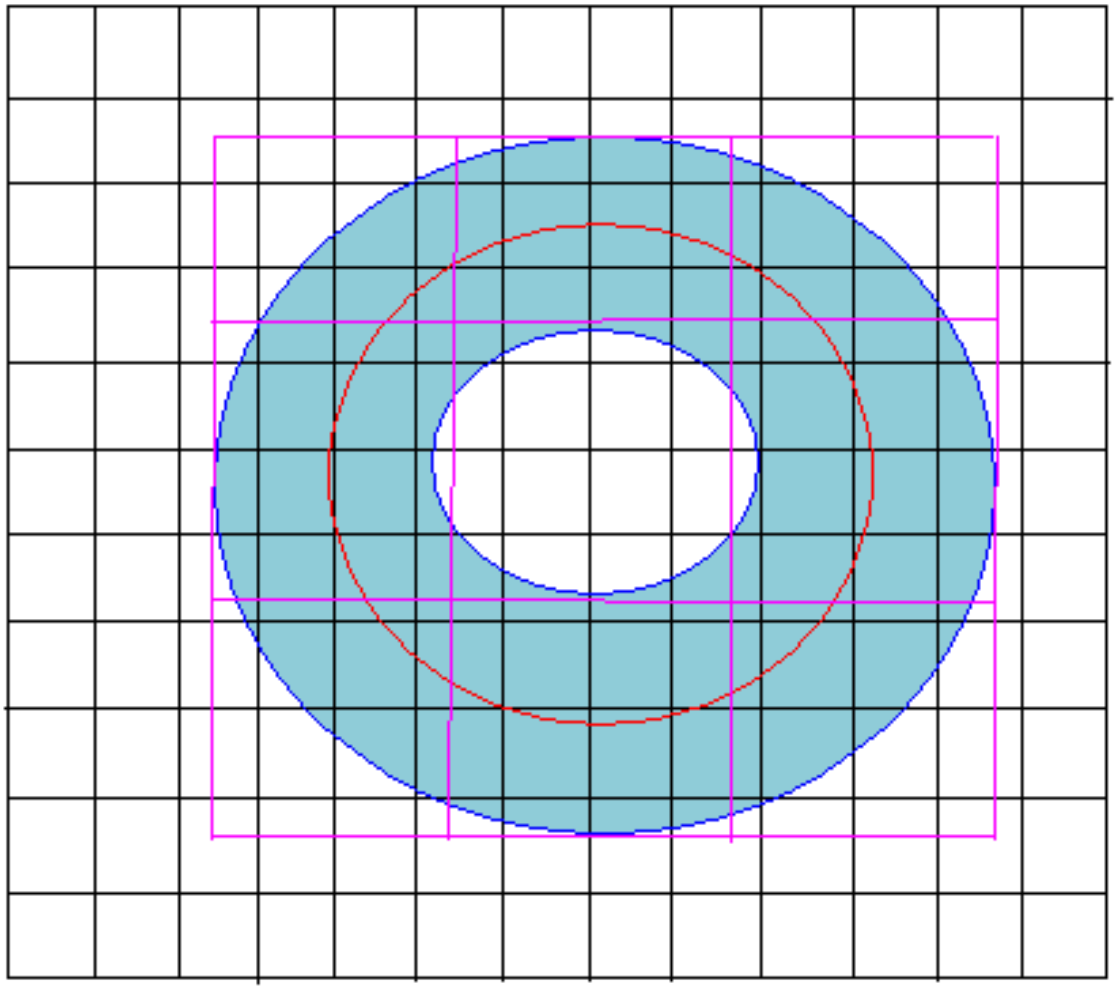
3.5.4 Experimental Results

Experimental results from the proposed framework are provided in this section. Since the effects of the narrow band method are well established and easy to understand, the results are mainly focused on the demonstration of the multi-scale method and the parallel computing method. Each of the parallel nodes in the experiments has two AMD $Athlon^{TM}$ 1.4GHz CPU, 1G bytes RAM, and runs the Red Hat Linux operating system. Curve evolution of serial computing is performed on one such node. For parallel computing, the message passing interface (MPI) scheme, which is the de facto industrial standard for parallel computing, is applied, and its LAM implementation is utilized.



(a)

Figure 3.28: Illustration of narrow band method. Red line represents the evolving curve. Blue region represents the narrow band in which the points need update.



(a)

Figure 3.29: Illustration of parallel computing for curve evolution. Red line represents the evolving curve. Blue region represents the narrow band. Pink lines show one solution to separate the narrow band into small regions, each for one parallel slave node.

In the parallel implementation, one node acts as a master node, and all the others act as slave nodes. The master node divides the whole task into several subtasks, sends them to each slave node, and processes the results. The master node also makes sure the CFL condition [130] is satisfied at each iteration and re-initializes the whole process when the boundary of the narrow band is reached. In our implementation, dynamic resource allocation as shown in Fig. 3.29 is utilized. This allocation method makes it easy for the master node to allocate tasks.

Fig. 3.30 shows the results of curve evolution using the multi-scale method. An image (a) of the original size 225 by 300 is scaled down a smaller image, which has the size 75 by 100. (b) displays the scaled image with the evolving curve initialized at the boundary. The curve evolution result of (b) is shown in (c), which takes 28 iterations. (d) represents the image which is scaled back to its original size using extrapolation, with the segmentation result in (c) as its initial contour. After 34 iterations, segmentation results (e) is achieved. The whole process takes 10.96 seconds. For comparison, segmentation result of (a) without using multi-scale methods is shown in (f). The segmentation process takes 49 iterations and 24.03 seconds. There is not a big difference between the results of (e) and (f), but the curve evolution speed is much faster using multi-scale methods.

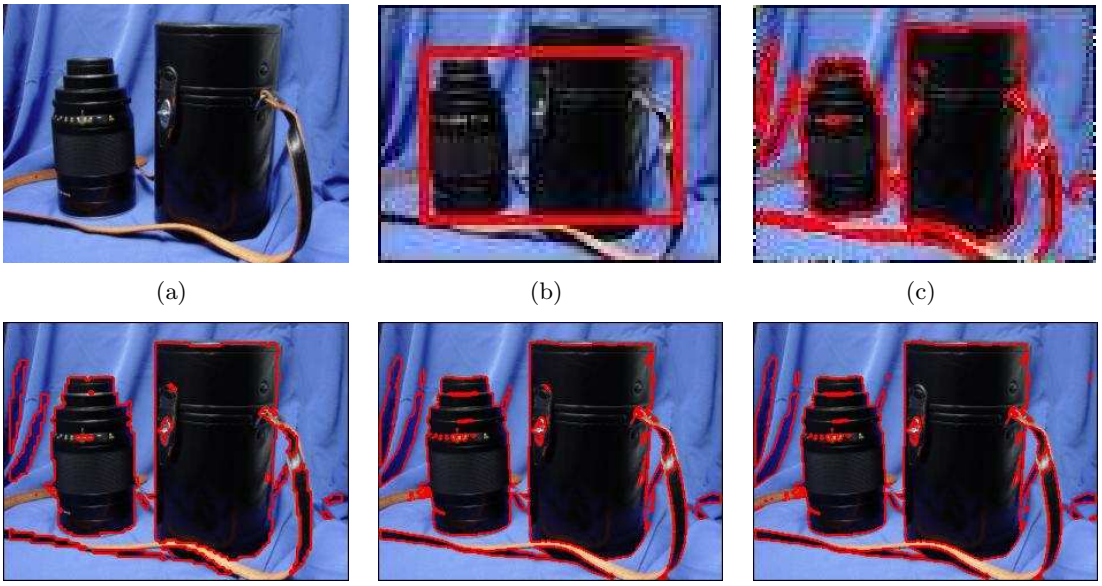
Fig. 3.31 shows the effects of the multi-scale method on the segmentation of a large image using curve evolution. (a) shows the original image of size 1920 by 2400. (b)-(d) represents the segmentation results of (a) in different scales. The image in (b) has size 128 by 160 and takes 272 iterations. (c) has size 640 by 800 and takes 100 iterations evolved from (b). (d) has size 1920 by 2400 and takes 100 iterations evolved from (c). From (a) to (d), the process takes 849.16 seconds. (e) and (f) shows the segmentation results of the original image without using multi-resolution after 800 iterations and 1600 iterations, respectively. It takes more than 2.5 hours to evolve the curve initialized at the boundary from (a) to (f). These results demonstrate the power of multi-scale method in curve evolution.

Fig. 3.32 displays the experimental results of curve evolution using parallel computing. (a) shows the original image of size 225 by 300 with initial curve at its boundary. (b) and (c) shows the results using 2 slave nodes and 4 slave nodes separately. No obvious differences can be perceptually observed between them. As for the time, (b) takes 19.18 seconds to be generated and (c) takes 20.26 seconds. Both of them run a little faster than serial computing (24.03 seconds).

In summary, a fast implementation framework of curve evolution is proposed, which combines narrow band method, multi-scale method and parallel computing method. Experimental results shows that the framework can effectively reduce the computational time during curve evolution. The future work includes the improvement of dynamic resource allocation in parallel computing and the implementation to combine multi-scale method and parallel computing.

3.6 Summary

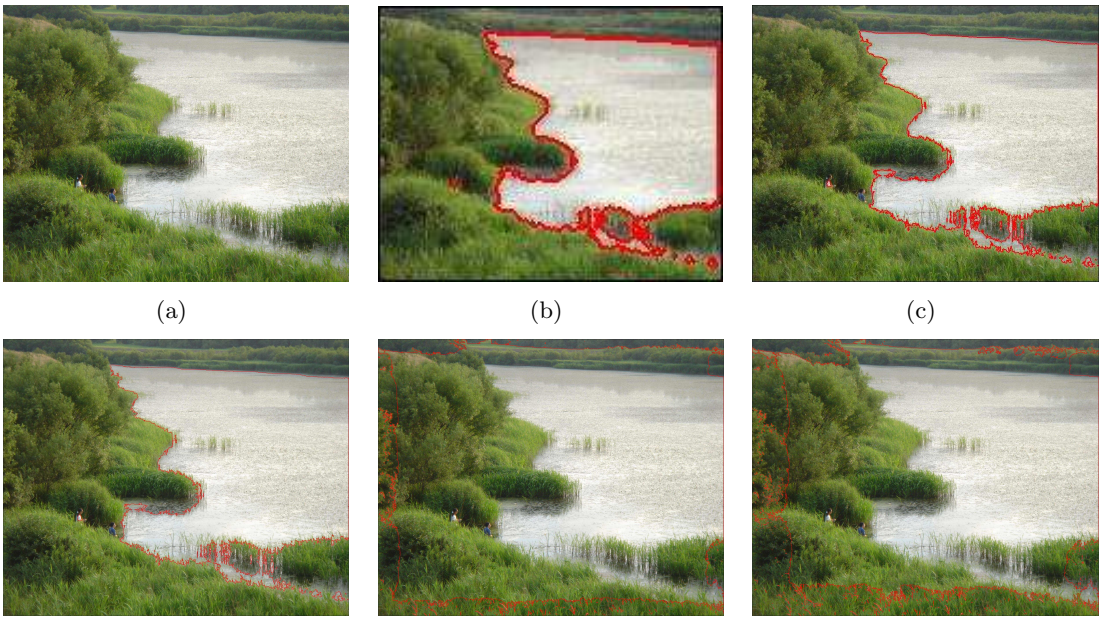
A new geometric snake model [109] is presented which utilizes both gradient and region information for image segmentation in this chapter. It is implemented by minimizing an energy functional and solving PDEs. This method works well in presence of weak edges. An integrated framework for curve evolution and anisotropic diffusion [108] is developed using PDEs and variational methods too. This method outperforms [109] in that it is more stable to the effects of noise. Efficient bottom-up segmentation methods [107] [111] [112] [113] are



(a) (b) (c)

(d) (e) (f)

Figure 3.30: Multi-scale curve evolution. (a) Original image, size = 225 * 300. (b) Scaled image with initial contour at the boundary, size = 75 * 100. (c) Segmentation result of the scaled image after 28 iterations. (d) Extrapolated image of (c) back to its original size. (e) Segmentation results of (d) after 34 iterations. During (a)-(e), user time = 10.906s, system time = 0.052s, elapsed time = 10.96s, CPU time percentage 99.9%. (f) Segmentation results of (a) without using multi-scale method (49 iterations, user time = 23.875s, system time = 0.103s, elapsed time = 24.03s, CPU time percentage 99.9%).



(a) (b) (c)

(d) (e) (f)

Figure 3.31: Multi-scale curve evolution on a large image. (a) Original image, size = 1920 * 2400. (b) Segmentation of scaled image after 272 iterations, size = 128 * 160. (c) Segmentation result of the scaled image from (b) after 100 iterations, size = 640 * 800. (d) Segmentation result of the original image from (c) after 100 iterations, size = 1920 * 2400. During (a)-(e), user time = 843.47s, system time = 5.37s, elapsed time = 849.16s, CPU time percentage 99.9%. (e) Segmentation results of (a) without multi-scale method after 800 iterations. (f) Segmentation results of (a) without using multi-scale method after 1600 iterations. It takes more than 2.5 hours to get (f) from (a) with the initial curve at the boundary.

(a) (b) (c)

Figure 3.32: Parallel computing of curve evolution. (a) image with initial contour, size = 225 * 300. (b) Segmentation results using 2 parallel slave nodes, user time = 0.148s, system time = 0.029s, elapsed time = 19.18s, CPU time percentage 0.8%. (c) Segmentation results using 4 parallel slave nodes, user time = 0.154s, system time = 0.029s, elapsed time = 20.26s, CPU time percentage 0.8%.

developed to minimize the piecewise constant Mumford-Shah functional. Future research may be focused on more efficient computations of these methods, for example, using graphic processors shown in Section 6.1.

Chapter 4

Preferential Image Segmentation using Mathematical Morphologies

A novel preferential image segmentation method is proposed in this chapter, which incorporates image segmentation and object recognition using mathematical morphologies. This method is able to preferentially segment objects which have similar intensities and boundaries with those objects in prior images. A tree of shapes [14] [21] [24] [61] [91] [99] is utilized to represent the content distributions in images, and curve matching is applied to compare the boundaries. The algorithm is invariant to contrast change and similarity transformations such as translation, rotation and scale. The experimental results show that the proposed approach is promising for applications such as object segmentation and video tracking with cluttered backgrounds. The methods presented in this chapter belong to the integration of image segmentation and object recognition in the proposed scheme 1.2.2 in Chapter 1, and they utilize techniques from mathematical morphologies.

4.1 Introduction

The objective of image segmentation is to segment an image into several regions so that the contents of each region represent meaningful objects. The segmentation results can then be utilized for post-processing stages such as object recognition. By image segmentation, the post-processing stages will become less challenging and computationally simpler.

Image segmentation dates back to the early 1980's. Edge detection methods such as the Canny detector [18] were widely applied for this task. Edge detection methods utilize intensity gradients to detect the boundaries of objects. However, edge detection methods usually generate edges that are not closed contours, and this causes difficulties for later processing such as object recognition. It has been shown [60] that it is hard to construct stable edge detection methods even using a multi-scale theory because of the edge-linking issue across scales and multiple-thresholding issues at each scale.

Curve evolution methods [105] [104] [130] have been popular for image segmentation since the early 1990's. These methods evolve the initialized curve(s) to the boundaries of objects in an image for its segmentation. The evolution of the curves may be driven by image gradient information [23] [78], region information [27] [71] [115] or their combinations [146]. These methods are theoretically solid and numerically stable. Moreover, these methods generate image segments enclosed by closed contours, which leads to straightforward post-processing.

(a) (b)

Figure 4.1: Illustration of preferential image segmentation. (a) The video frame in which only the car is of interests. (b) The prior image of the car.

Most curve evolution methods, however, are unsupervised and therefore are able to only segment "cartoon-like" simple images. For complicated cases such as triple junctions and images with cluttered background, these methods have to utilize hierarchical methods [115] for remedy, which are usually computationally intense.

The utilization of prior information, therefore, seems to be necessary for curve evolution methods in complicated cases of image segmentation. Several methods [38] [40] [47] [52] [53] [66] [87] [88] [114] have been proposed that utilize prior information for supervised image segmentation. These methods usually propose new variational energy functionals which integrate both the prior information and the gradient/region information in the image to be segmented. The minimizations of these functionals can lead to segmentation results. Shape priors are utilized in [38] [40] [47] [88]. Both intensity priors and shape priors are applied in [87]. Natural image statistics are utilized for natural image segmentation in [66]. These methods usually work better than unsupervised methods. But shape priors are primarily incorporated in supervised segmentation methods, and the information contained in the intensities has not always been effectively utilized. For example, the methods in [52] [53] utilize the intensity histograms only, and the spatial information is ignored. Furthermore, these methods usually have initialization problems because the energy functionals have multiple local minimums. The methods are therefore sensitive to the initial locations of the evolving curves.

The above methods tend to segment the whole image into several regions, which is challenging for images with cluttered background. On the other hand, this is not always necessary in real applications. The user may be interested in finding only the location of objects of interest. For example, only the car in Fig. 4.1(a) is of interests in video tracking problems. The cluttered background need not to be segmented in this application. The task, therefore, changes to segment from the image the regions whose contents are similar to the car in the prior image Fig. 4.1(b). Image segmentation and object recognition are combined in this sense. This is the key idea of "preferential image segmentation", which means to preferentially segment objects of interests from an image.

The idea of preferential image segmentation bears some similarities to object detection methods from images [1]. These methods detect the existence and rough location of objects in an image, e.g. in [1], using a sparse, part-based object representation and a learning

method. However, these methods do not provide the exact locations of the object boundaries, which is required by image segmentation. Supervised image classification methods in [12] [126] and the image parsing method in [148] are also similar to preferential image segmentation. The image classification methods in [12] [126] generalize the level set methods to classify an image into multiple regions by means of wavelets [12] and variational methods [126], respectively. The image parsing method [148] utilizes probabilistic methods to unify segmentation, detection and recognition.

A novel preferential image segmentation method is proposed in this chapter from the perspective of image topologies. This method is motivated by utilizing prior information into curve evolution models. However image topologies may provide even better results for complicated cases. The proposed method utilizes the tree of shapes [97] [99] to represent images. This representation provides a hierarchical tree for the objects contained in the level sets of images. The hierarchical structure is utilized to select the candidate objects from the image. The boundaries of the selected objects are then compared with that of the objects selected from the prior image. By means of the tree of shapes and curve matching, the proposed method is able to preferentially segment objects with closed boundaries from complicated images. It is more straightforward to utilize prior information in this way than with the curve evolution methods, and there is no initialization problem. Furthermore, the method is invariant to contrast change and similarity transformations such as translation, rotation and scale. The method has been shown to work in presence of noise.

This chapter is organized as follows. Background information on image representation using the tree of shapes and background information on curve matching are provided in Section 4.2. A novel preferential image segmentation method is proposed in Section 4.3, followed by Section 4.4 which shows experimental results. Section 4.5 describes the evaluation of the performance of the proposed method using a provided large image data set. A user interface incorporating the image database, a web browser and the proposed method is presented in Section 4.6. A summary and future research directions are provided in Section 4.7.

4.2 Background

Background information for the proposed method is introduced in this section. Section 4.2.1 introduces how an image can be represented using connected components of sets of finite perimeters [5], in which a space of functions of weakly bounded variations (WBV) is utilized. Section 4.2.2 shows how an image is represented using the tree of shapes, where a tree structure is introduced for image representation. Section 4.2.3 describes the relationship between color and geometry in natural images. Section 4.2.4 introduces the techniques of planar curve matching, which can be utilized to compare the boundaries of different objects. The scale space theory [98] for the tree of shapes introduced in Section 4.2.5 provides the potential extension of the proposed method to multi-scale analysis.

4.2.1 Image Representation using Connected Components of Sets of Finite Perimeters

This subsection provides a review of the theory of image representation using connected components of set of finite perimeters in the space of functions with weakly bounded variations (WBV), as introduced in [5] [13] [25]. The theory provides a theoretical foundation for

image processing using mathematical morphologies and the preferential image segmentation methods developed in this research.

Several definitions are provided before the properties and the theorems are introduced. Both the definitions and the theorems are reproduced from [5] [13] [25] as noted to provide background information to the reader. The references should be consulted for additional details and proofs.

For a Lebesgue measurable set $E \in \mathbb{R}^N$, $N \geq 2$, its Lebesgue measure in [124] is denoted as $|E|$, and its upper densities and lower densities of E at x are respectively defined as

$$\overline{D}(E,x) = \limsup_{r\to 0^+} \frac{|E \cap B(x,r)|}{|B(x,r)|}, \quad \underline{D}(E,x) = \liminf_{r\to 0^+} \frac{|E \cap B(x,r)|}{|B(x,r)|}, \tag{4.1}$$

where $B(x,r)$ represents a ball centered at x with radius r. If $\overline{D}(E,x) = \underline{D}(E,x)$, then it is called the density of x at E, denoted by $D(x,E)$.

The essential interior $\mathring{E}^M$, the essential closure $\overline{E}^M$ and the essential boundary $\partial^M E$ of a measurable set E are defined in [5] as follows:

$$\mathring{E}^M := \{x : D(x,E) = 1\}, \quad \overline{E}^M := \{x : \overline{D}(x,E) > 0\} \tag{4.2}$$

$$\partial^M(E) := \overline{E}^M \cap \overline{\mathbb{R}^N \setminus E}^M = \{x : \overline{D}(x,E) > 0, \overline{D}(x,\mathbb{R}^N \setminus E) > 0\}. \tag{4.3}$$

From the Lebesgue differentiation theorem [124] (page 121), the measure theoretic interior of $\mathring{E}^M$ is $\mathring{E}^M$ itself, and [5]

$$\partial^M(E) = \mathbb{R}^N \setminus \left(\mathring{E}^M \cup \overbrace{\mathbb{R}^N \setminus E}^{\circ\,M} \right) \tag{4.4}$$

A measurable set $E \subseteq \mathbb{R}^N$ has *finite perimeter* [5] in $\mathbb{R}^N$ if there exists a positive finite measure μ and a Borel function $\nu_E : \mathbb{R}^N \to S^{N-1}$ (called generalized inner normal to E) such that the following generalized Gauss-Green formula holds

$$\int_E \text{div } \phi \; dx = -\int_{\mathbb{R}^N} < \nu_E, \phi > d\mu \quad \forall \phi \in {C_c}^1(\mathbb{R}^N, \mathbb{R}^N) \tag{4.5}$$

Here the measure $\nu_E \mu$ is the distributional derivative of χ_E, which will be noted by D_{χ_E}, while $\mu = |D_{\chi_E}|$ is its total variation. The *perimeter* $P(E,B)$ of E in a Borel set $B \subseteq \mathbb{R}^N$ is defined [5] by $|D_{\chi_E}|(B)$. $P(E)$ will be used instead when $B = \mathbb{R}^N$.

By Riesz theorem [124], a measurable set $E \subseteq \mathbb{R}^N$ has finite perimeter if and only if

$$\sup \left\{ \int_E \text{div } \phi \; dx : \phi \in {C_c}^1(\mathbb{R}^N, \mathbb{R}^N), |\phi| \leq 1 \right\} < \infty \tag{4.6}$$

and in this case the supremum equals the perimeter [5].

The space of functions of bounded variations $BV(\Omega)$ is defined [50] as the space of all those functions $u \in L^1(\Omega)$ whose distributional derivative is representable as a $\mathbb{R}^N$-valued measure $Du = (D_1 u, D_2 u, ..., D_N u)$ with finite total variation in Ω,

$$\int_\Omega u \text{ div } \phi \; dx = -\sum_{i=1}^N \int_\Omega \phi_i dD_i u \quad \forall \phi \in [C_c^1(\Omega)]^N \tag{4.7}$$

The total variation $|Du|$ of a BV function u is defined [50] as the total variation of the vector measure Du. The norm of the space $BV(\Omega)$ has the following property: $\|u\|_{BV} = \|u\|_{L^1} + |Du|(\Omega)$. The space $BV_{loc}(\Omega)$ is the space of all those functions that belong to $BV(\tilde{\Omega})$ for every open set $\tilde{\Omega} \subset\subset \Omega$.

The relationship between sets of finite perimeter and the functions of bounded variations is easily seen [50]: a subset $E \subseteq \mathbb{R}^N$ has finite perimeter in Ω *if and only if* $u = \chi_E \in BV_{loc}(\Omega)$ and $|Du|(\Omega) < \infty$.

The space of functions of bounded variations $BV(\Omega)$ is utilized frequently for image processing, as advised in [122]. This space is more general than the space of continuous functions $C_0(\Omega)$ (or piecewise continuous functions used by the Mumford-Shah model [100]). The methods designed in this space are therefore more general. The BV space still has limitations though, e.g., a black-white image can not be modeled as a BV funtion.

For image processing problems such as image segmentation, specific sets with finite perimeters which correspond to the objects of interest are desired. It is therefore necessary to study the decomposability (see the definition below) of a set with finite perimeter.

Let $E \subseteq \mathbb{R}^N$ be a set with finite perimeter. E is defined in [5] to be *decomposable* if there exists a partition (A, B) of E such that $P(E) = P(A) + P(B)$ and both $|A|$ and $|B|$ are strictly positive. E is *indecomposable* [5] if it is not decomposable. An important decomposability theorem, stated and proved by [5], is reproduced below.

Theorem 4.2.1 (Decomposition theorem [5])**.** *Let E be a set with finite perimeter in $\mathbb{R}^N$. Then there exists a unique finite or countable family of pairwise disjoint indecomposable sets $\{E_i\}_{i\in I}$ such that $|E_i| > 0$ and $P(E) = \sum_i P(E_i)$. Moreover,*

$$\mathcal{H}^{N-1}\left\{\mathring{E}^M \setminus \bigcup_{i\in I} \mathring{E}_i^M\right\} = 0 \tag{4.8}$$

and the E_i's are maximal indecomposable sets, i.e. any indecomposable set $F \subseteq E$ is contained (mod $\mathcal{H}^N$) in some set E_i.

$\mathcal{H}^{N-1}$ in the above theorem represents the Hausdorff measure of dimension $(N-1)$ [124]. The details of the proof are presented in [5].

This theorem builds the theoretical foundation of the decomposition of an image into several subsets when the image is represented by a function of bounded variation and its domain by a set with finite perimeter. This theorem guarantees the decomposability of the image domain, and thus provides a way for image segmentation. A definition is provided in this way.

Definition 4.2.2. *[5] In view of the previous theorem, the sets E_i are defined to be the M-connected components of E and this family is denoted by $\mathcal{CC}^M(E)$. The index set I is always chosen as a subset of $\mathbb{N}$, with $0 \in I$.*

The relationship between the M-connected components and the connected components of a set with finite perimeter is provided in [5].

The concepts of "hole" and "saturation" [5], whose definitions are based on the decomposition theorem introduced above, provide a way to study the spatial relationship between sets of finite perimeters. The decomposition of the measure theoretic boundary also requires these concepts.

Definition 4.2.3 (Holes, saturation [5]). *Let E be an indecomposable set. A hole of E is defined as any $M-$connected component of $\mathbb{R}^N \setminus E$ with finite measure. The Saturation of E, denoted by sat(E), is defined as the union of E and its holes. In the general case when E has finite perimeter, we define*

$$sat(E) := \bigcup_{i\in I} sat(E_i) \quad where \quad \mathcal{CC}^M(E) = \{E_i\}_{i\in I} \tag{4.9}$$

E is saturated if sat(E)=E.

Definition 4.2.4. *[5] Any indecomposable and saturated set of $\mathbb{R}^N$ is called simple.*

In general a decomposition in M-connected components does not lead directly to a canonical decomposition of the boundary. The following definitions of "exterior" and "Jordan boundary" help to achieve this goal.

Definition 4.2.5 (Exterior [5]). *If $E \subseteq \mathbb{R}^N$ has finite perimeter and $|E| < \infty$, we call exterior of E the unique (mod $\mathcal{H}^N$) M-component of $\mathbb{R}^N \setminus E$ with infinite measure. The exterior of E will be denoted by ext(E).*

Definition 4.2.6 (Jordan boundary [5]). *A set J is a Jordan boundary if there is a simple set E such that $J = \partial^M E$ (mod $\mathcal{H}^{N-1}$).*

Let $\partial^M E$ describe a collection of "external Jordan boundaries" J_i^+ and "internal Jordan boundaries" J_i^- satisfying some inclusion boundaries. Let J_∞ be a Jordan curve whose interior is $\mathbb{R}^N$, and let J_0 be a Jordan curve whose interior is empty. Also set $\mathcal{H}^{N-1}(J_\infty) = \mathcal{H}^{N-1}(J_0) = 0$. Denote by $\mathcal{S}$ the extended class of Jordan boundaries including J_∞ and J_0. The following theorems shows the decomposability of $\partial^M E$ in Jordan boundaries.

Theorem 4.2.7 (Decomposition of $\partial^M E$ in Jordan boundaries [5]). *Let $E \subseteq \mathbb{R}^N$ be a set of finite perimeter. Then there is a unique decomposition of $\partial^M E$ into Jordan boundaries $\{J_i^+, J_k^- : i, k \in \mathbb{N}\} \subseteq \mathcal{S}$ such that*

(i) Given $int(J_i^+)$, $int(J_k^+)$, $i \neq k$, they are either disjoint or one is contained in the other; given $int(J_i^-)$, $int(J_k^-)$, $i \neq k$, they are either disjoint or one is contained in the other. Each $int(J_i^-)$ is contained in one of the $int(J_k^+)$.

(ii) $P(E) = \sum_i \mathcal{H}^{N-1}(J_i^+) + \sum_k \mathcal{H}^{N-1}(J_k^-)$.

(iii) If $int(J_i^+) \subseteq int(J_j^+)$, $i \neq j$, then there is some Jordan boundary J_k^- such that $int(J_i^+) \subseteq int(J_k^-) \subseteq int(J_j^+)$. Similarly, if $int(J_i^-) \subseteq int(J_j^-)$, $i \neq j$, then there is some Jordan boundary J_k^+ such that $int(J_i^-) \subseteq int(J_k^+) \subseteq int(J_j^-)$.

(iv) Setting $L_j = \{i : int(J_i^-) \subseteq int(J_j^+)\}$, the sets $Y_j = int(J_j^+) \setminus \cup_{i\in L_j} int(J_i^-)$ are pairwise disjoint, indecomposable and $E = \cup_j Y_j$.

Theorem 4.2.8. *[5] Let $\{J_i^+, J_k^- : i, k \in \mathbb{N}\} \subset \mathcal{S}$ satisfy the following conditions*

(i) Given $int(J_i^+)$, $int(J_k^+)$, $i \neq k$, they are either disjoint or one is contained in the other; given $int(J_i^-)$, $int(J_k^-)$, $i \neq k$, they are either disjoint or one is contained in the other. Each $int(J_i^-)$ is contained in one of the $int(J_k^+)$.

(ii) Each two different Jordan boundaries of the system $\{J_i^+, J_k^- : i, k \geq 0\}$ *are disjoint (mod* $\mathcal{H}^{N-1}$*).*

(iii) If $int(J_i^+) \subseteq int(J_j^+)$*,* $i \neq j$*, then there is some Jordan boundary* J_k^- *such that* $int(J_i^+) \subseteq int(J_k^-) \subseteq int(J_j^+)$*. Similarly, if* $int(J_i^-) \subseteq int(J_j^-)$*,* $i \neq j$*, then there is some Jordan boundary* J_k^+ *such that* $int(J_i^-) \subseteq int(J_k^+) \subseteq int(J_j^-)$*.*

(iv) $\sum_i P(J_i^+) + \sum_k P(J_k^-) < \infty$*.*

Let $E = \cup_j Y_j$*, where*

$$Y_j := int(J_j^+) \setminus \bigcup_{i \in L_j} int(J_i^-). \tag{4.10}$$

Then E *is a set of finite perimeter and* $\partial^M E = \cup_i J_i^+ \cup \cup_k J_k^-$ *(mod* $\mathcal{H}^{N-1}$*).*

Theorem 4.2.7 and Theorem 4.2.8, whose proofs are presented in [5], provide a way to study the boundaries of the decomposed domains of an image. Occlusions in an image can also be handled by the decomposition of the boundaries into Jordan boundaries.

The representation of the boundary of a set E of finite perimeter by a family of nested Jordan boundaries can be obtained by the family of saturations and holes of the M-connected components of E. But this representation has a drawback: It is not invariant under complementation. Another drawback of the representation is the absence of a natural order.

There exists a family of nested boundaries which is invariant under completion. This family is given by $\partial^M\{u \leq k\}$ where k is even for the external boundaries and odd for the internal ones and $u : \mathbb{R}^N \to \mathbb{N}$ is the BV_{loc} function characterized by the following theorem.

Theorem 4.2.9 ([5])**.** *Let* $E \subseteq \mathbb{R}^N$ *be a set of finite perimeter. Then there exists a unique map* $u \in BV_{loc}(\mathbb{R}^N, \mathbb{N})$ *such that*

(i) $u = \chi_E$ *mod 2 and all sets* $\{u \leq k\}$ *are indecomposable;*

(ii) $|Du| = \mathcal{H}^{N-1} \llcorner \partial^M E$*;*

(iii) $u = \chi_E$ *in the* M*-connected component of* E *or* $\mathbb{R}^N \setminus E$ *with infinite measure.*

Definition 4.2.10 (Topographic function [5])**.** *We call the function* u *given by the previous theorem the topographic function of* E*, and denote it by* u_E*. We also call the sets*

$$\partial^M\{u \leq 2k\}, \quad \partial^M\{u \leq 2k+1\} \qquad k \in \mathbb{N} \tag{4.11}$$

respectively the external and the internal boundaries of E*.*

It can be seen from this definition that $u_E + 1 = u_{\mathbb{R}^N \setminus E}$ whenever $|E| < \infty$, and therefore complementation maps internal (external) boundaries into external (internal) boundaries, which means it is invariant under complementation.

Note that all the detailed developments and proofs of the theorems are contained in [5]. Please refer to [5] for more details and explanations. These properties are listed here as background information. They will be utilized to explain the proposed methods in Section 4.3.

4.2.2 Image Representation using the Tree of Shapes

This subsection introduces the practical applications of techniques using the theories mentioned in the previous subsection.

The tree of shapes [14] [21] [24] [61] [91] [99] represents images based on the techniques of contrast-invariant mathematical morphologies [60] [97]. This method is based on the mathematical theories shown in the previous subsection [5], especially Theorem 4.2.1, Theorem 4.2.7 and Theorem 4.2.9. Theorem 4.2.1 shows the decomposability of a set with finite perimeter, which is helpful for image segmentation. Theorem 4.2.7 demonstrates that the boundaries of the sets of finite perimeters can be utilized in image processing since these boundaries are connections of Jordan curves. Theorem 4.2.9 builds the relation between the BV space and the sets of finite perimeters.

The representation of an image using a tree of shapes utilizes the inferior or the superior of a level line to represent an object, and takes the boundary of the inferior area as the shape of the object. Therefore, only closed shapes are generated. This representation also provides a tree structure to represent the spatial relationship for the objects in an image.

For a gray image $u : \Omega \to \mathbb{R}$ with $\Omega \subset R^2$, the upper level set χ_λ of value λ and the lower level set χ^μ of value μ are defined in [21] as

$$\chi_\lambda = \{x \in R^2, u(x) \geq \lambda\} \tag{4.12}$$

$$\chi^\mu = \{x \in R^2, u(x) \leq \mu\} \tag{4.13}$$

The above definitions have several advantages. First, they represent regions instead of curves in an image, which provide a way to handle the contents inside the regions. Second, they are invariant to the contrast changes in an image, which may be caused by the change of lighting [60] [97]. Third, closed boundaries are acquired for each upper level set or lower level set, which can be utilized for shape matching of the regions. In comparison, the level lines defined by $l_\lambda = \{x \in R^2, u(x) = \lambda\}$ usually generate open curves rather than closed curves in real images. For an image modeled as a function of bounded variation, the regions represented by level sets are connected components. Fourth, their representations are complete for images, which means that the family of the upper level sets χ_λ (or the family of the lower level sets χ^μ) is sufficient to reconstruct the image [60] [97] because of the following relationship [21]:

$$u(x) = \sup\{\lambda | \mathrm{x} \in \chi_\lambda\} = \inf\{\mu | \mathrm{x} \in \chi^\mu\} \tag{4.14}$$

Note that the geometrical inclusion holds for the level sets. The family of upper (lower) level sets is decreasing (increasing) because [21]

$$\lambda \leq \mu \Rightarrow \chi_\lambda \supset \chi_\mu \quad \text{and} \quad \chi^\lambda \subset \chi^\mu \tag{4.15}$$

The nesting of level sets provides an inclusion tree for an image. The inclusion tree from the family of upper level sets and the tree from the family of lower level sets, however, can be different if the connected components are directly utilized. The concept of "shape" is introduced to generate a unique inclusion tree for an image. A shape is defined as the connected components of a level set and the holes inside them. Fig. 4.2 shows an example of the tree of shapes generated for a piecewise-constant image. A tree of shapes shown in Fig. 4.2(h) is constructed for the image in Fig. 4.2(a). The whole image acts as the root of the tree, which locates at the top level. The shapes in the same level are spatially disjoint

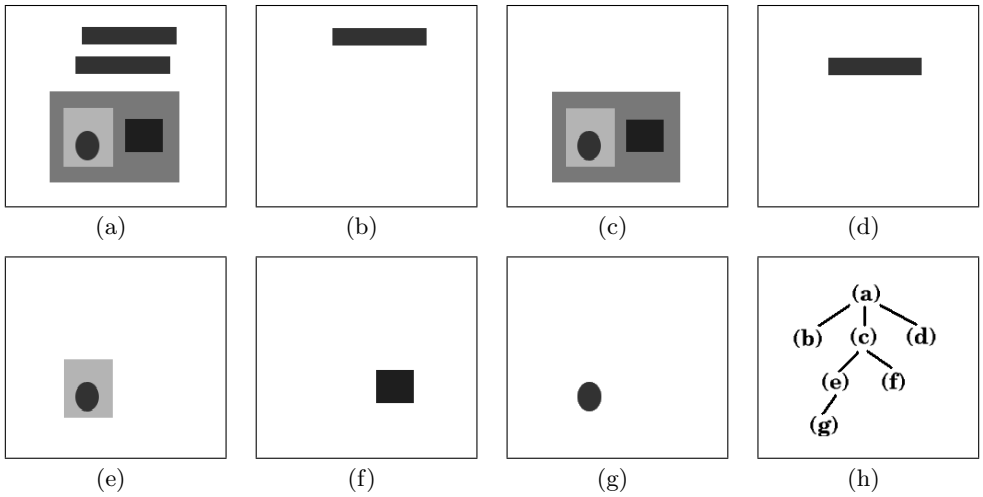


Figure 4.2: Illustration of the tree of shapes. (a) The original image, which locates at the root of the inclusion tree. (b)(c)(d) Shapes in the first layer of the tree of shapes. (e)(f) Shapes in the second layer of the tree of shapes. (g) Shapes in the third layer of the tree of shapes. (h) The Structure of the tree of shapes.

in the image. The shapes in the lower level are spatially included in the shapes in the next higher level. The tree of shapes, therefore, provides a natural way to represent the spatial relationships between the shapes in the image. Please refer to [97] [99] for the efficient construction of the tree of shapes.

4.2.3 Color and Geometry in Mathematical Morphologies

The total order (or lexicographical order) proposed in [33] [63] is utilized in the proposed method below to compare the color vectors in a color image. Let $\mathbf{C_1} = (Y_1, S_1, H_1)$ and $\mathbf{C_2} = (Y_2, S_2, H_2)$ represent the color vectors in two pixels in a color image. The meanings of Y_1, S_1, and H_1 differ in different color models. The total order provides a way to compare these color vectors. It is defined as follows

$$\mathbf{C_1} < \mathbf{C_2} \quad \text{if} \quad \begin{cases} Y_1 < Y_2 & \text{or} \\ Y_1 = Y_2 & \text{and} \quad S_1 < S_2 \quad \text{or} \\ Y_1 = Y_2 & \text{and} \quad S_1 = S_2 \quad \text{and} \quad H_1 < H_2 \end{cases} \tag{4.16}$$

The definition in (4.16) bears some similarity with the concept of conditional expectations shown in Section 4.2.3 when Y_1 represents the gray level in the image.

Y_1 actually represents the gray level in the color image in the proposed method. An improved HLS ($IHLS$) space [64] is utilized for the color model. The color in every pixel is represented with three channels (Y, S, H), which corresponds to the gray level, saturation and hue respectively. The $IHLS$ space, compared to other spaces such as HLS and HSV, has the property of a "well-behaved" saturation coordinate. The $IHLS$ space always has a small numerical value for near-achromatic colors, and is completely independent of the brightness function. For a pixel with color (R, G, B) in the RGB space, and the corresponding pixel with color (Y, S, H), the transformation from the RGB space to the IHLS space is

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{4.17}$$

(a) (b) (c)

Figure 4.3: Illustration of tree of shapes for a color Image. (a) A color image of interests. (b) A gray image from (a). (c) A tree of shapes built for (a).

$$S = \max(R, G, B) - \min(R, G, B) \tag{4.18}$$

$$H = \begin{cases} 360^o - H' & \text{if} \quad B > G \\ H' & \text{otherwise} \end{cases} \tag{4.19}$$

where

$$H' = \frac{R - 0.5G - 0.5B}{(R^2 + G^2 + B^2 - RG - RB - BG)^{\frac{1}{2}}} \tag{4.20}$$

The proposed method uses the transformation and the total order in (4.16) to extract the shapes and build the tree of shapes for color images. The inverse transformation from the IHLS space to the RGB space is not utilized here. Please refer to [64] for more details.

Fig. 4.3 illustrates the construction of a tree of shapes for a color image. Fig. 4.3(a) displays a color image of interest. The color image is synthesized so that its gray version, shown in Fig. 4.3(b), contains no shape information. Using the IHLS color space (4.17-4.20) and the total order (4.16), a tree of shapes is built for the color image, shown in Fig. 4.3(c).
.

4.2.4 Planar Curve Matching

Planar curve matching is introduced in this subsection. The method in [91] defines the shape of a curve as a conjunction of shape elements and further defines the shape elements as any local, contrast invariant and affine invariant part of the curve. These definitions are oriented to provide invariance to noise, affine distortion, contrast changes, occlusion, and background.

The shape matching between two images are designed as the following steps:

(i) Extraction of the level lines for each image. The level set representations (4.12)(4.13) are utilized here for the extraction. The level lines is defined as the boundaries of the connected components as shown before.

(ii) Affine filtering of the extracted level lines at several scales. This step is applied to smooth the curves using affine curvature deformation to reduce the effects of noise.

(iii) Local encoding of pieces of level lines after affine normalization. Both local encoding and affine normalization are designed for local shape recognition methods. This step will help to deal with occlusions in real applications.

(iv) Comparison of the vectors of features of the images.

The performance of curve matching between two curves is calculated after affine filtering, curve normalization and local encoding. Suppose C_1 and C_2 are two curves for matching, and S_1 and S_2 are pieces from C_1 and C_2 respectively. The performance to take S_1 and S_2 as the matching between C_1 and C_2 is

$$Score = \frac{l_1 \times l_2}{L_1 \times L_2} \tag{4.21}$$

where $l_1 = arclength(S_1)$, $l_2 = arclength(S_2)$, $L_1 = arclength(C_1)$ and $L_2 = arclength(C_2)$. The maximum score over all possible matching pieces is taken as the matching between the curves C_1 and C_2.

Please refer to [91] for details of the mathematical definitions and the implementation issues.

4.2.5 Scale Space Theory for the Tree of Shapes

Scale space analysis [82] [117] [158], has been popular in image processing. Scale space theory for the tree of shapes is introduced in [98]. Since the image to be processed is represented by a tree of shapes, it is intuitive that the scale space of a tree of shapes is related to the size of its shapes. This is the case, and this result bears similarity to the grain filter in [25]. Let T_t represent the multi-scale operator with parameter t. Then the application of the operator T_t to an image is equivalent to removing all the shapes whose areas are less than t from the inclusion tree and then adjusting the tree accordingly. The formal mathematical formulation is shown as follows.

Let $|B|$ denote the area of a set B. Denote $\phi(B)$ the smallest simply connected set containing B. Let $\mathcal{B}_t$ be the family of closed connected sets B where $\phi(B)$ contains the origin $\mathbf{O}$ and that if $\mathbf{O}$ is in a hole H of B, then $|\phi(H)| \leq t$. Then the multi-scale operator

$$T_t u(x) = \sup_{B \in \mathcal{B}_t} \inf_{y \in x+B} u(y) \tag{4.22}$$

Properties of the multi-scale operator are provided in [98] .

The scale space theory of the tree of shapes has not been utilized in the proposed method yet. But it provides the potential to extend the proposed method using the multi-scale operator. The extension is expected to be more stable than the proposed method with careful selection of scales. This is a topic for future work.

4.3 Preferential Image Segmentation

A preferential image segmentation algorithm is proposed in this section. The proposed algorithm is illustrated using an example for clarification. Suppose we need to preferentially segment objects similar to the one indicated in Fig. 4.4(a) from the cluttered image shown in Fig. 4.4(b). As illustrated in Section 4.2.2, the tree of shapes provides a natural way to represent the spatial relationship of the shapes in an image. The tree of shapes, therefore, is a good candidate tool to utilize the intensity information in the prior image for preferential segmentation. A naturally intuitive idea is to construct the trees of shapes for both images and then find the node in the tree of image in Fig. 4.4(b) whose properties are similar to the node in Fig. 4.2(c).

The tree of shapes, however, usually generate large amounts of shapes, especially for complicated images. For example, 3832 shapes, as shown in Fig. 4.4(c), are contained in the tree of shapes for the image in Fig. 4.4(b). Note that every closed red curve corresponds to the boundaries of a shape in the tree of shapes in Fig. 4.4(c).

The boundaries of shapes provide valuable information for the similarities between shapes. However, comparison of all the boundaries corresponding to shapes in the image to be segmented would be computationally intense. It is necessary to narrow down the candidate shapes from the image to be segmented for each shape from the prior image before their boundaries can be compared. The intensity information contained in the shapes provides good measures for the similarities between shapes in real images. The intensity information by means of the tree of shapes (e.g. Fig. 4.2(h)) includes the following features

(i) The number of objects Nd contained directly in the shape, which corresponds to the number of direct children of the shape in the tree. $Nd = 2$ for the indicated shape in Fig. 4.4(a).

(ii) The total number of objects Nt contained in the shape, which corresponds to the total number of children below the shape in the tree. $Nt = 3$ for the indicated shape in Fig. 4.4(a).

(iii) The relative area change A between the shape and its direct children. Suppose the area of the shape is S, and the areas of its direct children are S_i, where $1 \leq i \leq N_d$, the relative area change is then defined as

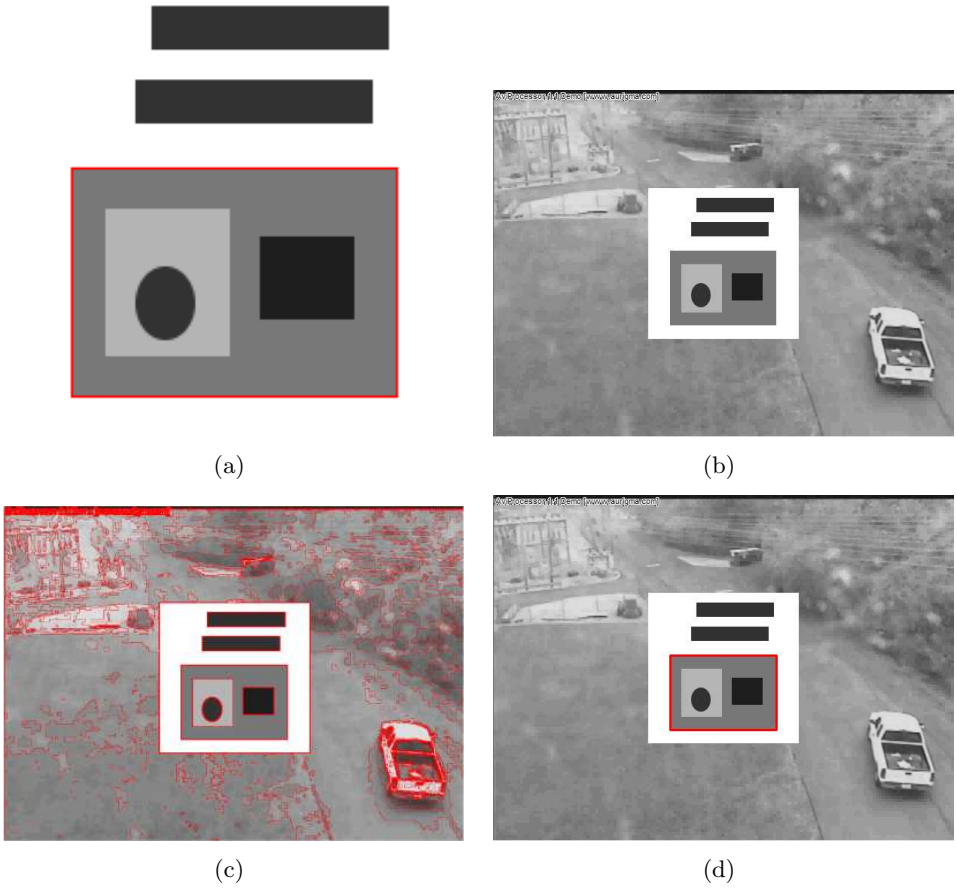
$$A = \prod_{i=1}^{N_d} \frac{S_i}{S} \tag{4.23}$$

(iv) The rough similarities R of the boundaries of the shapes, which can be represented as the ratio of the circumferences C squared to the area S of the boundaries of the shapes, i.e., $R = C^2/S$.

These features for two shapes should be very close if they match. Exact matching may not be achieved because of the differences between shapes, the effects of noise, and computational errors. Thresholds should be set for coarse matching to affect a rapid exclusion of most candidate shapes. For example, set the threshold $Nd_{thres} = 1$, $Ratio_{thres} = 0.3$. A shape is considered to roughly match the prior match if $abs(Nd_{prior} - Nd) \leq Nd_{thres}$, $abs(Nt/Nt_{prior} - 1) < Ratio_{thres}$, $abs(A/A_{prior} - 1) < Ratio_{thres}$, $abs(R/R_{prior} - 1) < Ratio_{thres}$. These thresholds may be adjusted for different applications.

The number of candidate shapes decreases substantially by means of the intensity features extracted from the tree of shapes. In the case of Fig. 4.4(b), the feature Nd decreases the number of candidate shapes from 3832 to 977; the feature Nt decreases the number from 977 to 83; the feature A decreases the number from 83 to 1; the feature R retains this candidate. The candidate shape left over matches the prior shape, as shown in Fig. 4.4(d). The process takes 1.70 seconds.

In most cases, however, the candidate shapes will decrease from thousands to tens, but not 1. Curve matching as introduced in Section 4.2.4 is then performed on the boundaries of the resulting candidate shapes and the prior shapes. The candidate shape which best matches the prior's curve is taken as the preferential segmentation result.



(a) (b) (c) (d)

Figure 4.4: Illustration of of preferential image segmentation using the tree of shapes. (a) The object of interest in the prior image. (b) The image to be segmented. (c) A simplified tree of shapes of the image in (b). (d) Segmentation results using the proposed preferential image segmentation. Image size: 460 * 612. CPU time: 1.70s.

In summary, the proposed method models an image with a function of bounded variation and utilizes a tree of shapes for image representation. This method encodes the prior information for preferential segmentation as a tree of shapes. The method is invariant to translation, rotation and scale transformations because both feature extraction and boundary matching are invariant to these transformations.

The proposed method bears similarities with template matching. Both methods utilize prior information to locate objects from images. But these methods are different in various aspects. First, the proposed method models an image as a function of bounded variation, while template matching usually models an image as a continuous function. Second, the proposed method utilizes level sets for image representation and represents spatial relationship by a tree structure, while template matching represents an image as pixels by means of sampling theorem. Third, the proposed method handles invariance to rotation, scale and translation automatically, while template matching needs to handle these invariances explicitly. Fourth, the proposed method is easy to extend to segmentation by parts, while template matching is hard for occlusions. More importantly, the proposed method segments a most similar object from an image, which combines segmentation and recognition. A template matching method, on the other hand, only recognizes the location, orientation and scale information of a known object. There is no segmentation in template matching. Experimental results for this comparison are shown in the next section.

4.4 Experimental Results

Experimental results from the proposed method are provided in this section. The proposed method is implemented using the MEGAWAVE package (http://www.cmla.ens-cachan.fr). The program runs on a computer which has two Intel(R) Pentium(R) 3.2GHz CPUs, 1G bytes RAM, and runs the Red Hat Enterprise Linux operating system. The CPU times given in this chapter are the sums of system CPU times and user CPU times. The system CPU time is usually very small, typically 0.01-0.08 seconds.

Two implementation issues need to be clarified here. First, the representative shapes are manually selected from the inclusion tree of the prior image. The "llview" function of the MEGAWAVE package is utilized to visualize the inclusion tree for shape selection. This is acceptable because the operation only need to be performed once. Automatic selection is the focus of current research. Second, interpolation methods are utilized in the construction of the inclusion tree for high precision. This may generate many shapes whose areas are very close (e.g. 95-98%). These shapes may correspond to only one object in the image. A sensitivity measure is utilized to handle this issue. The shape will not be considered as an object until the ratio of its area to that of its precedents falls below a certain threshold (e.g. 80%).

Fig. 4.5 shows the preferential image segmentation of a car from an image with cluttered background. For the prior image Fig. 4.5(a), the representative shape is selected as shown in Fig. 4.5(b). A very similar shape is selected from the image Fig. 4.5(c), as shown in Fig. 4.5(d). Curve matching is not necessary here because only one candidate is left over. The process takes 1.53 seconds. For comparison, the results in Fig. 4.5(d) are also achieved by means of template matching, whose implementation by the author takes 192.57 seconds, which is very time consuming compared to the proposed method (126x). Fig. 4.5(f) shows the intermediate segmentation results using the Chan-Vese curve evolution method [27] without using any prior information for the initialization shown in Fig. 4.5(e). It can

be seen that the curve evolution method segments the brighter area of the image, which includes parts of the car, but it can not segment the whole car. The curve evolution method is computationally intense, requiring almost 12 minutes in this case. Furthermore, the post processing for the segmentation results is challenging because of the cluttered background. The curve evolution methods based on prior information may provide better segmentation results, but it can be expected that they would be also computationally intense. Thorough comparisons are a subject of future research.

Fig. 4.6 shows the performance of the proposed method for a MRI brain image and a car logo. The prior shape enclosed in the red line in Fig. 4.6(a) is used to segment the image in Fig. 4.6(b), which also contains the segmentation results. Only 0.61s was required for the segmentation of an image of size 218 * 282. Similarly, the prior shape in Fig. 4.6(c) is utilized for the segmentation of Fig. 4.6(d), which also contains the segmentation results. This took 4.98s for the segmentation of an image of size 600 * 450.

The rotation invariance of the proposed method is illustrated in Fig. 4.7. The shape shown in Fig. 4.7(a) is utilized as the prior for preferential image segmentation in Fig. 4.7(b). The candidate shapes resulting from intensity features are shown in Fig. 4.7(c) Both shapes are similar to the one in Fig. 4.7(a). Fig. 4.7(d) shows the final result, which agrees with the ground truth.

The proposed method is also invariant to scale change, as shown in Fig. 4.8. The object shown in Fig. 4.8(a) (size: 147 * 113) is extracted from Fig. 4.8(d), and is then scaled up to Fig. 4.8(b) (size: 195 * 150), which is utilized as the prior for preferential segmentation in Fig. 4.8(d). Fig. 4.8(c) shows the prior shape for Fig. 4.8(b). Fig. 4.8(d) shows the segmentation results. The object is found using a scaled prior image, which demonstrates scale invariance.

Contrast invariance and intensity invariance of the proposed method are demonstrated in Fig. 4.9. Fig. 4.9(a) shows the object extracted from the image to be segmented. The image contrast is increased, as shown in Fig. 4.9(b). The object is successfully segmented from Fig. 4.9(d) using the selected prior in Fig. 4.9(c). The intensity levels are increased for the image in Fig. 4.9(e). The results in Fig. 4.9(g) show that the object has been successfully segmented by means of the prior shape in Fig. 4.9(f).

The results shown in Fig. 4.10 show that the proposed method works in the presence of noise. The prior shape in Fig. 4.10(a) are utilized for preferential segmentation. Fig. 4.10(b) shows the segmentation results for a noisy image. Fig. 4.10(c) displays the results for the same noisy image after diffusion. The proposed method works in both cases, though the shapes of the segmented regions are slightly different from the prior object's shape due to the effects of noise. It can also be expected that the method may fail in the case of heavy noise because noise affects the intensity features and curve matching.

A collection of 26 images is utilized to test the performances of the proposed method. The objects in 16 images are successfully segmented based on the prior image in Fig. 4.8(c). Fig. 4.11 shows the segmentation results of four exemplar images. The results show that the proposed method works for complicated images with shadows.

The proposed method is applied to face images in Fig. 4.12. Twenty face images (image_0001.jpg - image_0020.jpg from the Caltech face image database, which is available at http://www.vision.caltech.edu/archive.html) are tested. The face in image_0018.jpg is extracted for the prior image. Three images with different backgrounds are segmented successfully, as shown in Fig. 4.12(b), Fig. 4.12(c), and Fig. 4.12(d). Expression changes in the face, such as winking eyes, and opening and closing of eyes or mouth, may cause sufficient

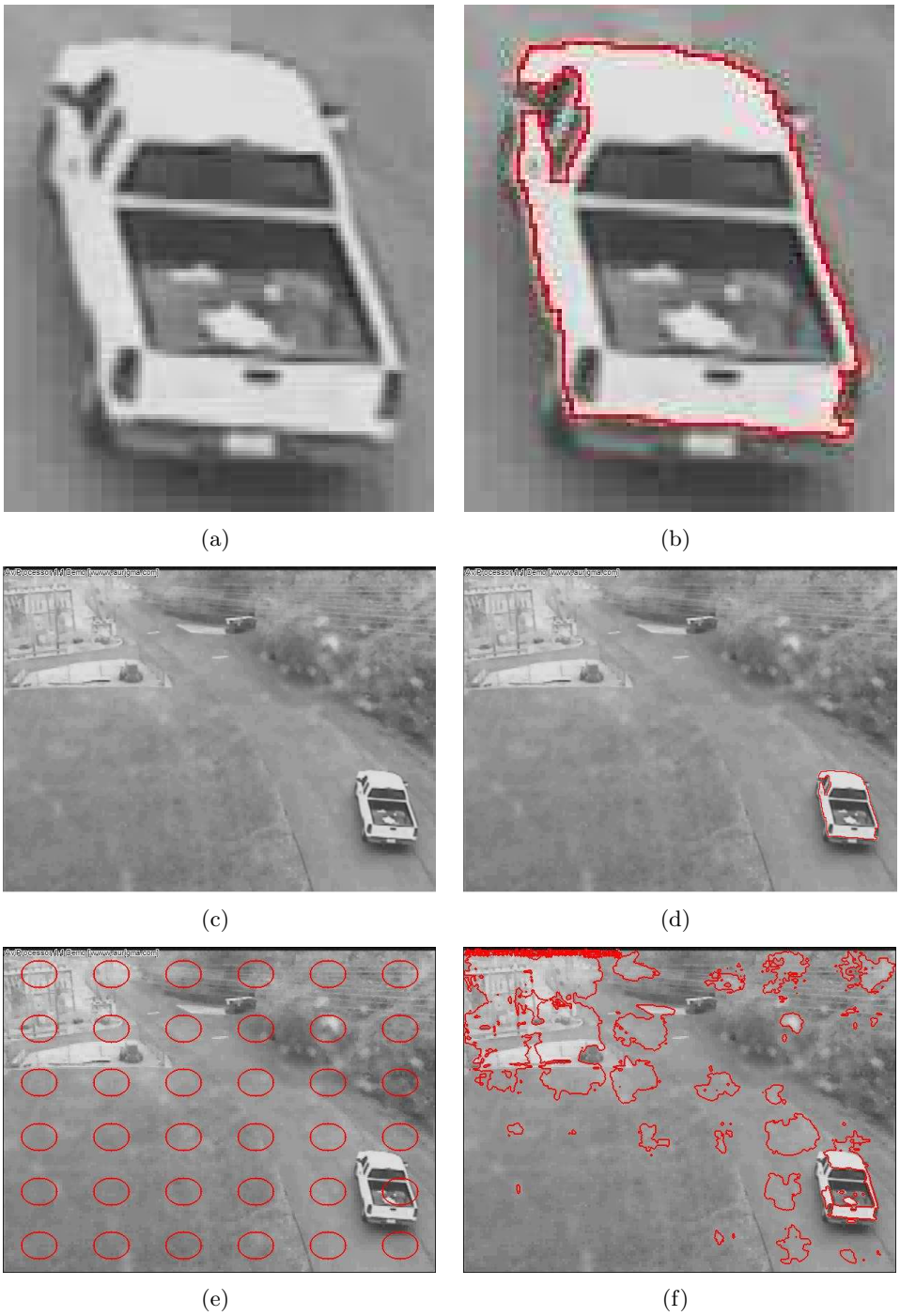


Figure 4.5: Preferential image segmentation of a car using the tree of shapes. (a) The prior image (image size: 460 * 612). (b) The object of interest in the prior image. (c) The image to be segmented. (d) Segmentation results using the proposed preferential image segmentation. The region enclosed by the red line corresponds to the segmentation result similar to (b). CPU time: 1.53s. (e) The initialized curves for the Chan-Vese curve evolution. (f) The intermediate segmentation result using curve evolution after 800 iterations. CPU time: 712s.

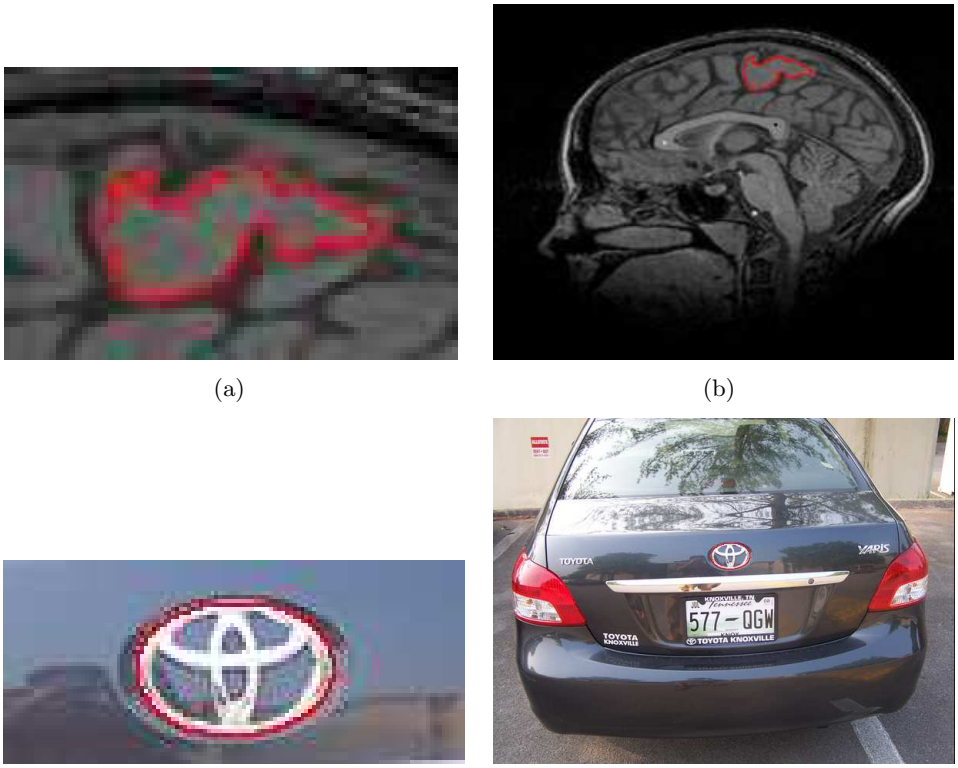
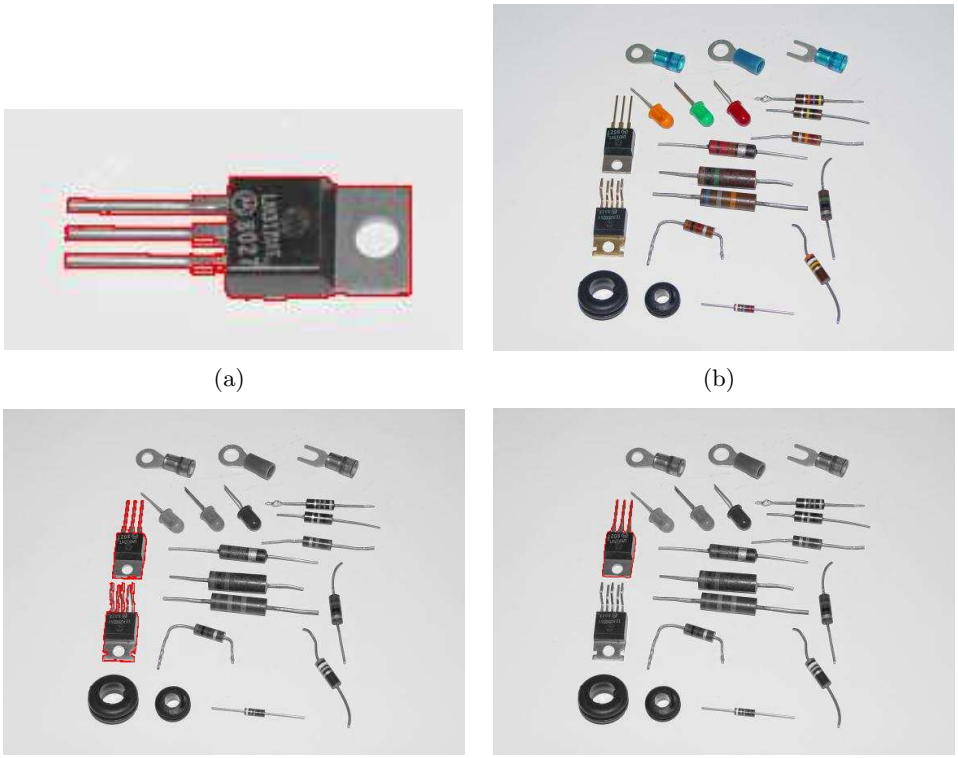


Figure 4.6: More examples for preferential image segmentation. (a) The prior image for a medical image. (b) Segmentation results of using the prior shape from (a). Image size: 218 * 282. CPU time: 0.61s. (c) The prior image for a car logo. (d) Segmentation results using the prior shape from (c). Image size: 600 * 450. CPU time: 4.98s. The regions enclosed by the red lines corresponds to the segmentation results.



(a) (b)

(c) (d)

Figure 4.7: Illustration of rotation invariance of the proposed method. (a) The prior image with object of interests. (b) The image to be segmented. Image size: 426 * 568. (c) Candidate shapes from the intensity prior. (d) Segmentation result. The region enclosed by the red line corresponds to the segmentation result. CPU time: 2.74s.

(a) (b) (c) (d)

Figure 4.8: Illustration of scale invariance of the proposed method. (a) The object extracted from the image to be segmented. Size: 147 * 113. (b) The rescaled image from (a). Size: 195 * 150. (c) Prior shape selected from (b). (d) Segmentation result. The region enclosed by the red line corresponds to the segmentation result. CPU time: 2.73s.

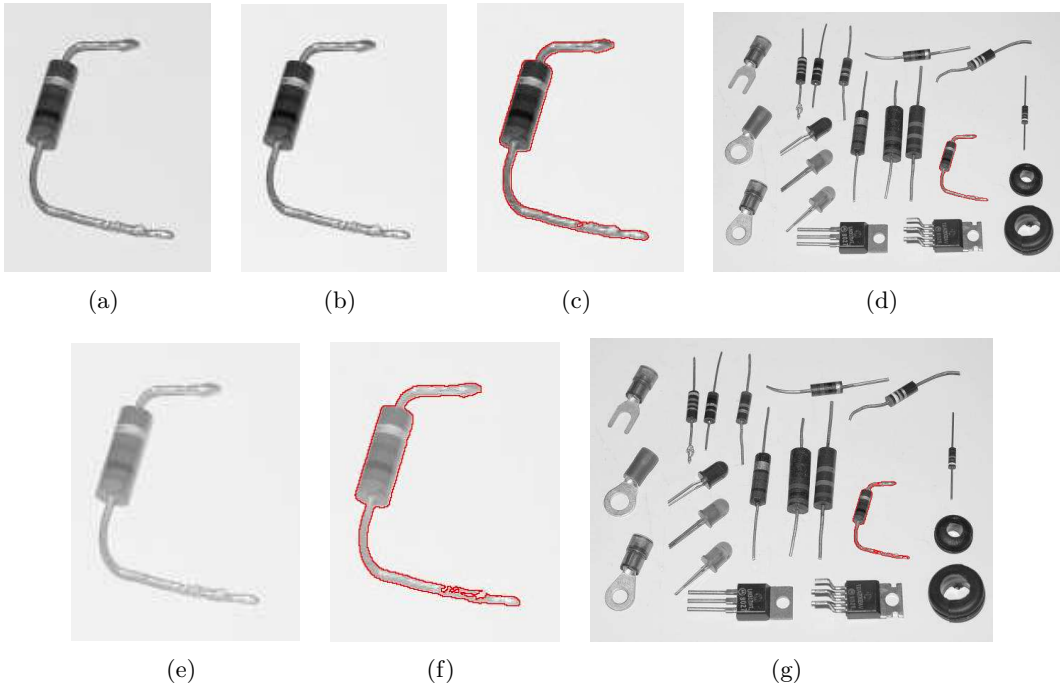


Figure 4.9: Illustration of scale invariance of the proposed method. (a) The object extracted from the image to be segmented. (b) The image with higher contrast. (c) Prior shape selected from (b). (d) Segmentation result using (c). Image size: 426 * 568. CPU time: 1.61s. (e) The image with higher brightness. (f) Prior shape selected from (e). (g) Segmentation result using (f). Image size: 426 * 568. CPU time: 2.57s.

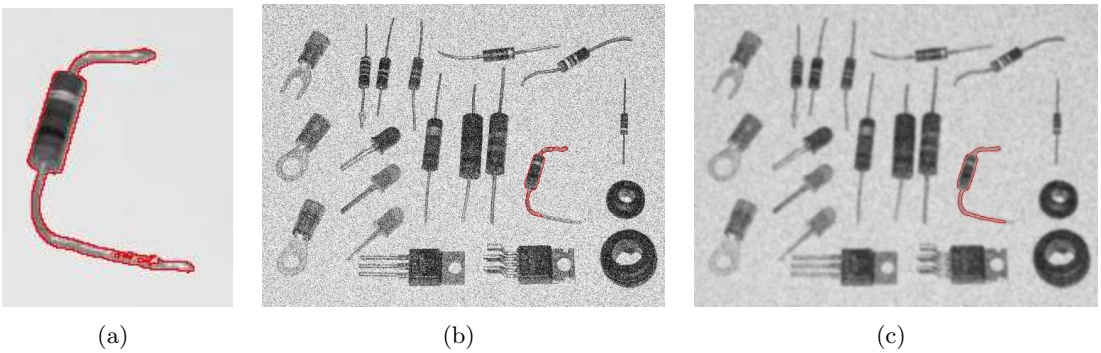
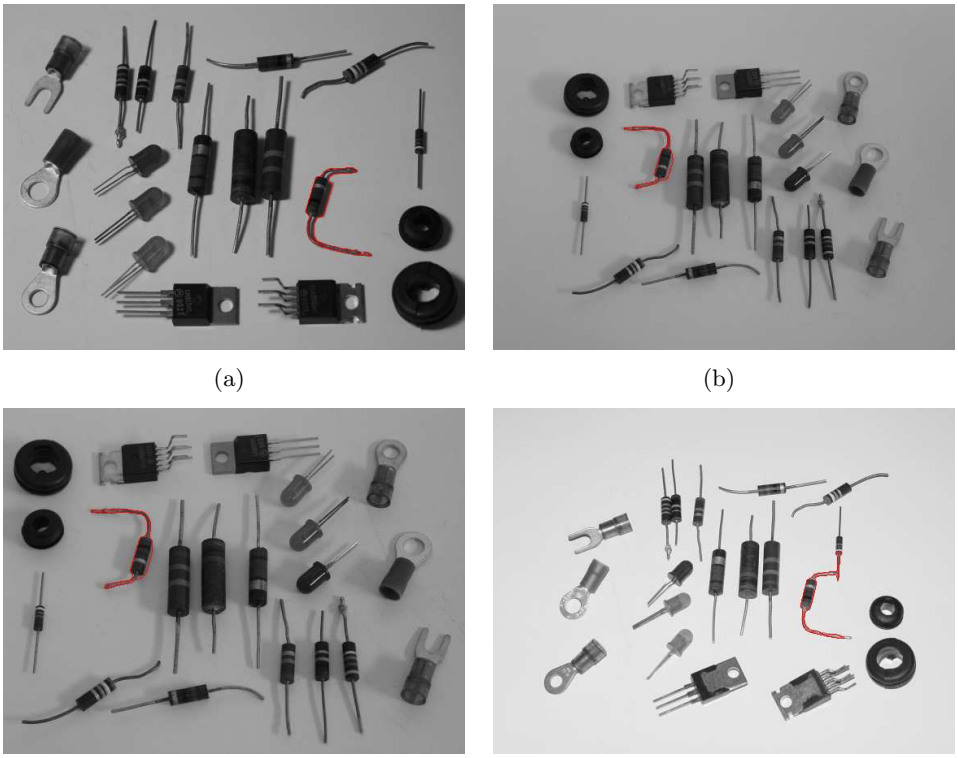


Figure 4.10: The proposed method works to the effects of noise. (a) The prior shape. (b) Segmentation results for a noisy image (Size: 568 * 426). CPU time: 4.91s. (c) Segmentation results for the noisy image in (b) after diffusion. CPU time: 3.68s.



(a) (b) (c) (d)

Figure 4.11: Preferential image segmentation for a collection of images. (a) Segmentation results. CPU time: 3.62s. (b) Segmentation results. CPU time: 3.09s. (c) Segmentation results. CPU time: 4.26s. (d) Segmentation results. CPU time: 2.03s. Image size: 460 * 612.

changes in the face images to introduce difficulties for preferential image segmentation using the proposed method. Segmentation by parts may be a good solution to this issue.

4.5 Performance Evaluation of Preferential Image Segmentation using the Tree of Shapes

The previous sections show the theoretical background, algorithm details and experimental results of the proposed preferential image segmentation method. The results demonstrate that the proposed method is efficient, invariant to contrast changes and similarity transformations, and applicable to various applications. A performance evaluation of the proposed method using a large set of natural images is presented in this section. The results here will give a better indication of the performance of the proposed methods in real applications.

This section presents the procedure used for performance evaluation and the corresponding results. The experimental results are also provided in this section. An introduction to the image dataset is described in Section 4.5.1. The way to select prior shapes from the training images is illustrated in Section 4.5.2. This is followed by the details of the experimental results in Section 4.5.3.

4.5.1 Introduction to the Image Dataset

The image dataset contains a training set of 26 images and a test set of 95 images. These images are taken under different conditions of lighting, backgrounds, shadows, displacements, and rotations, and examples of close contact between objects and strong occlusions. Every training image contains 20 objects whose prior information is known. The number of objects varies in the test images. The total number of objects in the 95 test images is 520. Fig. 4.13(a) shows the indices of the prior objects in a training image. Fig. 4.13(b) - Fig. 4.13(d) shows examples of the training images with good lighting and clear background. Fig. 4.13(e) and Fig. 4.13(f) are examples of the training images with light shadows. Fig. 4.13(g) and Fig. 4.13(h) are examples of the training images with strong shadows. Occlusions and rotations can be seen in Fig. 4.13(c) and Fig. 4.13(d).

The test images are complicated by occlusion, textured background, strong shadow and close contact. Fig. 4.14 shows examples of the images to be segmented in the test set. Note that the training images have no textured backgrounds, which introduces more challenges to the segmentation task. The test images are categorized into several classes: clear background, cluttered (or textured) background, light shadow, strong shadow, close contact, occlusion, separated. The first two classes represent the background information; the middle two classes show the lighting conditions; the last three attributes represent the spatial relationships between objects. These categories may overlap. Among the 95 test images, 47 test images have clear backgrounds, e.g. Fig. 4.14 (a)(d)(g)(h); 33 test images have cluttered background, e.g. Fig. 4.14 (b)(c)(e)(f); 7 test images have light shadows, e.g. Fig. 4.14 (f); 34 test images have strong shadows, e.g. Fig. 4.14 (g)(h); 30 test images have close contacts among their objects, e.g. Fig. 4.14 (e); 41 images have occlusions, e.g. Fig. 4.14 (b)(c)(d)(f)(g); 24 test images have spatially separated objects. These categories will be analyzed separately to determine the influences of these factors on the segmentation results.

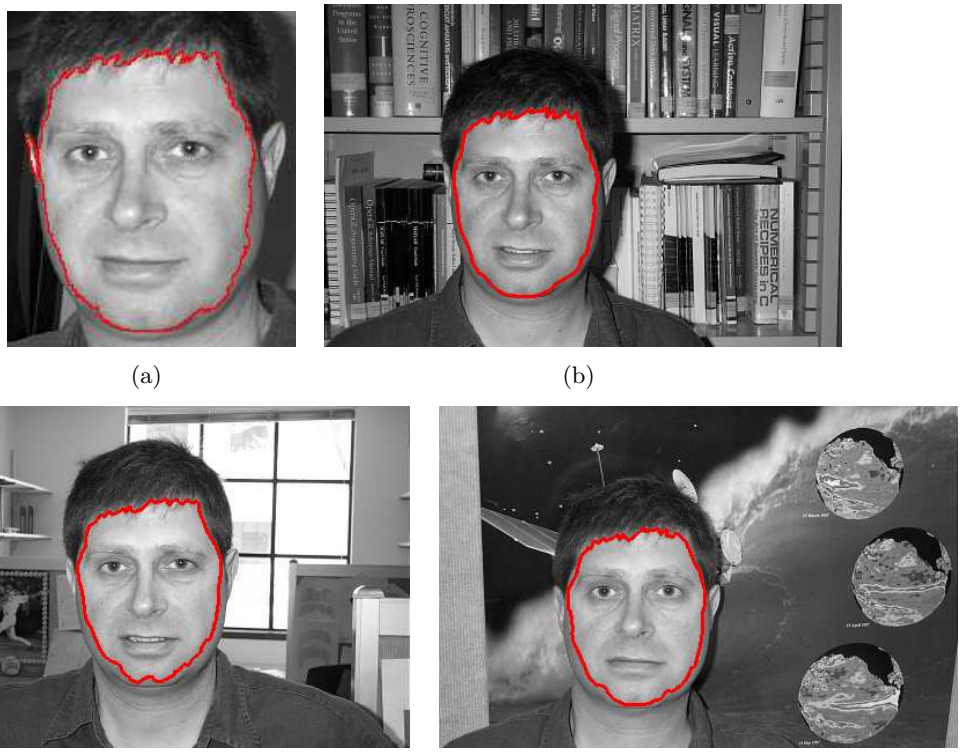


Figure 4.12: Preferential image segmentation of face images (from http://www.vision.caltech.edu/archive.html) using the Tree of Shapes. (a) The prior image with selected shape. (b) Segmentation results (image size: 296 * 448). CPU time: 2.43s. (c) Segmentation results for another face image (size: 296 * 448). CPU time: 2.26s. (d) Segmentation results (size: 296 * 448). CPU time: 2.59s.

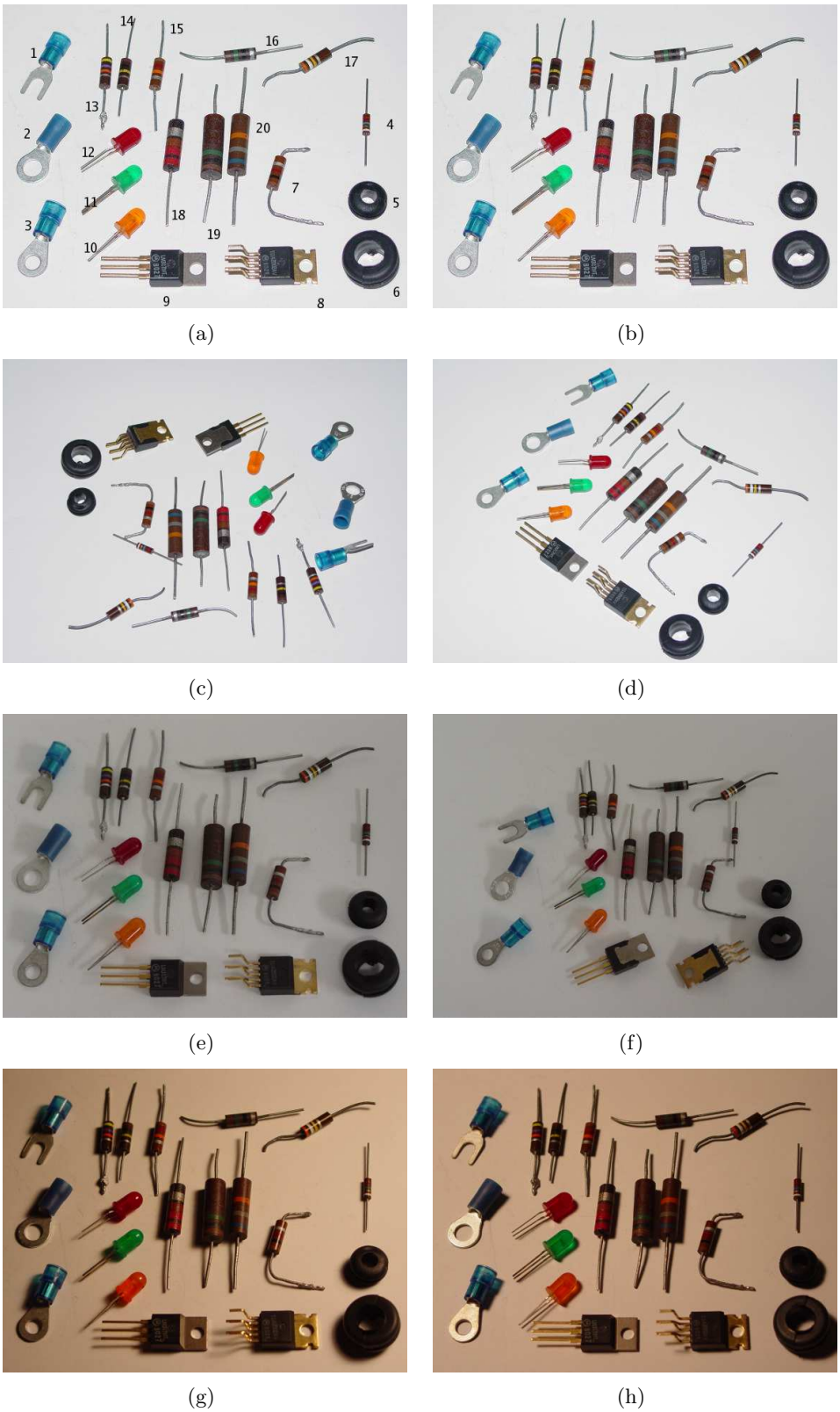


Figure 4.13: Examples of prior images in the training set. (a) The indices of objects in the prior image. (b)-(h) Examples of prior images with different background, lighting, shadows, poses and occlusions.

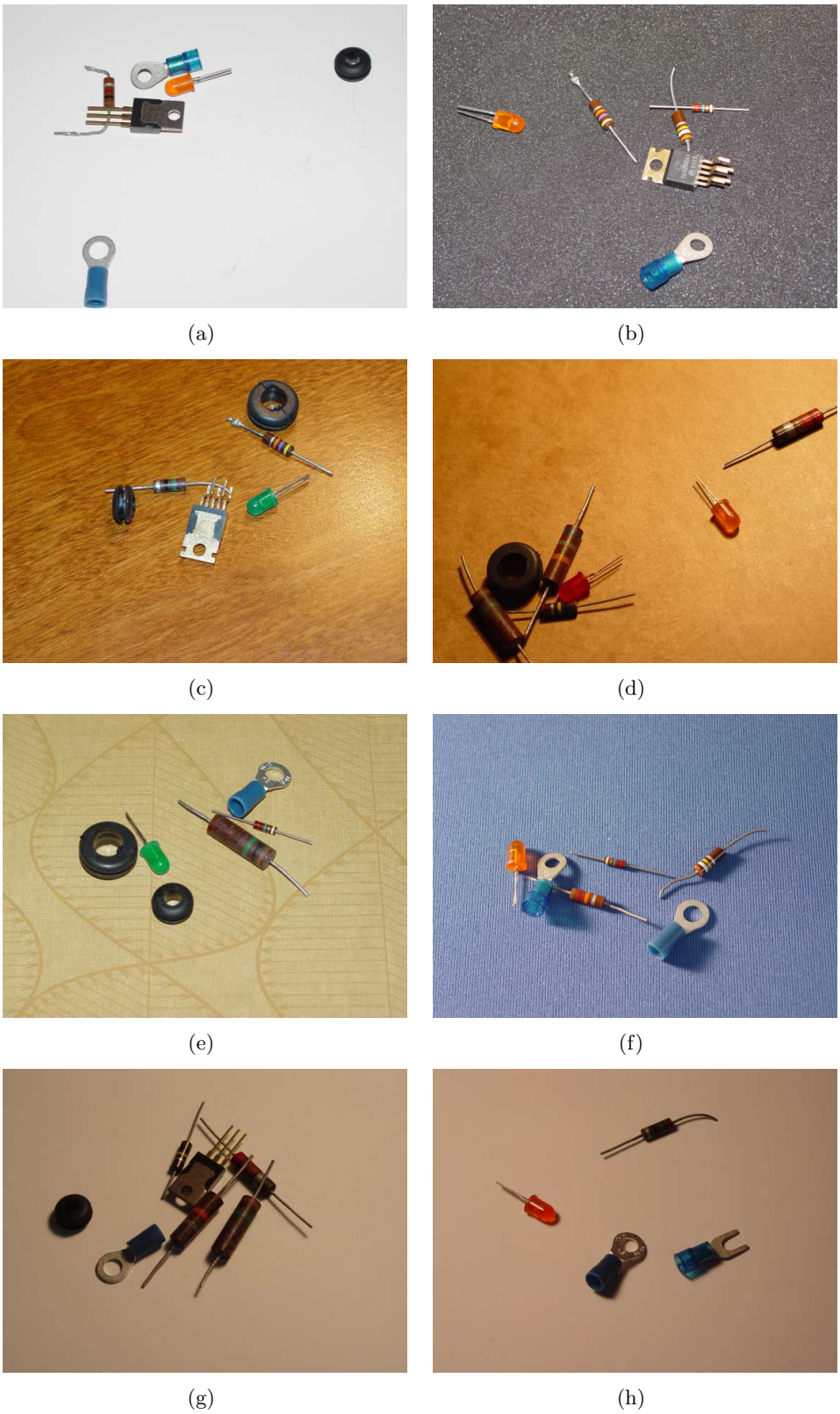
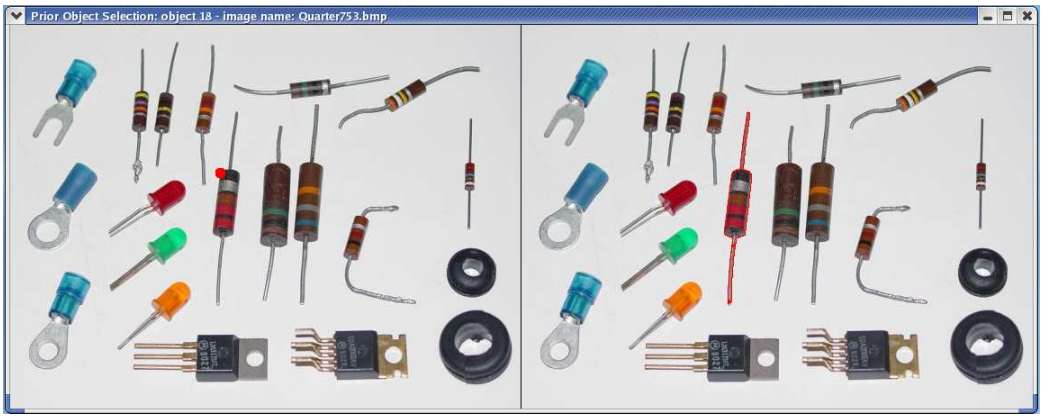


Figure 4.14: Examples of images to be segmented in the test Set. (a)-(h) Examples of test images with different background, lighting, shadows, poses and occlusions.



(a)

Figure 4.15: Illustration of shape prior selection in training images.

4.5.2 Prior Selection

The experiments are designed to utilize the prior information provided in the training set to segment the objects in the test set. The priors in the images of the training set are first manually segmented. An interface for the prior selection has been developed with the help of the CImg library [147]. The program will preferentially select the shape which contains the point left-clicked by the mouse and which has at least one direct child. The selected shape will be removed by right clicking the mouse if the selected object is not satisfactory. Fig. 4.15 shows the procedure used to select object 18 from the training image Quarter753.bmp. The user clicks the pixel under the red cursor, located in the left part of the training image, and the boundary of the selected object is shown in the right part.

Fig. 4.16 shows several examples of prior objects selected from the training images. Fig. 4.16(a) and Fig. 4.16(b) shows prior objects selected from images with no shadows or occlusions. Fig. 4.16(c)-Fig. 4.16(f) shows prior objects selected in the presence of light or strong shadows, which degrades the prior selection process. Parts of the prior objects are selected when the whole objects are difficult to select because of the presence of shadow or occlusion, as shown in Fig. 4.16(g) and Fig. 4.16(h).

The manual selection procedure is offline and identifies the priors from the training images. The prior information utilized in the proposed method contains only the number of direct children N_d, the total number of children N_t, the relative area change A, the rough similarities R, and the coordinates of the pixels on the boundaries of the object. This information is stored in a sequential text file on the hard drive. The text file will be read into memory when the test images are processed.

As an example, the extracted information for the shape prior selected in Fig. 4.15 (object 18 in Quarter753.bmp) is displayed below.

```
1 1 52 1 0 116.199 525 362717 2 0 252 205 ...
```

Each of the numbers is stored as a line in the record. These numbers are displayed as a line for the convenience of visualization. The first number (1) represents the type of the level set. Its value is set to 1 if the selected shape prior is a upper level set; otherwise 0. The second number (1) represents the direct children of the selected shape prior. The third number (52) stores the number of total children. The forth number stands for the ratio of

the relative area change, defined in Eq. 4.23. The fifth number (0) represents the depth of the current shape. The sixth number (116.199) represents the boundary-squared-to-area ratio of the selected shape. The numbers left over are utilized for boundary matching. The seventh number (525) shows the size of the boundary for the selected shape. The eighth number (362717) represents the maximum size available for the boundary storage. The ninth number (2) is the dimension of the boundary pixels, which is usually 2. The tenth number in the file represents the data for program use. The eleventh number and the twelfth number represents the coordinates of a pixel in the boundary. The rest of the file shows the coordinates of all other pixels contained in the boundary of the selected shape.

4.5.3 Experimental Results

In the prior selection stage, one object is selected from the training images at a time. Thus $26 \times 20 = 520$ prior objects are obtained. The prior information contained in every object is utilized to segment each of the 95 test images. Thus $26 \times 20 \times 95 = 49,400$ experiments are performed to evaluate the performance of the proposed methods. In the detection of object 1 from the test image Quarter779.bmp, for example, all 26 priors for object 1 are utilized one at a time, and the one that matches best is recorded as the segmentation result.

Human Visual Evaluation. The most reliable way to evaluate the experimental results is by human eyes. 456 out of 520 objects (87.7%) in the 95 test images have been correctly segmented according to this evaluation procedure. Less than 20 objects were falsely segmented for those 1380 objects that do not exist in test images using a human evaluation criteria. This is a false alarm rate of about 1.4% using that evaluation.

However, results from human evaluations are subjective. Users may get different results from their evaluations because different criteria are utilized. A loose criterion is utilized here for an illustration. Fig. 4.17 shows the groups of objects which are taken as inseparable by the visual evaluation criteria. For example, if object 1 is not shown in an image, but the program segments object 2 (or object 3) as the result, it is taken as false alarm. The accuracy may increase to above 90%, but at the same time the false alarm rate rises 24% (331 out of 1380). This shows that the false alarm rate of the proposed algorithm lies between 1.4% and 24%, depending on the criteria. Similarly, the accuracy lies between 87.7% and 100%.

Fig. 4.18 and Fig. 4.19 show several segmentation results. The left column in Fig. 4.18 shows the prior object, while the right column shows the segmentation result. Fig. 4.18(b) shows an example where the proposed method works very well for good lighting with no occlusions. Fig. 4.18(d) shows an example where an object can be segmented by parts in cluttered background. Fig. 4.18(f) shows that an object can be segmented when it is placed in different poses. Fig. 4.18(h) shows that the proposed method may also work well in the presence of strong shadows.

Fig. 4.19 shows several segmentation results from challenging problems. The left column in Fig. 4.19 shows the prior object, while the right column shows the segmentation results. The results in Fig. 4.19(b) show that close contacts between objects may introduce difficulties for preferential segmentation. Fig. 4.19(b) shows the performance of the method in textured background and strong shadows. Because of occlusion, object 16 can not be segmented in Fig. 4.19(f). Fig. 4.19(h) shows that a cluttered background can cause preferential segmentation to fail.

The previous discussion shows the necessity for automatic and objective evaluation. Human visual performance evaluation is subjective rather than objective. It also requires

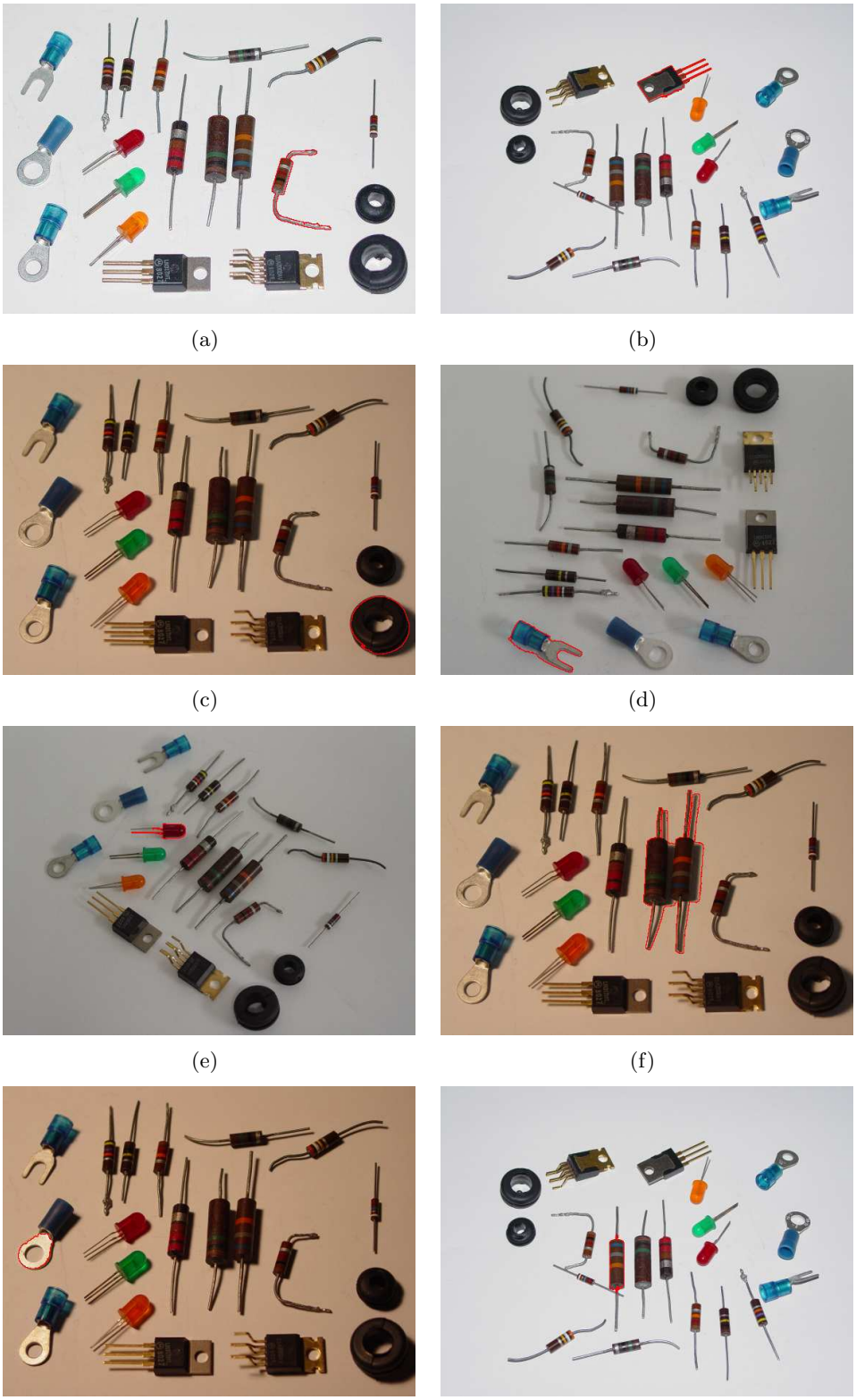
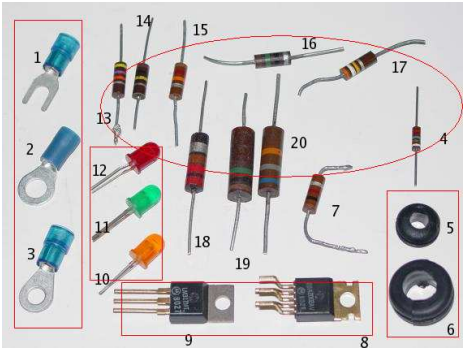


Figure 4.16: Examples of prior objects selected from the training images. (a)-(h) Examples of prior objects selected from the training images images with different background, lighting, shadows, poses and occlusions.



(a)

Figure 4.17: A Loose criteria for false alarm evaluation. (a) Groups of objects which are taken as inseparable by the visual evaluation criteria.

human intervention and is therefore not automatic, which is a big disadvantage for large image datasets or autonomous operations. Automatic performance evaluation needs a quantitative evaluation of the match quality between the prior object (e.g. Fig. 4.19(a)) and the corresponding segmented object (e.g. Fig. 4.19(b)). An automatic performance evaluation using the location center is presented below.

Automatic Performance Evaluation using the location center The locations of segmented objects provides a way to evaluate the performance of the proposed method. If the center of a segmented object is close to the center of visual observation, then preferential segmentation may be considered as successful. Otherwise it may be considered a failure. This method does not give a thorough evaluation because different segments may have the same center, and that the content and boundary information are not utilized. But it provides a way to evaluate the locations of the segmented objects.

In the implementation, the locations of the objects are first visually determined first. The locations of the segments are calculated by the program, and the Euclidean distances between the corresponding locations are calculated. The minimum of the distances between all priors is taken as the final result. If an object does not exist in an image, or the program does not segment anything for an object, the location of that object is taken as (0,0). Therefore, this evaluation method produces a small false alarm (false positive) rate and a high accuracy (true positive) rate relative to other evaluation strategies.

Fig. 4.20 shows the figures of the true positive rate (Fig. 4.20(a)), the false positive rate (Fig. 4.20(b)), the false negative rate (Fig. 4.20(c)) and the true negative rate (Fig. 4.20(d)) with respect to the distance threshold in pixels, as well as the ROC curve (with the true positive rate as the vertical axis and the false positive rate as the horizontal axis) (Fig. 4.20(e)). One issue about the ROC curve is that the maximum false positive rate is less than 20%. A trade-off between detection capability and error rates can be achieved by choice of detection threshold. If a 91% detection rate is desired, a threshold of 206.47 is chosen, yielding a false positive rate of 8.7% and a failure-to-detect rate of 9.0%.

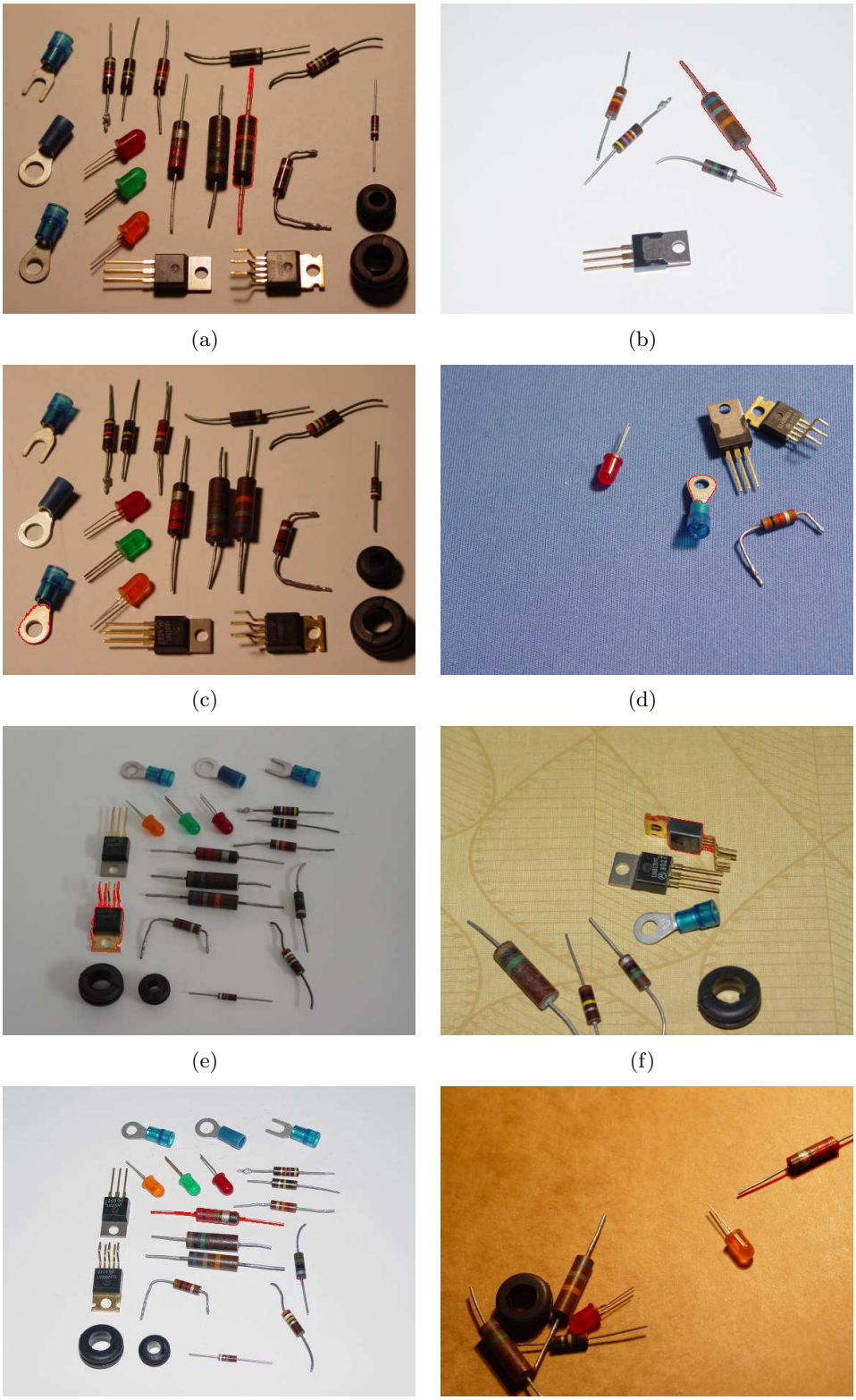


Figure 4.18: Examples of segmentation results. (a)(c)(e)(g) Prior objects in the training images. (b)(d)(f)(h) Segmentation results of the test images using the priors in (a)(c)(e)(g) respectively.

Figure 4.19: Examples of Segmentation Results. (a)(c)(e)(g) Prior objects in the training images. (b)(d)(f)(h) Segmentation results of the test images using the priors in (a)(c)(e)(g) respectively.

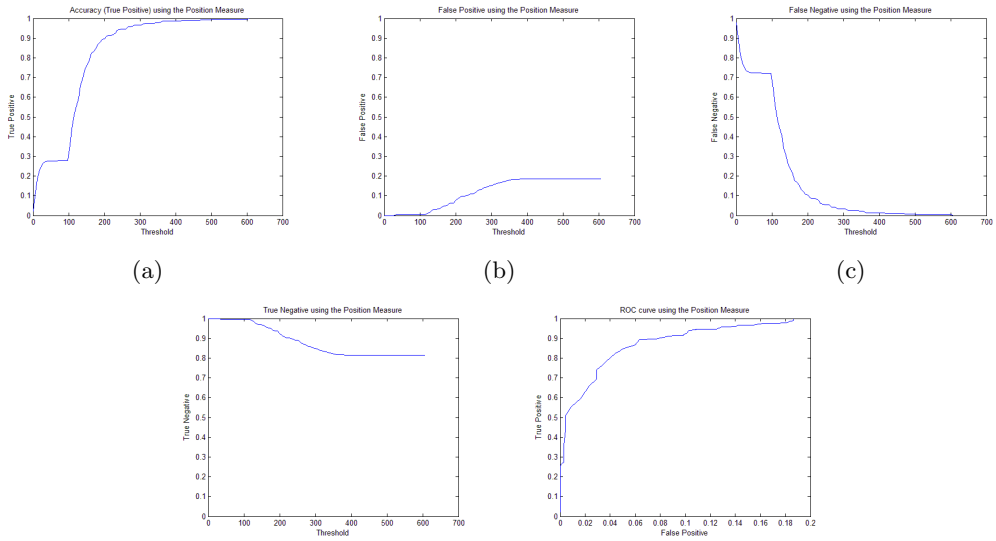


Figure 4.20: Performance evaluation with respect to the distance threshold. (a) Accuracy rate (true positive rate). (b) False alarm rate (false positive). (c) False negative rate (failure to detect). (d) True negative rate. (e) ROC curve for the evaluation.

4.6 Preferential Image Segmentation User Interface using an Image Database and the Web

A user interface for preferential image segmentation is presented in the section using an image database and web interfaces. The user interface provides a prototype for future applications, which corresponds to the system proposed in Scheme 1.2.2 in Chapter 1. The interface uses a web browser such as Firefox and can be used to illustrate the performance of the proposed algorithm. It can also be extended to support applications such as medical image processing, facial image processing, and more complicated video tracking and alarm systems.

The interface utilizes **MySQL** [45] to manage the image database, which contains both the prior images and the images to be segmented. These images are stored as files on a hard drive and can be accessed by their names and locations stored in the database. Their names and contents can also be stored in the database using the **BLOB** type data by means of the **PerlMagick** (http://www.imagemagick.org/script/perl-magick.php) Package. The preferential image segmentation algorithm is implemented using **C++**. The **CGI** package [45] is utilized to develop the web interface, which allows the user to control the segmentation process. The **Perl** programming language [151] (http://www.perl.org) is utilized to coordinate the image database (**MySQL**), the segmentation algorithm (**C++**) and the web interface (**CGI**). More details are be provided below with illustrations.

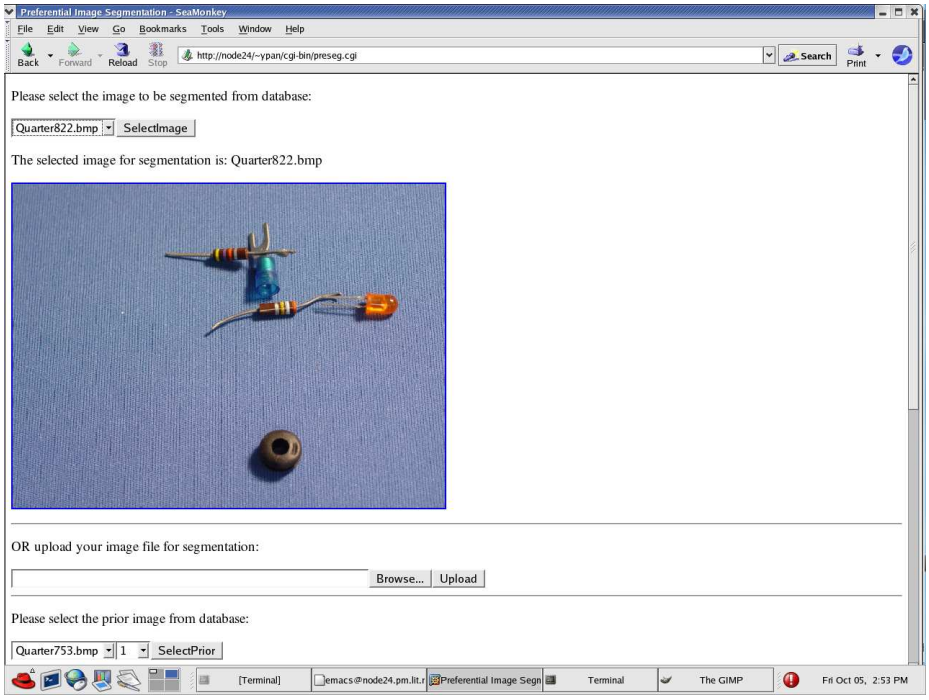
The Apache web server is used and receives a request when a user types the URL (e.g., http://node24/ ypan/cgi-bin/preseg.cgi) in the web browser. A server script connects to the MySQL image database and retrieves the names of the prior images and the test images stored in the database. This is done via the **Perl DBI** interface [45]. The retrieved image names are placed in pop-up boxes in the web page for selection. The same procedure is used for the prior images. Pressing the button "SelectImage" (or "SelectPrior") sends a request the Apache web server to show the selected test image (or the selected prior image). The image containing the segmentation result will be displayed automatically if it exists. Otherwise the button "Preferential Image Segmentation" on the bottom of the web page can be pressed to perform the segmentation and display the results. Fig. 4.21 and Fig. 4.22 illustrate this process.

The user can also upload an image to the web server and have it segmented. Fig. 4.23 shows the upload process, in which the image is uploaded and its information is stored in the image database.

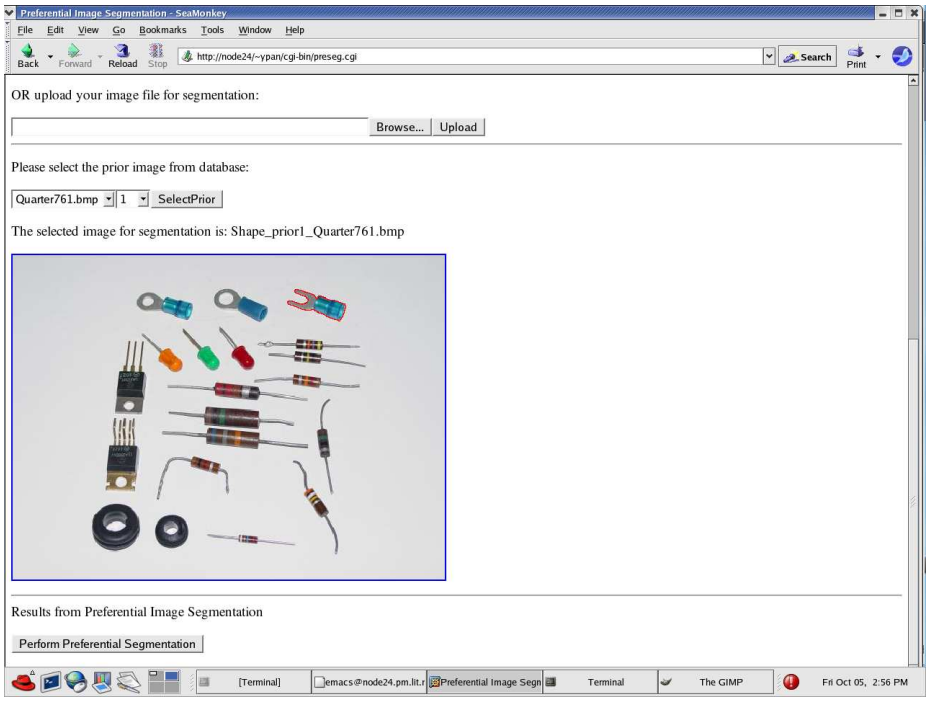
4.7 Summary and Future Research

A novel preferential image segmentation method is proposed in this chapter. The method utilizes both the intensity and shape prior information by means of the tree of shapes and boundary matching. It is invariant to contrast change and similarity transformations such as scale, rotation and translation. Experimental results show that the proposed method is able to preferentially segment objects from complicated cases. Future research on segmentation methods will be focused on the multiscale analysis of the proposed method and the topic of segmentation by parts.

The proposed method is also applied to large image databases to examine its performance. In the performanceevaluation procedure, the detection rate is above 87.7% and the false alarm rate lies between 1.4% and 24% by human observation. The results are shown

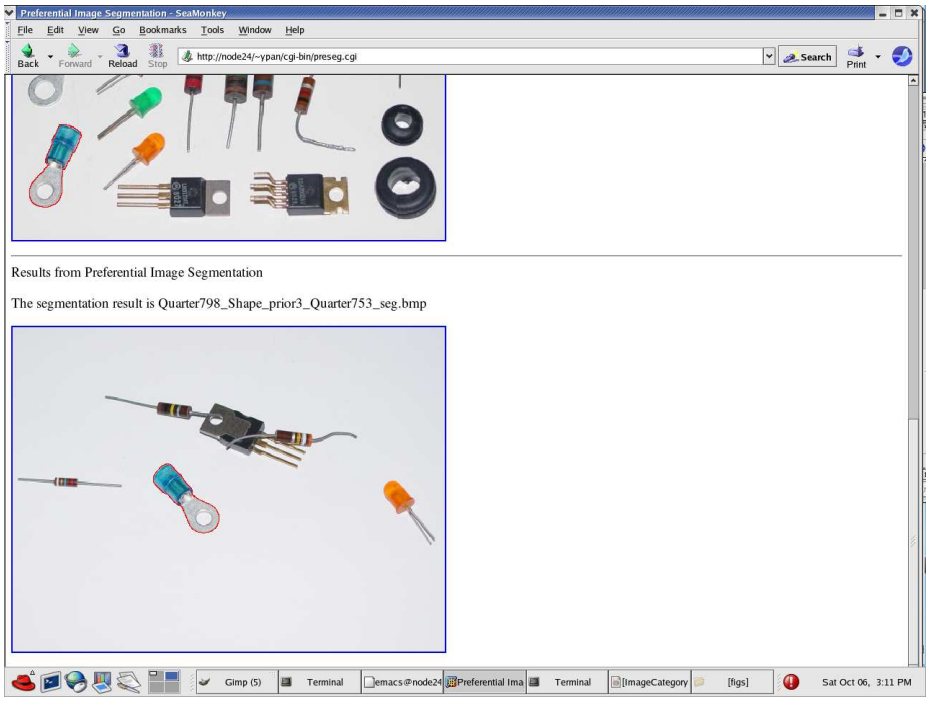


(a)



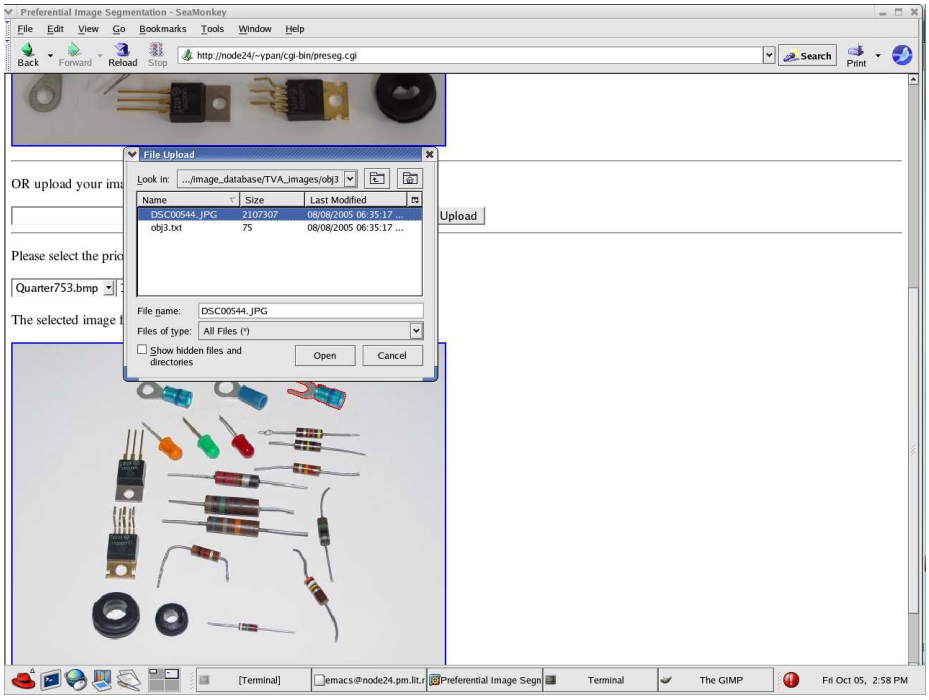
(b)

Figure 4.21: Select and visualize images from a image database. (a) A test image is selected and visualized. (b) A prior image is selected and visualized.

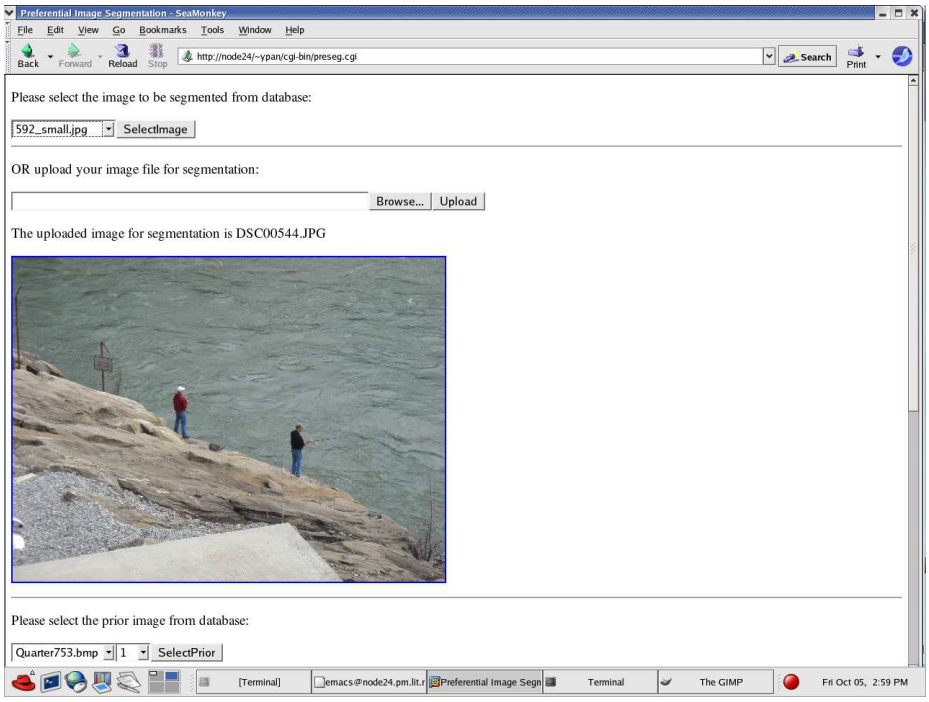


(a)

Figure 4.22: Retrieve and visualize segmentation results. (a) A segmentation result is retrieved and displayed.



(a)



(b)

Figure 4.23: Upload, visualize and store an image to be segmented. (a) Upload an image from a local computer. (b) Visualize the uploaded image and store its information to a image database.

to be very promising for images with different lightings, backgrounds and occlusions. A web interface combined with MySQL, Perl and CGI Package is also provided for the illustration of the algorithm. The web interface provides the prototype for real applications. Future work on performance evaluation can focus on the selection (or design) of measures. Some results in the area of content based image retrieval may be useful for this topic.

Chapter 5

Image Segmentation by Matching Distributions

This chapter presents two approaches to segment objects by matching distributions. Section 5.1 presents a probabilistic curve evolution model using particle filters for image segmentation [110]. Section 5.2 shows the way to preferentially segment objects by matching histogram-based distributions using the Kullback-Leibler distance.

5.1 Probabilistic Model for Curve Evolution using Particle Filtering

5.1.1 Introduction

Active contour models are among the most state-of-art image segmentation methods. In these models, a simple closed curve C initialized in an image evolves according to attributes of the image and the curve itself. Image segmentation is achieved by stopping the evolving curve at boundaries between objects.

Numerous active contour models have been proposed. In [78]- [115], gradient information and/or region information from the image are utilized. Although the ways to use the image information are different, these models are all geometric flows.

Image segmentation can also be viewed as a specific tracking problem: tracking of object boundaries using curve evolution from an initial position. In this sense, Kalman snakes [144]- [118] are alternatives for image segmentation. However, Kalman snakes and their extensions are only applicable for systems with Gaussian noise. This assumption fails in curve evolution, which has nonlinear dynamics with unknown noise.

Another approach is based on particle systems, which were first introduced for shape modeling [120] [141] and then extended to image segmentation [69]. In particle systems, particles move under the influence of constructed force fields. The particles can be generated, modified and terminated. The particles can evolve to form a shape of interest [120] [141] or stop at object boundaries for segmentation [69].

A new probabilistic active contour model [110] is formulated, in which curve evolution is a state prediction and estimation problem for a nonlinear dynamical system. Particle filters [9] [43], which are applicable to nonlinear systems with non-Gaussian noise, are used to solve the state prediction and estimation problem in a Bayesian framework. The states in the model are defined as the properties of a narrow band of image pixels around the

evolving curve. By means of level set methods, the proposed model automatically deals with topological changes in image segmentation. Both gradient and region information are utilized to guide the curve's evolution.

The section is organized as follows. The geometric curve evolution model is introduced in section 5.1.2, and a probabilistic active contour model is proposed in section 5.1.3. In section 5.1.4, some background information on particle filters is provided. Several implementation problems are discussed in section 5.1.5. Section 5.1.6 provides an analysis of experimental results, followed by a brief conclusion in section 5.1.7.

5.1.2 The Geometric Active Contour Model

Geometric active contour models are briefly introduced to make complete this section. Some information about geometric active contour models introduced in chapter 3 is presented here to make this chapter self-contained. Basic concepts and their implementations are discussed first, followed by a more complicated model, which makes use of both gradient and region information for curve evolution.

In the geometric curve evolution model, the evolving curve C_t at time t is usually implicitly represented by the zero level set of a function of two dimensions ψ,

$$C_t = \{(x, y) : \psi(x, y, t) = 0\} \tag{5.1}$$

This formulation allows topological changes as the function ψ evolves, and this is a great advantage over parametric curve evolution methods.

A simple curve evolution model evolves the initial curve at a constant speed along its normal direction, making the curve expand or shrink. The curvature is utilized at each point on the curve to make the evolving curve smooth at all times. This model can be described as

$$\psi_t = (\upsilon + \epsilon\kappa)|\nabla\psi| \tag{5.2}$$

where υ is the constant speed of evolution, ϵ is a positive constant, and $\kappa = div(\nabla\psi/|\nabla\psi|)$ is the curvature, where div represents the divergence operator.

The curve evolution model of (5.2) is equivalent to the minimization of the length of the evolving curve, i.e.

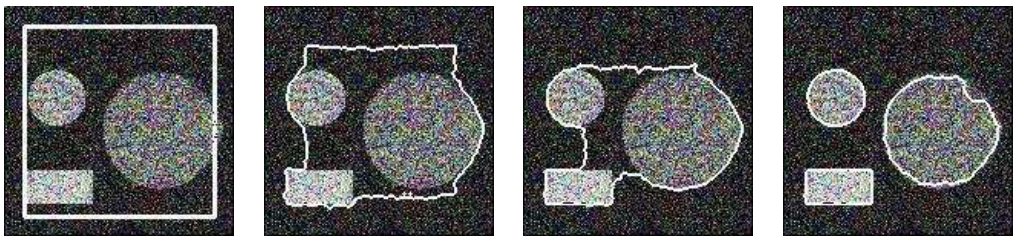
$$E(C) = \int_0^1 \left|\frac{\partial C}{\partial p}\right| dp \tag{5.3}$$

where C is the evolving curve, p is the parameter of C, and $\|\cdot\|$ represents the Euclidean norm operator.

This algorithm illustrates the fundamental idea of curve evolution; however, no information from the image is utilized. In order to perform image segmentation, both image gradient and region information should be used to guide the evolving curve to boundaries of objects in the image. A new curve evolution model is proposed in [109] that extends (5.2):

$$\psi_t = \alpha\phi_l(v + \epsilon\kappa)|\bigtriangledown\psi| + (1-\alpha)\gamma\delta_\beta(\psi)\{(I - c_2)^2 - (I - c_1)^2\} \tag{5.4}$$

Here, ψ is the evolving curve, and I represents the image. The first term in the right hand of (5.4) makes use of the gradient information in the image, where $\phi_l = 1/\{1 + \|\nabla I\|\}$. In this term, v is the inflationary term, and κ is the curvature. The second term uses region information. In this term, $\delta_\beta(x) = \beta/(\pi(x^2 + \beta^2))$ approximates a delta function. c_1 and



(a) (b) (c) (d)

Figure 5.1: Boundary detection using geometric curve evolution. Multiple objects are segmented from noisy images since geometric curve evolution can deal with topological changes automatically. (a) A noisy image with initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.

c_2 are the average values of the points inside and outside the evolving curve, respectively, and α, β, γ and ϵ are positive constant coefficients.

Following the approach of Sethian [130], a numerical algorithm for the model (5.4) can be derived:

$$\begin{aligned}\psi_{k+1} = \psi_k + \triangle t\{\alpha\{\max(\phi_{\mathrm{l}}\upsilon, 0)\nabla^+ + \min(\phi_{\mathrm{l}}\upsilon, 0)\nabla^-\} \\ + (1-\alpha)\cdot\gamma\delta_\beta(\psi)\{(I-c_2)^2 - (I-c_1)^2\}\}\end{aligned} \tag{5.5}$$

where

$$\nabla^+ = \{\max(\mathrm{D}^{-\mathrm{x}}, 0)^2 + \min(\mathrm{D}^{+\mathrm{x}}, 0)^2 + \max(\mathrm{D}^{-\mathrm{y}}, 0)^2 + \min(\mathrm{D}^{+\mathrm{y}}, 0)^2\}^{1/2} \tag{5.6}$$

and

$$\nabla^- = \{\max(\mathrm{D}^{+\mathrm{x}}, 0)^2 + \min(\mathrm{D}^{-\mathrm{x}}, 0)^2 + \max(\mathrm{D}^{+\mathrm{y}}, 0)^2 + \min(\mathrm{D}^{-\mathrm{y}}, 0)^2\}^{1/2} \tag{5.7}$$

in which D^{-x}, D^{+x}, D^{-y} and D^{+y} are the backward and the forward finite differences of ψ in the x and y directions, respectively.

Similarly, the curve evolution model in (5.4) is equivalent to the minimization of the following energy functional

$$E(C) = \alpha\cdot\int_0^1 \left|\frac{\partial C}{\partial p}\right| \phi dp + (1-\alpha)\gamma\left[\iint_{\mathrm{inside(C)}} |I-c_1|^2 dxdy + \iint_{\mathrm{outside(C)}} |I-c_2|^2 dxdy\right] \tag{5.8}$$

An illustration of this algorithm's behavior is given in Fig. 5.1, in which anisotropic diffusion methods [4] [117] are incorporated to reduce the effects of noise. Fig. 5.1(a) represents the image with the initial curve. Fig. 5.1(b) and Fig. 5.1(c) are intermediate results during curve evolution, and Fig. 5.1(d) is the final result. It can be clearly seen that the algorithm segments the image and converges to the boundaries.

5.1.3 Probabilistic Curve Evolution Model

The geometric curve evolution model is clearly nonlinear, and images typically contain noise, either Gaussian or non-Gaussian. A state-space model of the curve evolution can

be constructed, embedding traditional geometric curve evolution models in a probabilistic framework.

The State Model

In the state space model, points selected on and close to the evolving curve $\psi_k^i (i = 1...N)$ form the states of the model. The ideal motions of the states follow a distribution approximating

$$\psi_t = \alpha\phi(\upsilon + \epsilon\kappa)|\nabla\psi| \tag{5.9}$$

where $\phi = 1/\{1 + \|\nabla I\|\}$.

With the presence of additive noise, a finite difference model has the form:

$$\psi_{k+1} = \psi_k + \triangle\psi_k + n_k \tag{5.10}$$

where ψ_k is the vector of the values of the selected points in the image, n_k is the noise affecting process dynamics, and $\triangle\psi_k = \triangle t \cdot \alpha\{\max(\phi\upsilon, 0)\nabla^+ + \min(\phi\upsilon, 0)\nabla^-\}$ measures the difference between the values of the current state and the next state.

The conditional probability density function of ψ^{k+1} given ψ^k is modeled using a Gibbs (or Boltzmann) distribution of the form

$$p(\psi_{k+1}|\psi_k^i) = \frac{1}{P_s}\exp\{-|\psi_{\mathrm{k+1}} - (\psi_{\mathrm{k}}^{\mathrm{i}} + \triangle\psi_{\mathrm{k}}^{\mathrm{i}})|\} = \frac{1}{\mathrm{P_s}}\exp\{-|\psi_{\mathrm{k+1}} - \psi_{\mathrm{k+1}}^{\mathrm{i}}|\}$$

where $\psi_{k+1} \in [-1, 1]$. $p(\psi_{k+1}|\psi_k^i)$ is 0 when ψ_{k+1} is not in $[-1, 1]$, and P_s is a normalization constant, which takes the following value:

$$P_s = 2 - \exp(-1 - (\psi_{\mathrm{k}}^{\mathrm{i}} + \triangle\psi_{\mathrm{k}}^{\mathrm{i}})) - \exp(-1 + (\psi_{\mathrm{k}}^{\mathrm{i}} + \triangle\psi_{\mathrm{k}}^{\mathrm{i}})) \tag{5.11}$$

The Measurement Model

Region information in the image is utilized in the measurement model. Since region information depends on the evolving curve, which is the zero level set of the states $\psi_i^k (i = 1...N)$, this measurement model is a function of the states. Additive noise is assumed to corrupt the measured image data. The measurement model is postulated to be

$$Z_k = f(\psi_k) + \upsilon_k \tag{5.12}$$

where $f(\psi_k) = \triangle t \cdot (1 - \alpha)\gamma\{(I - c_2)^2 - (I - c_1)^2\}$, and $f(\psi_k)$ is normalized to lie between $-\triangle t$ and $\triangle t$. Comparing the formation of $f(\psi_k)$ with (5.5), it corresponds to the the region term of the increments, which is chosen to make use of region information in the geometric model.

The conditional probability density function of Z_k given ψ_k is assumed to be a Gibbs distribution:

$$p(Z_k|\psi_k^i) = \begin{cases} \frac{1}{P_z}\exp\{-|\mathrm{Z_k} - \mathrm{f}(\psi_{\mathrm{k}}^{\mathrm{i}})|\} & \psi_k \in [-1, 1] \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

where $\psi_k \in [-1, 1]$, and P_z is a normalization constant too, taking the form:

$$P_z = 2 - \exp(-1 - \mathrm{f}(\psi_{\mathrm{k}}^{\mathrm{i}})) - \exp(-1 + \mathrm{f}(\psi_{\mathrm{k}}^{\mathrm{i}})) \tag{5.14}$$

Initialization

To initiate the curve's evolution, the initial curve is represented by the zero level set of a two-dimensional plane ψ. The points inside the initial curve are assigned positive values, the points outside it are assigned negative values, and the points on the curve are given value 0. The value of each point is proportional to its distance to the initial curve.

The initial probabilities of the components of the states for the model are assumed to be exponential functions of their initial values, in which a Gibbs distribution is assumed:

$$p(\psi) = \begin{cases} \frac{1}{P_0}\exp\{-|\psi - \psi_0|\} & \psi \in [-1, 1] \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

where $\psi_0 \in [-1, 1]$, ψ is a component of the initial state and P_0 is a normalization constant:

$$P_0 = 2 - \exp(-1 - \psi_0) - \exp(-1 + \psi_0) \tag{5.16}$$

.

The state model and the measurement model are designed to simulate the geometric curve evolution model. In this way, the evolving contour, usually initialized to shrink from the image boundary, is forced to stop at boundaries of objects and thus accomplish image segmentation.

5.1.4 Particle Filters

An overview of the theory of particle filters is given in this section; additional information can be found in [9] [43]. A formation of nonlinear Bayesian state prediction and estimation will be described first.

Nonlinear State Estimation

In this section, a formal description of nonlinear state estimation (prediction and tracking) is given, cast in a Bayesian framework.

For a nonlinear system, the state space model can be described as

$$x_{k+1} = f_k(x_k, v_k) \tag{5.17}$$

$$y_k = h_k(x_k, n_k) \tag{5.18}$$

where x_k are states of the system, f_k and h_k are possibly nonlinear functions, v_k and n_k are i.i.d process noise sequences, $k \in \mathbb{N}$, the set of natural numbers.

Let $p(x_k|y_{1:k})$ be the probability density function (pdf) of x_k given the measurement $\{y_i, i = 1, 2, ..., k\}$. The tracking problem is: Given $p(x_k|y_{1:k})$ and $\{y_i, i = 1, 2, ...k + 1\}$, what are the predicted and updated densities $p(x_{k+1}|y_{1:k})$ and $p(x_{k+1}|y_{1:k+1})$?

State estimation in a Bayesian framework for the tracking problem of the nonlinear system includes two stages - the prediction stage and the update stage. The prediction stage uses the state model to obtain the pdf of the state at time $k + 1$, prior to receipt of a measurement y_{k+1}, via the Chapman-Kolmogorov equation

$$p(x_{k+1}|y_{1:k}) = \int p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k \tag{5.19}$$

where the Markov property of the state space model is used: $p(x_{k+1}|x_k, y_{1:k}) = p(x_{k+1}|x_k)$.

In the update stage, the posterior pdf, given a new measurement y_{k+1}, is calculated using Bayes' rule:

$$p(x_{k+1}|y_{1:k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})}{p(y_{k+1}|y_{1:k})} \tag{5.20}$$

where $p(y_{k+1}|y_{1:k})$ is a normalizing constant.

$$p(y_{k+1}|y_{1:k}) = \int p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})dx_{k+1} \tag{5.21}$$

Analytic solutions of (5.19-5.20) are typically not feasible for a nonlinear system with non-Gaussian noise. In these cases, particle filters, which will be introduced in the next subsection, provide an approximate solution.

Particle Filters

Particle filters [9] [43] are sequential Monte Carlo methods for the approximate solution of the nonlinear filtering equations (5.19-5.20). A probability density function $p(x)$ is approximated by a discrete random measure $\{x^{(n)}, w^{(n)}\}_{n=1}^{N}$, defined by particles $\{x^{(n)}\}$ and weights $\{w^{(n)}\}$. The probability distribution is approximated in the weak convergence by

$$p(x) = \sum_{n=1}^{N} w^{(n)}\delta(x - x^{(n)}) \tag{5.22}$$

where $\delta(.)$ is the delta function. The state estimate problem is converted to the prediction and update of the particles' positions and weights.

When sampling from $p(x)$ is computationally intractable, one can generate particles $x^{(n)}$ from a distribution $\pi(x)$, known as an importance function. Weights are assigned according to

$$w^{*(n)} = \frac{p(x)}{\pi(x)} \tag{5.23}$$

and then normalized to obtain the weights $w^{(n)}$. using the following formula:

$$w^{(n)} = \frac{w^{*(n)}}{\sum_{k=1}^{N} w^{*(k)}} \tag{5.24}$$

Suppose the posterior distribution $p(x_{1:k}|y_{1:k})$ is approximated by the discrete random measure
$\chi_k = \{x_{0:k}^{(n)}, w_k^{(n)}\}_{n=1}^{N}$. The estimation problem is transformed to: given χ_k and y_{k+1}, find χ_{k+1}. This reduces to finding $x_{k+1}^{(n)}$ and $w_{k+1}^{(n)}$.

If an importance function $\pi(x_{0:k+1}|y_{0:k+1})$ can be factored,

$$\pi(x_{0:k+1}|y_{0:k+1}) = \pi(x_{k+1}|x_{0:k}, y_{0:k+1})\pi(x_{0:k}|y_{0:k}) \tag{5.25}$$

then the solution to the tracking problem can be accomplished using the update:

$$x_{k+1}^{(n)} \sim \pi(x_{k+1}|x_{0:k}^{(n)}, y_{0:k+1}) \tag{5.26}$$

$$w_{k+1}^{(n)} \propto \frac{p(y_{k+1}|x_{k+1}^{(n)})p(x_{k+1}^{(n)}|x_k^{(n)})}{\pi(x_{k+1}^{(n)}|x_{0:k}^{(n)}, y_{0:k+1})} w_{0:k}^{(n)} \tag{5.27}$$

Selection of the importance function remains, which plays a key role in particle filtering. Two frequently used importance functions are the prior and the optimal importance function. If the importance function is selected as the prior importance function, given by $p(x_{k+1}|x_k^{(n)})$, then the weights update as follows [9] [43]:

$$w_{k+1}^{(n)} \propto w_k^{(n)} p(y_{k+1}|x_{k+1}^{(n)}) \tag{5.28}$$

The optimal importance function is designed to minimize the variance of $p(x_{k+1}^i|y_{1:k+1})/\pi(x_{k+1}^i|x_k^i, y_{k+1})$, which is $p(x_{k+1}|x_{0:k}^{(n)}, y_{0:k+1})$. The corresponding weight update is

$$w_{k+1}^{(n)} \propto w_k^{(n)} p(y_{k+1}|x_k^{(n)}) \tag{5.29}$$

Although the optimal importance function usually provides better performance than the prior importance function, it is more difficult to implement, because both integration and sampling from $p(x_{k+1}|x_{0:k}^{(n)}, y_{0:k+1})$ are required in this case. In the proposed model, the prior model is selected to be the importance function, and the weights are updated using (5.28). A more complete description for particle filters is provided in [9] [43].

A major problem with particle filtering is that the discrete random measure degenerates very quickly. In a few iterations, all the particles except a few are assigned negligible weights. Since the variance of the importance weights increases over time, it is not possible to avoid this degeneracy. To reduce its effects, re-sampling methods can be used. Re-sampling removes particles with very small weights and replicates those with large weights.

The above description gives the basis for most particle filters. Other particle methods, such as the sampling importance re-sampling (SIR) filter and the regularized particle filter (RPF) have also been proposed to deal with the problems listed above. They are special cases of the above sequential importance sampling (SIS) problem with modifications to importance functions and the re-sampling step. For more details, refer to [9] [43].

5.1.5 Implementation

In this section, several implementation issues are discussed. The first problem is sampling for a pdf. The method in [62] is utilized for this problem. Sampling in this method is performed by selection of a piecewise constant function whose integral over any interval approximates the pdf, and a pseudo-random number generator is used to select points according to this approximation density. Emphasis of this section will be placed on the implementations for both prediction and update.

The Prediction Step

In this probabilistic curve evolution model, evolution is accomplished in the prediction stage, predicting the states of the evolving curve at the next iteration using current state and the observed data (the region information). The prediction stage is implemented (5.19), from

which the prediction is derived:

$$
\begin{aligned}
p(x_{k+1}|y_{1:k}) &= \int p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k \\
&= \int_{-1}^{1} p(x_{k+1}|x_k) \sum_{i=1}^{N} w_k^{(i)} \delta(x_k - x_k^{(i)}) dx_k \\
&= \sum_{i=1}^{N} w_k^{(i)} \int_{-1}^{1} p(x_{k+1}|x_k)\delta(x_k - x_k^{(i)}) dx_k \\
&= \sum_{i=1}^{N} w_k^{(i)} p(x_{k+1}|x_k^{(i)}) \\
&= \sum_{i=1}^{N} w_k^{(i)} \cdot exp\{-|\psi_{k+1} - \psi_k^{(i)} - \nabla\psi_k^{(i)}|\} \qquad (5.30)
\end{aligned}
$$

$p(x_{k+1}|y_{1:k})$ is assumed to take the form of $exp\{-|\psi_{k+1} - \psi_{k+1}^{(i)}|\}$ to be consistent with the state model, in which $\psi_{k+1}^{(j)}$ represents the next state of the evolving contour. For the exponential distribution, most of the particles lie near the peak, and the absolute value of $\psi_{k+1} - \psi_{k+1}^{(i)}$ is very close to zero, in which case

$$
exp\{-|\psi_{k+1} - \psi_{k+1}^{(i)}|\} \approx 1 - |\psi_{k+1} - \psi_{k+1}^{(i)}| \qquad (5.31)
$$

The following formula is obtained and used to update the states of the evolving curve:

$$
\begin{aligned}
p(x_{k+1}|y_{1:k}) &= \frac{1}{P} exp\{-|\psi_{k+1} - \psi_{k+1}^{(i)}|\} \\
&= \frac{1}{P} exp\{-|\psi_{k+1} - \sum_{i=1}^{N} w_k^{(i)} (\psi_k^{(i)} + \nabla\psi_k^{(i)})|\} \qquad (5.32)
\end{aligned}
$$

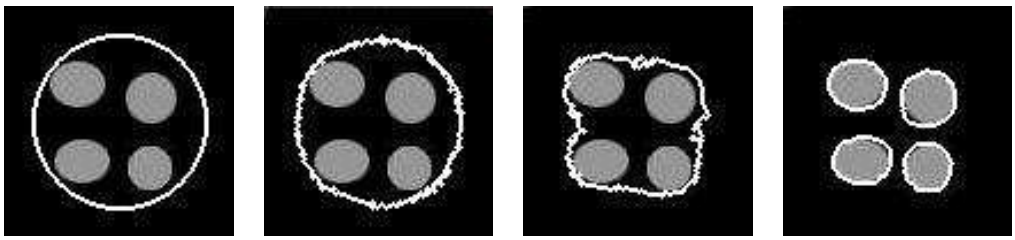
The state at the next iteration is the weighted summation of the updated densities of all the particles, which is intuitive. Experimental results show that this approximation method works.

The Update Step

In the update step, region information is utilized to direct the evolution of the evolving curve. The update step would normally be implemented using (5.28):

$$
w_{k+1}^{(n)} \propto w_k^{(n)} p(y_{k+1}|x_{k+1}^{(n)})
$$

However, a slight revision is made: If the sign of the region information $f(\psi_k) = \triangle t \cdot (1 - \alpha)\gamma\{|I - c_2|^2 - |I - c_1|^2\}$ is same as the difference $\nabla\psi_k^{(i)} = \psi_{k+1}^{(i)} - \psi_k^{(i)}$ for a particle, the weight for that particle is multiplied by $exp\{|f(\psi_k) * \phi_l|\}$; otherwise, the weight is multiplied by $exp\{|f(\psi^k) * \phi_l|\}$. In this way, both the gradient and the region information are utilized and affect the evolution of the curve, causing it to stop at object boundaries.



(a) (b) (c) (d)

Figure 5.2: Topological change of the probabilistic curve evolution. (a) An image with initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.

5.1.6 Experimental Results

In this section, experimental results are presented and analyzed. In all the experiments shown below, the parameter α is set to be 0.2. The narrow-band method in [130] is utilized.

For the first experiment, the ability of the proposed model to undergo topological change is tested. The number of particles for each state is set to be 1300. An initial simply connected closed curve is used from Fig. 5.2(a) - Fig. 5.2(d). The topological change of the evolving curve to four disjoint curves is successful.

Next, the role of the region information in the model is tested. The number of particles for each state is once again 1300. The evolving curve is set to shrink. The region information causes the evolving contour to converge to the boundary of the object. This is illustrated in images Fig. 5.3(a) - Fig. 5.3(d).

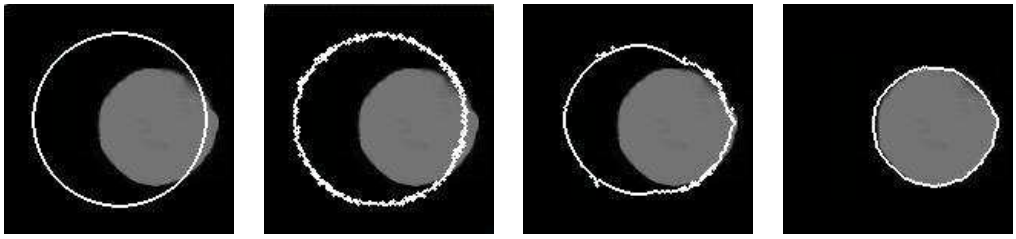
The third experiment applies the proposed model to a noisy real image, as shown in Fig. 5.4 and 5.5. As in Fig. 5.1, anisotropic diffusion methods [117] [4] are utilized to reduce the effects of noise. Comparison of Fig. 5.4 to Fig. 5.1 shows that the proposed probabilistic model achieves similar results to the geometric model. From Fig. 5.5(a) - Fig. 5.5(d), it can be seen that the airplane in the noisy image is successfully segmented.

5.1.7 Conclusion

A probabilistic curve evolution model is proposed, based on the geometric model and a Gibbs distribution. Both the gradient and the region information are utilized for curve evolution. Curve evolution is accomplished using particle filters in a Bayesian framework. Experimental results show that this model does work, and good results have been achieved. The results are comparable to those acquired using geometrical models. A potential problem with the proposed model is that it is computationally intense because of the large number of the states. Further research will be focused on the improvement of the model to increase evolution speed.

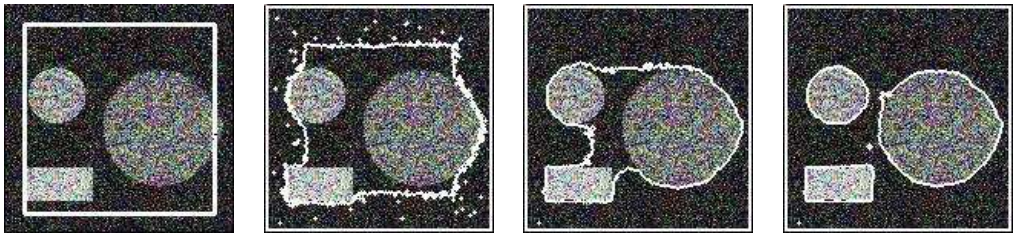
5.2 Preferential Image Segmentation by Matching Histogram-based Distributions

A weighted Kullback-Leibler distance is proposed in this section to measure the differences between two color distributions. The procedure follows [53] with some improvements.



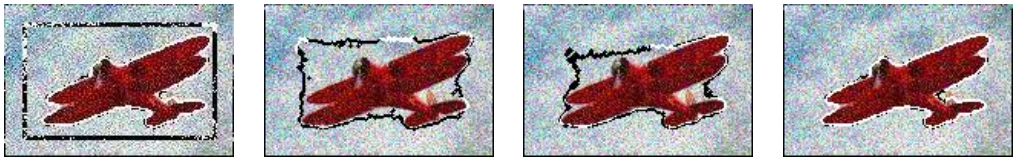
(a) (b) (c) (d)

Figure 5.3: The role of region information in the probabilistic curve evolution. (a) An image with initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.



(a) (b) (c) (d)

Figure 5.4: Segmentation of multiple regions with noise. (a) An image with initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.



(a) (b) (c) (d)

Figure 5.5: Segmentation of real noisy images. (a) An image with initial curve. (b) Intermediate results of curve evolution. (c) Intermediate results of curve evolution. (d) Final segmentation results.

Let z be a photometric variable of interest. Suppose $q(z)$ represents the color distribution of the prior image and $p(z)$ is the color distribution of parts of the image (e.g. the parts inside the evolving curve) to be segmented. The weighted Kullback-Leibler distance, which measures the difference between $q(z)$ and $p(z)$, is proposed:

$$\begin{aligned} K_w(\Omega) &= \int_Z q(z) w_{pq}(z) \log \frac{\mathrm{q(z)}}{\mathrm{p(z;\Omega)}} \mathrm{dz} \\ &= \eta - \int_Z q(z) w_{pq}(z) \mathrm{logp(z;\Omega)dz} \end{aligned} \tag{5.33}$$

Here η is the negative weighted differential entropy of the model distribution, and $w_{pq}(z)$ in (5.33) is a positive bounded function, which is used here as a weight function to measure the differences in spatial color distribution between the known object and the object to be segmented. Since the density distribution is represented using a histogram, $p(z;\Omega) = N(z;\Omega)/A(\Omega)$ where $A(\Omega)$ is the area of Ω, and

$$K_w(\Omega) = \eta + M \cdot \log(\mathrm{A}(\Omega)) - \int_{\mathrm{Z}} \mathrm{q(z) w_{pq}(z) log(N(z;\Omega))dz}$$

where $M = \int_Z q(z) w_{pq}(z) dz$. Thus

$$\frac{\delta K_w}{\delta \mathbf{c}} = \frac{M}{A} \frac{\delta A}{\delta \mathbf{c}} - \int_Z q(z) w_{pq}(z) \left[\frac{1}{N(z;\Omega)} \frac{\delta N(z)}{\delta \mathbf{c}} \right] dz$$

where $\mathbf{c}$ corresponds to an entry in the histogram.

Since from [53]and [163], we have

$$A(\Omega) = \int_\Omega dx \Rightarrow \frac{\delta A}{\delta \mathbf{c}} = \mathbf{n}$$

and

$$N(z;\Omega) = \int_\Omega \delta(z - Z(x)) dx \Rightarrow \frac{\delta N(z)}{\delta \mathbf{c}} = \delta(z - Z(\mathbf{c}))\mathbf{n}$$

Thus

$$\begin{aligned} \frac{\delta K_w}{\delta \mathbf{c}} &= \frac{M}{A}\mathbf{n} - \int_\Omega \{ \frac{q(z) w_{pq}(z)}{N(z)} \delta(z - Z(\mathbf{c})) dz \} \mathbf{n} \\ &= \{ \frac{M}{A} - \frac{q(Z(\mathbf{c})) w_{pq}(Z(\mathbf{c}))}{N(Z(\mathbf{c}))} \} \mathbf{n} \\ &= \frac{M \cdot p(Z(\mathbf{c})) - q(Z(\mathbf{c})) w_{pq}(Z(\mathbf{c}))}{N(Z(\mathbf{c}))} \mathbf{n} \\ &= \frac{M}{N(Z(\mathbf{c}))} \cdot \{ p(Z(\mathbf{c})) - \frac{q(Z(\mathbf{c})) w_{pq}(Z(\mathbf{c}))}{M} \} \mathbf{n} \\ &= \frac{M}{N(Z(\mathbf{c}))} \cdot \{ p(Z(\mathbf{c})) - \frac{q(Z(\mathbf{c})) w_{pq}(Z(\mathbf{c}))}{\int_Z q(Z(\mathbf{c})) w_{pq}(Z(\mathbf{c})) dz} \} \mathbf{n} \end{aligned} \tag{5.34}$$

Here the function w_{pq} needs to carry the spatial information of the distributions. Note that this function is assumed to be independent of the evolving curve C.

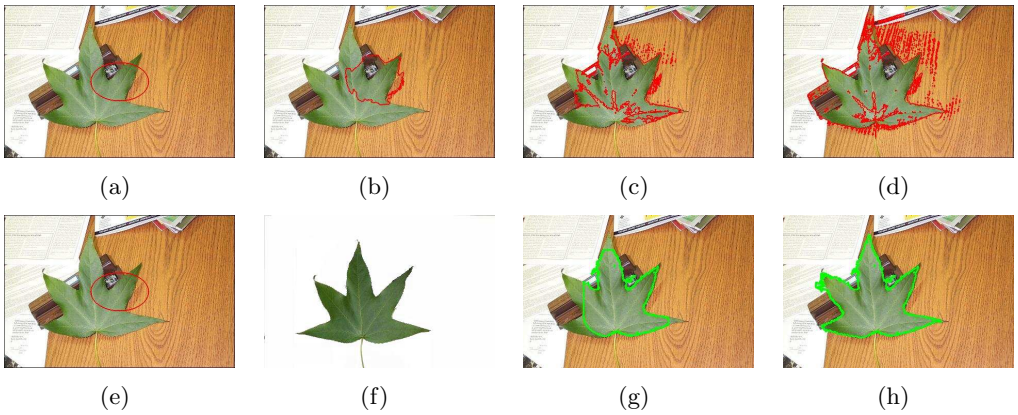


Figure 5.6: Segmentation of a leave image based on prior color information (a) An image with initial curve. (b) Intermediate results of curve evolution using the Chan-Vese model. (c) Intermediate results of curve evolution using the Chan-Vese model. (d) Final segmentation results using the Chan-Vese model. (e) A leave image with initial curve. (f) Image containing prior information. (g) Intermediate results of prior based curve evolution. (h) Final segmentation results of prior based curve evolution.

The above model generalizes the model in [53] in two ways. First, the model makes [53] a special case where $w_{pq} = 1$. Second, the model utilizes not only prior gray intensity information, but also prior color and texture information.

Preliminary experimental results of the proposed model are provided in Fig. 5.6. w_{pq} is chosen as 1 in the implementation, and the hue information in the prior image is utilized.

Fig. 5.6 compares segmentation results using the proposed model and the Chan-Vese model. Fig. 5.6 (a)-(d) show the results using the Chan-Vese model without using any prior information. Due to the complexity of the image, the Chan-Vese model does not generate good results. The leave in the center of the image is not segmented. Fig. 5.6 (e)-(h) show the results using the model above, in which the hue information of the model image Fig. 5.6 (f) is utilized. The leave in the image is successfully segmentation by utilizing the prior information.

In summary, an improvement of the preferential image segmentation model in [53] is proposed. First, color and texture prior information can be utilized in the model. Different weights of prior information can be utilized for better segmentation. Experimental results using prior color information are provided.

The model can be improved in several ways. Kernel based learning methods such as the Expectation-Maximization (EM) method and kernel-based principle component analysis [44] may replace histogram method to model the prior information. Alternative modifications of Kullback-Leibler divergence are also available in the framework of adaptive information filtering [46]. These modifications will be implemented and compared with the one proposed above.

5.3 Summary

Two approaches are developed to segment objects by matching distributions in this chapter. A probabilistic curve evolution model utilizes particle filters for image segmentation.

A preferential image segmentation method is presented to segment objects of interest by matching histogram-based distributions using the Kullback-Leibler distance.

Chapter 6

Conclusions

"A journey of a thousand miles begins with a single step."
—Confucius

This dissertation describes research results in image segmentation and its related areas, using the techniques of partial differential equations, variational methods, mathematical morphological methods and probabilistic methods. An integrated segmentation method using both curve evolution and anisotropic diffusion methods is described that utilizes both gradient and region information in images [108] [109]. A bottom-up image segmentation method is described that minimizes the Mumford-Shah functional [111] [112] [107] [113]. A preferential image segmentation method is described that is based on the tree of shapes in mathematical morphologies. A thorough evaluation of the proposed method is performed and a framework based web interface is developed. A probabilistic model is described that is based on particle filters for image segmentation. Parallel computing results using graphic processors are also presented. Applications such as video tracking using IP cameras are also explored.

These methods cooperate to formulate an integrated image processing system. Video frames are captured from IP cameras as the input to the system. The methods listed above are utilized for the preprocessing, segmentation and recognition of the captured images. An image database provides prior information for processing and stores the processed results for further processing. A framework for applications such as video tracking is implemented in the dissertation.

The dissertation shows the potentials of image processing in various aspects, including mathematical theories (PDEs, variational methods, morphological methods, probabilistic methods, etc.), computational techniques (multi-scale computation, parallel computation using graphic processors, etc.) and implementations (C++, Java, MySQL, Perl, matlab, OpenGL, etc.). These areas cover many issues that have not been thoroughly explored. Future research for every proposed method has been provided in the end of every chapter. An overview for future research in a broader view is provided here.

- Explore wavelet methods and stochastic methods for image processing. Wavelet methods [41] [93] provide good tools from the aspect of signal processing. Stochastic methods [8] [72] may provide more robust algorithms in many cases such as curve evolution. The application of both fields in image processing is a very good topic for further research. Their combination with the proposed methods in this dissertation may give out better performance in accuracy, stability, and efficiency.

- Examine pattern theory. A mathematical study of regular structures as in [59] in preferential image segmentation will benefit a lot because it will show the fundamental differences between patterns and give more hints to improve better performance for pattern recognition.

- Study the topics covered in the dissertation more deeply and more broadly. The dissertation covers several topics, each of which may be a huge topic itself. It is necessary to explore these topics further. For example, it would be beneficial to study Sobolev active contours [138] [94] [139] and stochastic active contours [72] [149] which are two important advances for active contours.

- Integrate the image processing system further. The framework of the system has been built in the dissertation. But it still needs work to make it more integrated and more efficient. The system may be suitable for more applications after that.

Advances in two topics are presented here before the conclusion of this dissertation. Image processing using graphic processors is shown in section 6.1. Applications for the preferential image segmentation on video capture and video tracking are presented in section 6.2.

6.1 Image Processing using Graphic Processors

Parallel computing using graphic processors (GPU) has become an important topic for image processing, as reviewed in section 2.3.5. Some results on image processing using graphic processors are presented here.

The **Perl OpenGL** package [103], the **Perl ImageMagick** package [67] and the **OpenGL Shading Language (GLSL)** [121] are utilized to accomplish the image processing results in this chapter.

Image processing using graphic processors usually proceeds as follows. The images to be processed are read from files into the CPU memory. The data from these images are shared by texture memories in the OpenGL pipeline, in this case by means of the **Perl OpenGL** package [103]. An advantage is that it is not necessary to copy the image data from CPU memory to GPU memory [103], which is time-consuming. The **fragment shader** and the **vertex shader** can access the data in texture memories using the **Sample2D** data structure provided by the GLSL language [121]. The GLSL language also provides the capacities for parallel processing. The processing takes place in the frame buffer of the OpenGL pipeline [134]. The results can then finally be copied from the frame buffer back to the CPU memory and stored as files. It has been tested in the OpenGL benchmarks that the performance of the **Perl OpenGL** approaches that of C when the module and the interpreter have been loaded into memory. Furthermore, the **Perl OpenGL** package saves the time to copy image data from CPU memory to GPU memory.

Fig. 6.1 shows some results on image processing using graphic processors. It performs the addition of two images I_1 and I_2 using $I = \alpha \cdot I_1 + (1-\alpha) \cdot I_2$ with $0 \leq \alpha \leq 1$. Fig. 6.1(a) (I_1) and Fig. 6.1(b) (I_2) show the images to be added. Fig. 6.1(c) and Fig. 6.1(d) shows the images in the frame buffer. Fig. 6.1(e), Fig. 6.1(f) and Fig. 6.1(g) show the processing results using nVidia GeForce 6800 with $\alpha = 0.25$, $\alpha = 0.50$ and $\alpha = 0.75$ respectively. Each addition procedure using graphic processors takes 0.0167 seconds. The same procedure implemented in CPU using Perl takes 0.0200 seconds on average. The GPU processing is 1.2 times faster than the CPU processing in this case.

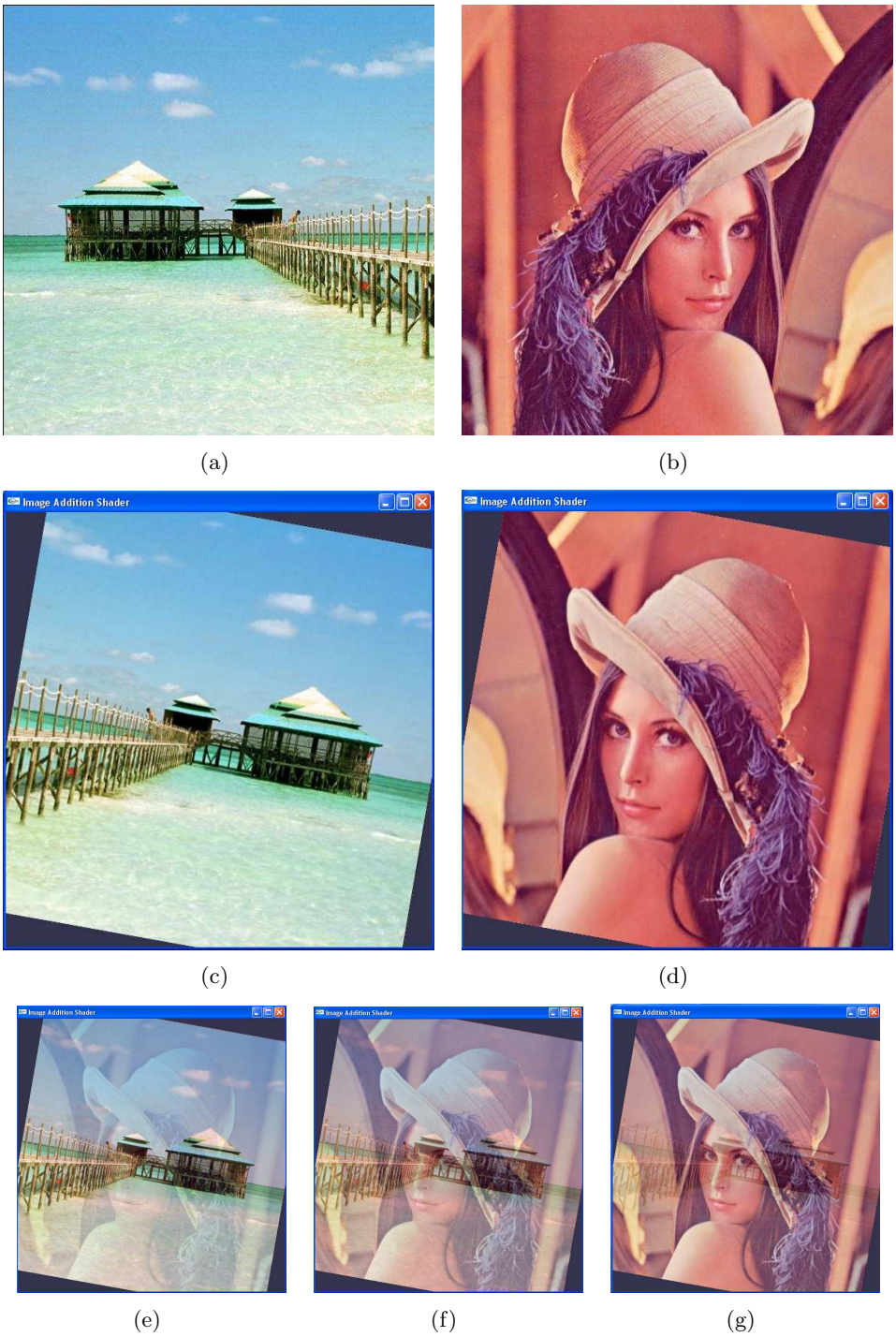


Figure 6.1: Image addition by GPGPU. (a) The image I_1 to be processed (size 512 * 512). (b) The image $I_2$to be processed (size 512 * 512). (c) The addition results with $\alpha = 0.0$. (d) The addition results with $\alpha = 1.0$. (e) The addition results with $\alpha = 0.25$. (f) The addition results with $\alpha = 0.50$. (g) The addition results with $\alpha = 0.75$. The addition takes 0.0167 second.

(a) (b) (c) (d)

Figure 6.2: Video capture from IP cameras using Java media framework. (a) Image Captured at the Science and Engineering Research Facility (SERF) at the University of Tennessee (UT). (b) Image Captured at the Center for International Education (CIE) at UT. (c) Image Captured at CIE at UT at night. (d) Image Captured at CIE at UT at night.

In summary, this chapter illustrates general purpose computation using graphic processors using an example of image mixing, which shows that graphic processors have much potential in image processing. The author has only one month to spend on GPGPU. There are a lot of future to be done in this topic. This includes the implementations of the bottom-up image segmentation method, the curve evolution methods and the preferential image segmentation methods shown in previous chapters.

6.2 Applications using Preferential Image Segmentation

The proposed preferential image segmentation methods may be applied to several real applications. Section 6.2.1 describes how to capture video frames from IP cameras. Section 6.2.2 shows the application of preferential segmentation methods on video tracking using IP cameras.

6.2.1 Video Capture from IP Cameras

IP cameras (also called network cameras) are very popular for video surveillance. These cameras have high resolution (as much as 640 by 480), low price (several hundred dollars) and small size. With a network connection and power supply, these cameras can be easily configured for surveillance.

Web servers are usually built inside these cameras so that the captured videos can be viewed conveniently using a web browser. The videos, however, need to be captured and processed for video tracking and other applications. A video capture program based on the Java Media Framework (JMF) [153] has been developed to capture the video frame sequences from AXIS 206 Network cameras [36]. These cameras cost 300 dollars each and can capture 30 frames per second with resolutions up to 640 by 480.

During the implementation, the URL and URLConnection Java class from JMF are utilized to communicate with the web server built in the cameras. An example the web server URL for a camera is "http://cam1.asa.utk.edu/axis-cgi/jpg/image.cgi?resolution=320x240". Note that parameters such as resolutions can be set in the URL. More detail on the API specifications is available at "http://www.axis.com/techsup/cam_servers/dev/cam_http_api.htm". The stream classes in JMF then retrieve the contents of the video frames and store them as JPEG images using JPEG codecs. Fig. 6.2 shows some sample images captured using this program.

In real-time video tracking applications, C++ is probably a better choice than Java for video capture and tracking. The captured video frames can be efficiently processed in memory and need not be stored as JPEG images.

6.2.2 Preferential Image Segmentation using IP Cameras for Object Tracking

IP cameras can provide images and videos in video tracking applications. The preferential image segmentation methods described in section 5.2 can be utilized to segment objects of interest. The proposed preferential segmentation methods are applied to video tracking in Fig. 6.3, in which Fig. 6.3(b), Fig. 6.3(c) and Fig. 6.3(d) correspond to the 5th, 10th, and 15th frame of a video clip. The prior shape provided in Fig. 6.3(a) is utilized for preferential segmentation in these video frames. The segmentation results show the correct locations of the car in each video frame. The prior information about the car in Fig. 6.3(a) enables the method to track the car for the first 15 video frames. The proposed method may be improved for video tracking by adaptively incorporating the prior information with the segmentation results from the previous video frames.

No motion information is utilized in the above results. Video tracking methods such as mean shift [34] may then be applied to track the movement of segmented objects. This will provide a framework for preferential video tracking.

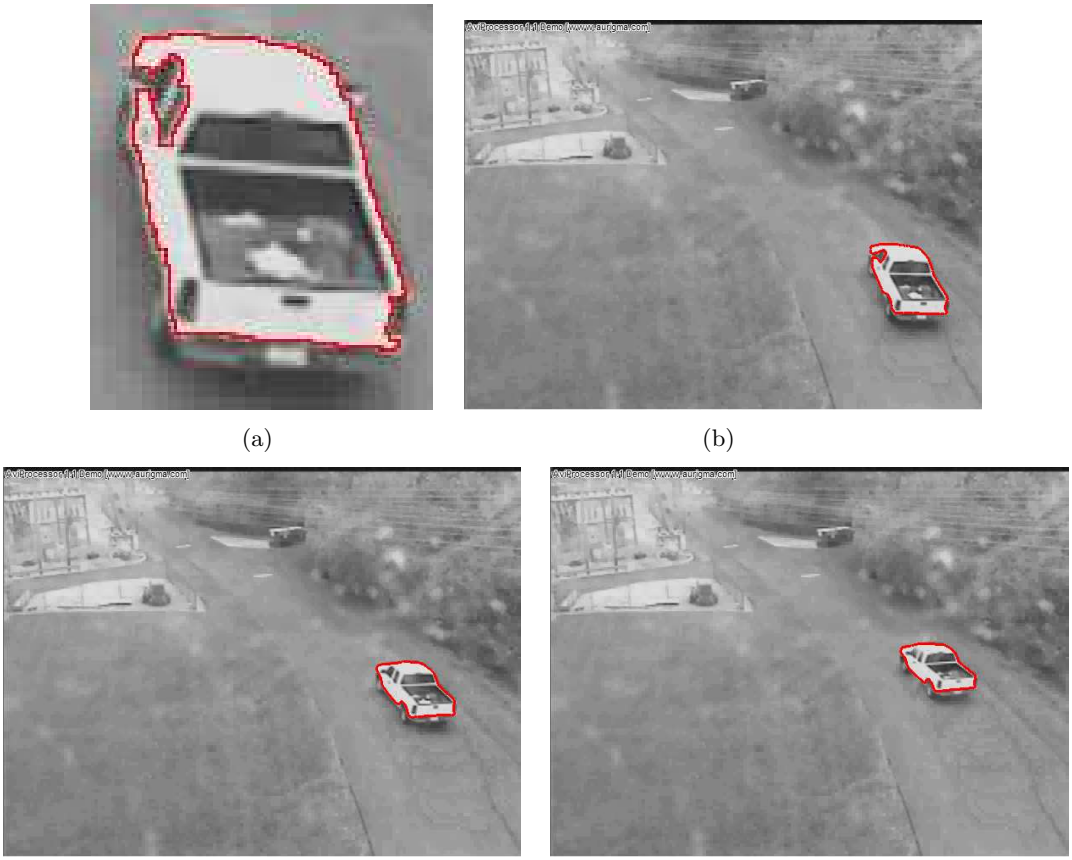
6.2.3 Summary for Video Tracking

Image acquisition using IP cameras are introduced in this chapter, followed by the application of the preferential segmentation methods in Chapter 4 to video tracking. It shows that the proposed morphological method works well for video tracking. Image acquisition will be incorporated to the framework presented in section 4.6 in future research. Automatic prior shape update is the focus of future research for video tracking using morphological methods in Chapter 4.

6.3 Summary

A summary and possible topics for future research are presented in this chapter. Advances on two topics, image processing using graphic processors and video processing using preferential image segmentation, are also shown in this chapter.

Image processing and computer vision have developed quickly. Their history is relatively short, and there still exist many open problems such as image segmentation and performance evaluation. The author strongly believes there will be much important progress in image processing in the next thirty years. Image processing will become a mature field and play an important role in our daily lives.



(a) (b) (c) (d)

Figure 6.3: Preferential image segmentation for video tracking. (a) The prior image with selected shape. (b) Segmentation results for the 5th video frame. Image size: 460 * 612. CPU time: 1.62s. (c) Segmentation results for the 10th video frame. Image size: 460 * 612. CPU time: 1.61s. (d) Segmentation results for the 15th video frame. Image size: 460 * 612. CPU time: 1.61s.

Bibliography

Bibliography

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] T. Akenine-Moller and E. Haines. *Real-Time Rendering*. A K Peters, second edition, 2002.

[3] L. Alvarez, A.-P. Blanc, L. Mazorra, and F. Santana. Geometric invariant shape representations using morphological multiscale analysis. *Journal of Mathematical Imaging and Vision*, 18(2):145–168, 2003.

[4] L. Alvarez, R. Lions, and J. Morel. Image selective smoothing and edge detection by nonlinear diffusion ii. *SIAM Journal on Numerical Analysis*, 29(3):845–866, 1992.

[5] L. Ambrosio, V. Caselles, S. Masnou, and J.M. Morel. Connected components of sets of finite perimeter and applications to image processing. *Journal of the European Mathematical Society*, 3(1):213–266, 2001.

[6] L. Ambrosio and V.M. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[8] G.B. Arous, A. Tannenbaum, and O. Zeitouni. Stochastic approximations to curve-shortening flows via particle systems. *Journal of Differential Equations*, 195(1):119–142, 2003.

[9] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[10] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147. Springer, 2002.

[11] G. Aubert and L. Vese. A variational method in image recovery. *SIAM Journal on Numerical Analysis*, 34(5):1948–1979, 1997.

[12] J. Aujol, G. Aubert, and L. Blanc-Féraud. Wavelet-based level set evolution for classification of textured images. *IEEE Transactions on Image Processing*, 12(12):1634–1641, 2003.

[13] C. Ballester and V. Caselles. The M-components of level sets of continuous functions in WBV. *Publicacions Matemátiques*, 45:477–527, 2001.

[14] C. Ballester, V. Caselles, L. Igual, and L. Garrido. Level lines selection with variational models for segmentation and encoding. *Journal of Mathematical Imaging and Vision*, 27(1):5–27, 2007.

[15] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 2003.

[16] P. Blomgren and T.F. Chan. Color TV: Total variation methods for restoration of vector valued images. *IEEE Transactions on Image Processing*, 7(3):304–309, 1998.

[17] E. Calabi, P.J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker. Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision*, 26(2):107–135, 1998.

[18] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[19] M.D. Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[20] R.A. Carmona and S. Zhong. Adaptive smoothing respecting feature directions. *IEEE Transactions on Image Processing*, 7(3):353–358, 1998.

[21] V. Caselles. Topographic maps and local contrast changes in natural images. *International Journal of Computer Vision*, 33(1):5–27, 1999.

[22] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.

[23] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[24] V. Caselles, J.L. Lisani, J.M. Morel, and G. Sapiro. Shape preserving local histogram modification. *IEEE Transactions on Image Processing*, 8(2):220–230, 1999.

[25] V. Caselles and P. Monasse. Grain filters. *Journal of Mathematical Imaging and Vision*, 17(3):249–270, 2002.

[26] T.F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, 2005.

[27] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[28] J. Chen, T.N. Pappas, A. Mojsilović, and B.E. Rogowitz. Adaptive perceptual color-texture image segmentation. *IEEE Transactions on Image Processing*, 14(10):1524–1536, 2005.

[29] Y. Chen, H.D. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.

[30] Y. Chen, B. Vemuri, and L. Wang. Image denoising and segmentation via nonlinear diffusion. *Computers and Mathematics with Applications*, 39(5/6):131–149, 1996.

[31] D.L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106(1):77–91, 1993.

[32] M.F. Cohen and R. Szeliski. The moment cameras. *Computer*, 39(8):40–45, 2006.

[33] B. Coll and J. Froment. Topographic maps of color images. In *Proceedings of International Conference on Pattern Recognition*, volume 3, pages 613–616, Barcelona, Spain, Sep. 2000.

[34] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, SC, USA, Jun. 2000.

[35] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[36] Axis Corp. Axis 206 network camera. *http://www.axis.com/products/cam_206/index.htm/*.

[37] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.

[38] D. Cremers and G. Funka-Lea. Dynamical statistical shape priors for level set based sequence segmentation. In *Variational and Level Set Methods in Computer Vision*, pages 210–221, 2005.

[39] D. Cremers, N. Sochen, and C. Schnorr. Multiphase dynamic labeling for variational recognition-driven image segmentation. In *Proceedings of European Conference on Computer Vision*, volume 3024, pages 74–86, Prague, Czech Republic, May 2004.

[40] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.

[41] I. Daubechies. *Ten Lectures on Wavelets*. Capital City Press, second edition, 1992.

[42] P. Debevec. Virtual cinematography: Relighting through computation. *Computer*, 39(8):57–65, 2006.

[43] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.

[44] R.O. Duba, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2000.

[45] P. DuBois. *MySQL: the definitive guide to using, programming and administering MySQL 4 databases*. Developer's Library, 2 edition, 2003.

[46] D. Erdogmus and J.C. Principle. From linear adaptive filtering to nonlinear information processing. *IEEE Signal Processing Magazine*, 23(6):14–33, 2006.

[47] Y. Chen etc. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002.

[48] L.C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.

[49] L.C. Evans. *Entropy and Partial Differential Equations*. http://math.berkeley.edu/ evans/entropy.and.PDE.pdf, 2001.

[50] L.C. Evans and R.F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.

[51] K. Falconer. *Fractal Geometry-Mathematical Foundations and Applications*. Wiley, 1990.

[52] D. Freedman, R.J. Radke, and T. Zhang. Model-based segmentation of medical imagery by matching distributions. *IEEE Transactions on Medical Imaging*, 24(3):281–292, 2005.

[53] D. Freedman and T. Zhang. Active contours for tracking distributions. *IEEE Transactions on Image Processing*, 13(4):518–526, 2004.

[54] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 23(11):69–96, 1986.

[55] S. Gao and T.D. Bui. Image segmentation and selective smoothing by using the Mumford-Shah model. *IEEE Transactions on Image Processing*, 14(10):1537–1549, 2005.

[56] P.J. Giblin and B.B. Kimia. On the intrinsic reconstruction of shape from its symmetries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):985–911, 2003.

[57] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson Education, second edition, 2002.

[58] M. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26(2):285–314, 1987.

[59] U. Grenander. *General pattern theory - A mathematical study of regular structures*. Clarendon Press, 1993.

[60] F. Guichard and J.-M. Morel. *Image Analysis and P.D.E.s*. Preprint, http://citeseer.ist.psu.edu/guichard01image.html, 2001.

[61] L. Guigues, J.P. Cocquerez, and H.L. Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.

[62] O. Haggstrom. *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, 2002.

[63] A. Hanbury and J. Serra. Mathematical morphology in the HLS colour space. In *Proceedings of British Machine Vision Conference*, pages 451–460, Manchester, United Kingdom, Sep. 2001.

[64] A. Hanbury and J. Serra. A 3d-polar coordinate colour representation suitable for image analysis. Technical report, PRIP-TR-77, PRIP, T.U. Wien, 2002.

[65] M.T. Heath. *Scientific Computing: An Introductory Survey.* McGraw-Hill Science/Engineering/Math, second edition, 2001.

[66] M. Heiler and C. Schnorr. Natural image statistics for natural image segmentation. *International Journal of Computer Vision*, 63(1):5–19, 2005.

[67] Perl ImageMagick. http://www.imagemagick.org/script/perl-magick.php.

[68] MathWorks Inc. *Matlab Wavelet Toolbox Manual.* http://www.mathworks.com/access/helpdesk/help/pdf_doc/wavelet/wavelet_ug.pdf.

[69] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink. Cpm: A deformable model for shape recovery and segmentation based on charged particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1320–1335, 2004.

[70] J.Chung and L. Vese. Image segmentation using a multilayer level-set approach. *CAM Report*, 3(53), 2005.

[71] S. Jehan-Besson, M. Barlaud, and G. Aubert. DREAM^{2}S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *International Journal of Computer Vision*, 53(1):45–70, 2003.

[72] O. Juan, R. Keriven, and G. Postelnicu. Stochastic motion and the level set method in computer vision: stochastic active contours. *International Journal of Computer Vision*, 69(1):7–25, 2006.

[73] E.R. Kandel, J.H. Schwartz, and T.M. Jessell. *Principles of Neural Science.* Elsevier, third edition, 1991.

[74] M. Kass, A. Witkin, and D. Terzopoulos. Snakes-active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[75] D.G. Kendall, D. Barden, T.K. Garne, and H. Le. *Shape and Shape Theory.* John Wiley & Sons, LTD, 1999.

[76] C. Kervrann and A. Trubuil. Optimal level curves and global minimizers of cost functionals in image segmentation. *Journal of Mathematical Imaging and Vision*, 17(2):153–174, 2002.

[77] S. Kichenassamy. The perona-malik paradox. *SIAM Journal on Applied Mathematics*, 57(5):1328–1342, 1997.

[78] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of International Conference on Computer Vision*, pages 810–815, Boston, MA, USA, Jun. 1995.

[79] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. Toward a computational theory of shape: an overview. In *Proceedings of European Conference on Computer Vision*, pages 402–407, Antibes, France, Apr. 1990.

[80] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3):189–224, 1995.

[81] R. Kimmel. Fast edge integration. *Geometric Level Set Methods in Imaging, Vision, and Graphics,* S. Osher and N. Paragios (Ed.), Springer, pages 59–74, 2003.

[82] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[83] G. Koepfler, C. Lopez, and J.M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM Journal on Numerical Analysis*, 31(1):282–299, 1994.

[84] H. Krim and A. Yezzi, editors. *Statistics and Analysis of Shapes*. Birkhauser Boston, 2006.

[85] T.S. Lee, D. Mumford, R. Romero, and V.A.F. Lamme. The role of the primary visual cortex in higher level vision. *Vision Research*, 38(15-16):2429–2454, 1998.

[86] A.E. Lefohn, J.M. Kniss, C.D. Hansen, and R.T. Whitaker. A streaming narrow-band algorithm: Interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):422–433, 2004.

[87] M. Leventon, O. Faugeraus, and W. Grimson. Level set based segmentation with intensity and curvature priors. In *Proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 4–11, Hilton Head Island, SC, USA, Jun. 2000.

[88] M.E. Leventon, W. Eric, L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 316–323, Hilton Head Island, SC, USA, Jun. 2000.

[89] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.

[90] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2001.

[91] J.L. Lisani, L. Moisan, P. Monasse, and J.M. Morel. On the theory of planar shape. *Multiscale Model Simulation*, 1(1):1–24, 2003.

[92] R. Malladi, J. Sethian, and B. Vemuri. Evolutionary fronts for topology independent shape modeling and recovery. In *Proceedings of European Conference on Computer Vision*, pages 3–13, Stocklholm, Sweden, May 1994.

[93] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, second edition, 1999.

[94] A. Mennucci, A. Yezzi, and G. Sundaramoorthi. Properties of Sobolev-type metrics in the space of curves. *Preprint, arXiv:math.DG/0605017*, 2006.

[95] B. Merriman, J.K. Bence, and S.J. Osher. Motion of multiple junctions: A level set approach. *Journal of Computational Physics*, 112(2):334 –363, 1994.

[96] F. Mokhtarian and A.K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[97] P. Monasse. *Morphological representation of digital images and application to registration*. PhD thesis, Ceremade, Paris, 2000.

[98] P. Monasse and F. Guichard. Scale-space from a level lines tree. In *Proceedings of International Conference on Scale-Space Theories in Computer Vision*, pages 175–186, Corfu, Greece, Sep. 1999.

[99] P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing*, 9(5):860–872, 2000.

[100] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, XLII:577–685, 1989.

[101] S.K. Nayar. Computational cameras: Redefining the image. *Computer*, 39(8):30–38, 2006.

[102] S.K. Nayar, V. Branzoi, and T.E. Boult. Programmable imaging: Towards a flexible camera. *International Journal of Computer Vision*, 70(1):7–22, 2006.

[103] Perl OpenGL. http://graphcomp.com/opengl/index.cgi?v=0111s5m1&r=s3m1.

[104] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[105] S. Osher and J. Sethian. Front propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[106] J.D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A.E. Lefohn, and T.J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

[107] Y. Pan, J.D. Birdwell, and S. M. Djouadi. Efficient bottom-up image segmentation using region competition and the Mumford-Shah model for color and textured images. In *Proceedings of International Symposium on Multimedia*, pages 376–390, San Diego, CA, USA, Dec. 2006.

[108] Y. Pan, J.D. Birdwell, and S.M. Djouadi. Image segmentation using curve evolution and anisotropic diffusion: An integrated approach. In *Proceedings of International Symposium on Multimedia 2005*, pages 387–394, Irvine, CA, USA, Dec. 2005.

[109] Y. Pan, J.D. Birdwell, and S.M. Djouadi. A new gradient and region based snake. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 745–748, Philadelphia, PA, USA, Mar. 2005.

[110] Y. Pan, J.D. Birdwell, and S.M. Djouadi. Probabilistic curve evolution using particle filters. In *Proceedings of IEEE Conference on Decision and Control and European Control Conference 2005*, pages 6335–6340, Seville, Spain, Dec. 2005.

[111] Y. Pan, J.D. Birdwell, and S.M. Djouadi. Bottom-up hierarchical image segmentation using region competition and the Mumford-Shah functional. In *Proceedings of International Conference on Pattern Recognition*, pages 117–121, Hong Kong, China, Aug. 2006.

[112] Y. Pan, J.D. Birdwell, and S.M. Djouadi. An efficient bottom-up image segmentation method based on region growing, region competition and the Mumford Shah functional. In *Proceedings of International Workshop on Multimedia Signal Processing*, pages 344–348, Victoria, BC, Canada, Oct. 2006.

[113] Y. Pan, J.D. Birdwell, and S.M. Djouadi. Efficient implementation of the Chan-Vese models without solving PDEs. In *Proceedings of International Workshop on Multimedia Signal Processing*, pages 350–353, Victoria, BC, Canada, Oct. 2006.

[114] N. Paragios and R. Deriche. Geodesic active contours for supervised texture segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages II: 422–427, Ft. Collins, CO, USA, Jun. 1999.

[115] N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. In *Proceedings of European Conference on Computer Vision*, volume II, pages 224–240, Dublin, Germany, Jun. 2000.

[116] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[117] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[118] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):564–569, 1999.

[119] M. Proesmans, E. Pauwels, and L. Gool. *Coupled Geometry-Driven Diffusion Equations for Low-level Vision*. Kluwer, 1994.

[120] W.T. Reeves. Particle systems-a technique for modeling a class of fuzzy objects. *Computer Graphics*, 17(2):359–76, 1983.

[121] R.J. Rost. *OpenGL Shading Language*. Addison Wesley, second edition, 2006.

[122] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[123] W. Rudin. *Functional Analysis*. McGraw-Hill, second edition, 1973.

[124] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, third edition, 1986.

[125] C. Sagiv, N.A. Sochen, and Y.Y. Zeevi. Integrated active contours for texture segmentation. *IEEE Transactions on Image Processing*, 1(1):1–19, 2004.

[126] C. Samson, L. Blanc-*Féraud*, G. Aubert, and J. Zerubia. A level set model for image classification. *International Journal of Computer Vision*, 40(3):187–197, 2000.

[127] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge Press, 2001.

[128] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, 1994.

[129] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.

[130] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, second edition, 1999.

[131] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proceedings of Internatioanl Conference on Computer Vision and Pattern Recognition*, pages 136–142, San Francisco, CA, USA, Jun. 1996.

[132] E. Sharon and D. Mumford. 2d shape analysis using conformal mapping. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 02, pages 350–357, Washington, DC, USA, Jun. 2004.

[133] Y. Shi and W.C. Karl. A fast level set method without solving pdes. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 97–100, Philadelphia, PA, USA, Mar. 2005.

[134] D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide*. Addison Wesley, fifth edition, 2005.

[135] K. Siddiqi, Y. Lauziere, A. Tannenbaum, and S. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Transactions on Image Processing*, 7(3):433–443, 1998.

[136] The Message Passing Interface (MPI) standard. http://www-unix.mcs.anl.gov/mpi/.

[137] The Parallel Virtual Machine standard (developed by Oak Ridge National Library). http://www.csm.ornl.gov/pvm/pvm_home.html.

[138] G. Sundaramoorthi, J.D. Jackson, A. Yezzi, and A.C. Mennucci. Tracking with sobolev active contours. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume I, pages 674– 680, New York, USA, Jun. 2006.

[139] G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Sobolev active contours. In *Proceedings of Variational and Level Set Methods in Computer Vision*, pages 109–120, Beijing, China, Oct. 2005.

[140] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998.

[141] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Computer Graphics*, 26(2):185–194, 1992.

[142] S. Teboul, L. Blanc-Féroud, G. Aubert, and M. Barlaud. Variational approach for edge-preserving regularization using coupled pdes. *IEEE Transactions on Image Processing*, 7(3):387–397, 1998.

[143] H. Tek and B.B. Kimia. Shock-based reaction-diffusion bubbles for image segmentation. *Technical Report LEMS-138*, 1994.

[144] D. Terzopoulos and R. Szeliski. *Active Vision*. MIT Press, 1992.

[145] L.N. Trefethen and III D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[146] A. Tsai, A. Yezzi, and A.S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.

[147] D. Tschumperle. C++ template image processing library. *http://cimg.sourceforge.net/*, 2004.

[148] Z. Tu, X. Chen, A.L. Yuille, and S.C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, Marr Prize Issue, 2005.

[149] G. Unal, H. Krim, and A. Yezzi. Stochastic differential equations and geometric flows. *IEEE Transactions on Image Processing*, 11(12):1405–1416, 2002.

[150] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[151] L. Wall, T. Christiansen, and R.L. Schwartz. *Programming Perl*. O'Reilly, 1996.

[152] A. Watt. *3D Computer Graphics*. Addison Wesley, third edition, 2000.

[153] Java Website. Java media framework. *http://java.sun.com/products/java-media/jmf/*, 2003.

[154] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, University of Kaiserslautern, Germany, 1996.

[155] J. Weickert and B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the perona-malik paradox. *Advances in Computer Vision,* F. Solina and W.G. Kropatsch and R. Klette and R. Bajcsy (Eds.), pages 1–10, 1997.

[156] R.T. Whitaker and S.M. Pizer. A multi-scale approach to nonuniform diffusion. *CVGIP: Image Understanding*, 57(1):99–110, 1993.

[157] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, second edition, 2003.

[158] A. Witkin. Scale-space filtering. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1019–1021, Karlsruhe, Germany, Aug. 1983.

[159] X. Xie and M. Mirmehdi. RAGS: Region-aided geometric snake. *IEEE Transactions on Image Processing*, 13(5):640–652, 2004.

[160] A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13(1/2):195 –216, 2002.

[161] Y. You and M. Kaveh. Differences in the behaviors of continuous and discrete anisotropic diffusion equations for image processing. In *Proceedings of International Conference on Image Processing*, volume 3, pages 249–253, Chicago, IL, USA, Oct. 1998.

[162] Y. You, W. Xu, A. Tannenbaum, and M. Kaveh. Behavioral analysis of anisotropic diffusion in image processing. *IEEE Transactions on Image Processing*, 5(11):1539–1553, 1996.

[163] S. C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/ MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

Vita

Yongsheng Pan received his B.S. in Underwater Acoustic Electrical Engineering in July 1999 from Harbin Engineering University, Harbin, China. He received his M.S. in Biomedical Engineering in March 2002 from Shanghai JiaoTong University, Shanghai, China. Yongsheng Pan enrolled in the doctoral program in the Department of Electrical and Computer Engineering at the University of Tennessee, Knoxville in 2003. He joined the Laboratory for Information Technologies as a graduate assistant, working on image processing under the direction of his advisor Dr. Birdwell and his coadvisor Dr. Djouadi. He will graduate with a Doctor of Philosophy in electrical engineering from the University of Tennessee in December 2007. He will work as a postdoctoral associate in the school of computing in the University of Utah under the direction of Dr. Whitaker starting January 2008.