TOPOLOGICALLY ADAPTABLE DEFORMABLE MODELS FOR MEDICAL IMAGE ANALYSIS

by

Timothy J. McInerney

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

© Copyright by Timothy J. McInerney 1997

National Library of Canada

Acquisitions and Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada

Acquisitions et services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre reference

Our file Notre reference

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-28015-2

Canada

Abstract

Topologically Adaptable Deformable Models
for
Medical Image Analysis

Timothy J. McInerney
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
1997

The increasingly important role of medical imaging in the diagnosis and treatment of disease has opened an array of challenging problems centered on the computation of accurate geometric models of anatomic structures from medical images. A promising approach to tackle such problems is the use of deformable models. These powerful models have proven to be effective in segmenting, visualizing, matching, and tracking anatomic structures by exploiting (bottom-up) constraints derived from the image data together with (top-down) *a priori* knowledge about the location, size, and shape of these structures. Furthermore, deformable models support highly intuitive interaction mechanisms that allow medical scientists and practitioners to bring their expertise to bear on the image interpretation task.

This thesis proposes, develops, and applies a new class of discrete deformable models for the segmentation and analysis of medical images. This new class of deformable models is embedded and defined in terms of an *Affine Cell Decomposition* (ACD) framework – a theoretically sound framework that significantly extends the abilities of traditional deformable models, enabling topological flexibility among other features. ACD-based deformable models can be used to extract, reconstruct and analyze even the most complex biological structures from medical images.

The ACD framework combined with a novel and original reparameterization algorithm creates a simple but elegant and powerful mechanism for multiresolution deformable curve, surface, and solid models to "flow" or "grow" into objects with complex geometries and topologies, and adapt their shape to conform to the object boundaries. Multiple instances of the models can be dynamically created or destroyed and can seamlessly split or merge to adapt to object topology. ACD-based models maintain the traditional parametric physics-based formulation of deformable models, allowing them to incorporate *a priori* anatomic knowledge in the form of energy and force-based constraints, and provide intuitive interactive capabilities. Furthermore, by defining the models in terms of the ACD framework, "hard" geometric, topological, and global shape constraints can be efficiently realized. These combined properties lead to a robust, elegant, and highly automated method of linking sparse or noisy local image features into a complete, consistent, and analytical object model.

Acknowledgements

I would like to express my sincerest gratitude to my supervisor, Demetri Terzopoulos. I have learned a great deal under his tutelage and without his guidance this thesis certainly would not have been realized. There are many qualities that make a good scientist, but I believe the qualities that are most important and that I most admire are those that Demetri possesses in abundance: dedication, diligence, creativity, and insight.

My special thanks go to Professor Nicholas Ayache for generously agreeing to serve as external appraiser of my thesis, and Professor A. N. Venetsanopoulos who acted as external examiner. Many thanks to my thesis committee members, Professors Michiel Van de Panne, James Stewart, and Geoff Hinton, who provided valuable comments and suggestions. Special thanks to Michiel Van de Panne for serving as internal appraiser.

I would like to thank fellow students from the vision lab Sean Culhane, Robert Majka, Richard Mann, Radek Grzeszczuk, Victor Lee, and former students Xiaoyuan Tu, Hong Qin, for the many discussions, ideas, advice, and their friendship. A special thanks to Sean Culhane for patiently and repeatedly answering all of my systems questions over the years. Thanks to Luiz Velho for providing his software for my perusal and to Piotr Jasiobedzki for providing the retinal image. Thanks also to Kathy Yen for cheerfully helping me with the myriad of administrative details that constantly cropped up over the years.

The CT heart images used in this thesis come from Dr. Eric Hoffman of the University of Pennsylvania Medical School and were redistributed to us courtesy of Dr. Dmitry Goldgof, CSE Dept, University of South Florida. The Canny-Deriche-Monga edge detector used in this thesis is courtesy of Gregoire Malandain, Olivier Monga and Rachid Deriche of the National Institute in Computer Science and Control Theory (INRIA), France. The MR brain image volume was provided courtesy of Dr. Christos Davatzikos of the Johns Hopkins School of Medicine. The MRA image volume was provided courtesy of Allen R. Sanderson, Henry Bushwell, and Dennis Parker of the University of Utah, Dept. of Radiology and Dept. of Comp. Science. The CT lung image volume was provided courtesy of Dr. Wayne Mitzner, Division of Physiology, Johns Hopkins University.

I am very grateful for the generous financial support I received from NSERC, the Department of Computer Science, and Demetri Terzopoulos.

Finally, I would like to thank my parents, Murray and Elaine. Their life-long interest in knowledge and learning sets an example for us all.

I dedicate this thesis to my beautiful wife, Katharine Tapscott. Her love, support, and encouragement have truly made this all possible. She is the center of my world.

Contents

1 Introduction **1**

1.1 Problem Statement 4

1.2 ACD-based Deformable Models 6

1.3 Contributions 7

1.3.1 The Affine Cell Decomposition Framework 7

1.3.2 Topologically Adaptive Deformable Models 8

1.3.3 Constraint Mechanisms 8

1.3.4 Prototype Interactive System and Applications 10

1.4 Thesis Organization 11

2 Mathematical Foundations of Deformable Models **13**

2.1 Energy-Minimizing Deformable Models 15

2.1.1 Dynamic Deformable Models 17

2.1.2 Discretization and Numerical Simulation 18

2.1.3 Deformable Surface Models 20

3 Medical Image Analysis with Deformable Models **22**

3.1 Image Segmentation with Deformable Curves 23

3.1.1 Implicit Deformable Contour Models 26

3.2 Volume Image Segmentation with Deformable Surfaces 27

3.3 Incorporating A Priori Knowledge 30

4 Topologically Adaptable Snakes **33**

4.1 Model Overview 33

4.1.1 Relationship to Front Propagation 34

4.2 Model Description 35

4.2.1 Internal Forces 36

4.2.2 External Forces 37

4.2.3 Numerical Integration 37

4.3 Simplicial Cell Decomposition 38

4.3.1 Simplicial Approximation 40

4.3.2 Iterative Reparameterization 43

4.3.3 Topological Transformations 50

4.3.4 The T-Snake Algorithm 51

4.4 Limitations of T-Snakes 52

5 T-Snakes Features and Constraints **55**

5.1 T-Snakes Features 55

5.1.1 Geometric Coverage and Intrinsic Parameterization 55

5.1.2 Topological Adaptability 57

5.1.3 Multi-resolution Capability 57

5.1.4 Conversion to/from Classical Snakes 62

5.1.5 Multiple T-Snakes 63

5.1.6 Model Element "Cooling" Process 65

5.1.7 Interactive Control 66

5.2 T-Snakes Constraints 68

5.2.1 Geometric Constraints 68

5.2.2 Statistical Constraints 70

5.2.3 Topological Constraints 73

5.2.4 Global Shape Constraints 76

6 Topologically Adaptable Surfaces **83**

6.1 Model Description 83

6.1.1 3D Simplicial Cell Decomposition 84

6.1.2 Iterative Reparameterization 86

6.1.3 Topological Transformations 89
6.2 T-Surface Features 89
6.2.1 Geometric Coverage and Topological Adaptability 90
6.2.2 Multiresolution Capability 93
6.3 T-Surface Constraints 93
6.3.1 Topological Constraints 94
6.3.2 Global Shape Constraints 97

7 Segmentation of Medical Images with T-Snakes and T-Surfaces 99

7.1 Prototype Medical Image Processing System 99
7.2 Image Segmentation with T-Snakes 101
7.2.1 Segmentation of Vertebra Phantom Cross-sections 101
7.2.2 Segmentation of Vasculature from Retinal Images 101
7.2.3 Segmentation of Brain Contours 105
7.2.4 Segmentation of the Corpus Callosum 109
7.2.5 Segmentation of Neuronal Cells 111
7.3 Volume Image Segmentation with T-Surfaces 113
7.3.1 Vertebra Phantom Segmentation and Reconstruction 113
7.3.2 Segmentation of the Left Ventricle and Aorta 115
7.3.3 LV Segmentation with T-Surfaces and Superquadrics 118
7.3.4 Segmentation of Brain Ventricles 118
7.3.5 Segmentation of Airway Structures of the Lung 121
7.3.6 Segmentation of Cerebral Vasculature 121
7.3.7 Segmentation of the Cerebral Cortex 124

8 Conclusions 126

8.1 Comparison of T-Snakes/T-Surfaces to Alternative Models 127
8.1.1 Comparison to Dynamic Particle Systems 128
8.1.2 Comparison to Level Set Evolution Techniques 128
8.2 Research Directions 129
8.2.1 Model Simplification 130

8.2.2 Robustness 130

8.2.3 Curve versus Surface versus Solid Models 130

8.2.4 Automatic Initialization, Shape Recovery and Analysis 131

A Level Set Curve Evolution **132**

B Classification Algorithm **135**

C Superquadric Cells and Cell Fitting **137**

C.1 Dynamic Deformable Superquadric Cells 138

C.2 Point-to-point Cell Constraints 139

C.3 Cell Blending 139

Bibliography **140**

List of Figures

1.1 (a) MR image slice of the brain. (b) Cells in an EM photomicrograph. (c) Ultrasound image of fetus. (d) Portion of a fluorescent angiogram of the retina. (e) CT image slice of canine heart. (f) PET image slice of brain. 2

2.1 Snake (white) attracted to cell membrane in an EM photomicrograph [12] . 14

2.2 Snake deforming towards high gradients in a processed cardiac image, influenced by "pin" points and an interactive "spring" which pulls the contour towards an edge [61] 18

2.3 Closed deformable surface being pulled by a spring force showing the effect of various α_{ij} and β_{ij} settings [61] : (a) $\alpha_{ij} = 0.8$, $\beta_{ij} = 0$. (b) $\alpha_{ij} = \beta_{ij} = 0.5$. (c) $\alpha_{ij} = 0$, $\beta_{ij} = 0.8$. 21

3.1 (a) Intensity CT image slice of LV. (b) Edge detected image. (c) Initial snake. (d)-(f) Snake deforming towards LV boundary, driven by "inflation" force. [61] 24

3.2 (a) Deformable "balloon" model embedded in an edge detected CT image volume deforming towards LV edges. Three orthogonal slices of the volume are shown. (b) Reconstruction of LV [61] 29

4.1 Freudenthal triangulation. 39

4.2 A 3-simplex (far right) and examples of its $k-1$-dimensional faces, where (from left to right) $k = 0, 1$, and 2 respectively. 40

4.3 Simplex classification. 41

4.4 Simplicial approximation (dashed-line) of an object contour (solid-line) using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded. 42

4.5 Phase 1 of T-snake reparameterization: (a) T-snake expands and moves off grid during deformation step, (b) new model nodes are computed, (c) new T-snake nodes and elements. 44

4.6 Phase 2 of T-snake reparameterization: (a) a model element of an expanding T-snake may pass over a grid vertex, (b) subspace partitioning (c) new grid vertices (light-shaded), (d) new T-snake. 46

4.7 Formation of subspaces used in the labeling of a grid vertex. 47

4.8 Classification combinations. 49

4.9 Examples of T-snake topological transformations: (a) self-intersection, (b) shrinking and splitting, (c) merging. The resulting T-snake(s) after the transformations are shown as the dotted line(s). Node reconnections occur automatically in the shaded triangles so that inside and outside grid vertices are separated by a model element. 50

4.10 J_1 triangulation. 53

5.1 T-snake flowing into object with protrusions. 56

5.2 T-snake flowing along a spiral tube-like object. 57

5.3 Initialization insensitivity of T-snakes. A T-snake may be seeded essentially anywhere within or surrounding an object. 58

5.4 T-snake flowing and splitting to segment object with complex topology. . 59

5.5 T-snake shrinking, wrapping and splitting around multiple objects. . . . 60

5.6 T-snake representation of circular object using 3 grid resolutions: (a) 4 × 4 (b) 8 × (8 c) 16 × 16. 61

5.7 T-snake nodes on coarse-resolution grid are preserved on finer resolution grid. 61

5.8 Redistribution of snake nodes after discarding simplicial grid. Grid resolution: (a) 4 × 4 (b) 8 × 8 (c) 16 × 16. 62

5.9 Segmentation of the blood vessels in angiogram of retina using using multiple T-snakes. 63

5.10 Collision detection and avoidance using model interior region "identifiers". A T-snake cannot move into "territory" already occupied by another T-snake. 64

5.11 Two models colliding. detecting and avoiding intersection. 64

5.12 Reconstruction of preprocessed anatomical regions of the brain using multiple models. The collision detection feature was used as an extra constraint when reconstructing regions 1 and 3. 65

5.13 T-snakes seeds growing. shrinking. merging. splitting and disappearing to automaticlly recover vertebral phantom parts. 66

5.14 T-snake flowing along a spiral tube-like object. The active part of the T-snake is in white. the inactive part is black. 67

5.15 T-snake flowing along retinal artery is "pulled" into smaller branch by user. 68

5.16 Example of geometric "hard" fixed point constraints. 69

5.17 CT image slice of LV and edge-detected version. 70

5.18 T-snake segmenting LV image slice using statistically-weighted inflation force and image edge forces. 71

5.19 Close-up view of LV segmentation: (a) image intensity slice. (b) edge-detected image slice. 72

5.20 Segmentation of LV without incorporating image intensity statistics. . . . 72

5.21 At least 6 T-snake model elements are needed to contain a grid vertex. . 74

5.22 Example of topology-preserving deformation. In (a). the model elements in triangles A, B, C, F form a contiguous section of the T-snake and cannot cause a topology change when the grid vertex (light-shaded square) is burned. In (b). if element A. for example. burns the grid vertex. a topology change will occur. forming two T-snakes (one inner T-snake and one outer). 75

5.23 Example used to illustrate the time dependency of the model element validation proces. During one deformation step, element A may validly burn vertex 1 and element B may validly burn vertex 2. A topology change will occur resulting in two T-snakes, one inner surrounding vertices 3 and 4, and one outer T-snake. 76

5.24 Example of T-snake topological constraint. The T-snake flows around the donut-shaped object recovering its shape. It then collides with itself and is unable to interpenetrate. 77

5.25 A close-up view of the self-collision zone. In (b) the same example is repeated with a finer resolution grid. 78

5.26 Examples demonstrating T-snake topological constraint. 78

5.27 Superquadric "cells" fitted to tubular-shaped object. 79

5.28 (a),(b) Superquadric cell growing and fitting to section of object. (c) Parametric "skin" contour of cell deforming away from cell. Maximum displacement regions are used as seed sites for "child" cells. 80

5.29 Superquadric cells growing and "splitting" to capture coarse shape of object. 80

5.30 (a) Superquadric cells blended to form initial T-snake. (b) Final fitted T-snake (c) Fitted T-snake and underlying cells. 81

6.1 (a) 3D Freudenthal triangulation. (b) Cubic cell division into 6 tetrahedra: $\rho_0 = (p_0, p_1, p_3, p_7)$, $\rho_1 = (p_0, p_1, p_5, p_7)$, $\rho_2 = (p_0, p_2, p_3, p_7)$, $\rho_3 = (p_0, p_2, p_6, p_7)$, $\rho_4 = (p_0, p_4, p_5, p_7)$, $\rho_5 = (p_0, p_4, p_6, p_7)$. 84

6.2 Intersection of object boundary with grid tetrahedra. Either one triangle is generated or one quadrilateral which can be divided into two triangles. The "signs" of the tetrahedra vertices indicate whether the vertex is inside or outside the object. 85

6.3 Approximation of a sphere using a 4 × 4 × 4 grid. 85

6.4 Formation of subspaces used in the classification algorithm. (a) Three planes are formed by joining the two nodes of a model triangle edge with a grid vertex. (b) The three planes partition space into 8 subspaces. . . . 86

6.5 (a) A model triangle moves to a new position during a deformation step and potentially burns a grid vertex. (b) The 8-sided polyhedron formed from the model triangle at current and new positions. (c) The 8 subspaces used to classify a model triangle at its new position and determine if the grid vertex was burned. 87

6.6 T-surface flowing into an object with a bifurcation. The dark shaded regions are "frozen" and have been removed from the computation. . . . 90

6.7 T-surface deforming and merging with itself to recover toroidal object. . 91

6.8 T-surface recovering shape of object with complex topology. 91

6.9 T-surface shrinking, wrapping, and splitting to recover interlocked ring objects. 92

6.10 Approximation of a sphere using (a) $4 \times 4 \times 4$ grid and (b) $8 \times 8 \times 8$ grid. 93

6.11 Conversion of T-surface sphere approximation to classical deformable surface model using (a) $4 \times 4 \times 4$ grid and (b) $8 \times 8 \times 8$ grid. Note the redistribution of model nodes from the action of the internal forces. . . . 93

6.12 T-surface flowing around torus. The topology constraint prevents the model from merging with itself. It collides but cannot not penetrate. . . 95

6.13 Cross-sectional slices of the fitted T-surface. 95

6.14 T-surface fit to object with complex topology. The topology constraint prevents the T-surface from merging with itself. (b) Cross-section of the fitted model illustrating that no topology change has taken place. 96

6.15 T-surface shrinking and wrapping around torus. The topology constraints prevents the T-surface from splitting. 96

6.16 Superquadric cells growing and splitting to recover object with bifurcation. 97

6.17 (a) Result of blend of superquadric cells to form initial T-surface. (b) Final fit of T-surface. 98

7.1 Segmentation of two cross sectional images of a human vertebra phantom. 101

7.2 Segmentation of the blood vessels in angiogram of retina. The top row is an image sequence showing a snake flowing and branching along a vessel. 102

7.3 Superquadric cells growing and splitting to recover vessels in clipped retinal image. 104

7.4 The result of the cell growing-splitting process. 104

7.5 (a) Superquadric cells blended to form initial T-snake. (b) Final T-snake segmentation. (c) T-snake with underlying cells. 105

7.6 T-snake used to segment gray-matter/white-matter interface in MR brain image slice. 106

7.7 (a) Closeup of T-snake fit. (b) Fitted T-snake (black) overlaid on manually segmented image. 106

7.8 T-snake shrinking and segmenting CSF/gray matter interface in MR brain image slice. 107

7.9 (a) Closeup of T-snake segmentation. (b) Fitted T-snake overlaid on manually segmented image. 108

7.10 Closeup of fitted T-snake near narrow cavity. Topological constraints prevent T-snake self-intersection. 108

7.11 (a) T-snake fitted to second MR brain image slice. (b) Fitted T-snake overlaid on manually segmented image. 109

7.12 Superquadric cells growing and splitting to recover coarse shape of corpus callosum from MR brain image slice. 110

7.13 Final fitted superquadric cells. 110

7.14 (a) Blended superquadric cells forming initial T-snake. (b) Final fitted T-snake. (c) Example of T-snake segmentation without global shape constraint. 111

7.15 T-snake (derived from closed polygon) used to segment neuronal cells. The T-snake is constrained to pass through the initial user-specified points. . 112

7.16 (a) (b) T-snake formed from user-specified closed polygon deforms to segment neuronal cell. (c) Soft point constraints are used to pull T-snake off spurious edges. 112

7.17 T-surface segmenting vertebra phantom from CT image volume. 113

7.18 Two views of the fitted T-surface. 114

7.19 Several cross-sections of T-surface overlaid on CT vertebra image volume slices. 114
7.20 CT sagittal image slice of canine heart. 115
7.21 T-surface segmenting LV and aorta from CT image volume. 116
7.22 Final fitted T-surface and four cross-sections of the model. 117
7.23 Superquadric cells growing and splitting to segment coarse shape of LV and part of aorta. 119
7.24 (f) Blend of superquadric cells to form initial T-surface. (g) Final fitted T-surface. 119
7.25 T-surface segmentation of ventricles from MR image volume of the brain. 120
7.26 T-surface segmentation of the airway structures of a canine lung from HRCT image volume. Several cross-sections of the fitted model are also shown. 122
7.27 T-surface segmentation of cerebral vasculature from MRA image volume. 123
7.28 T-surface segmentation of the cerebral cortex from MR image volume. (a) front view, (b) top view, (c) side view. 125

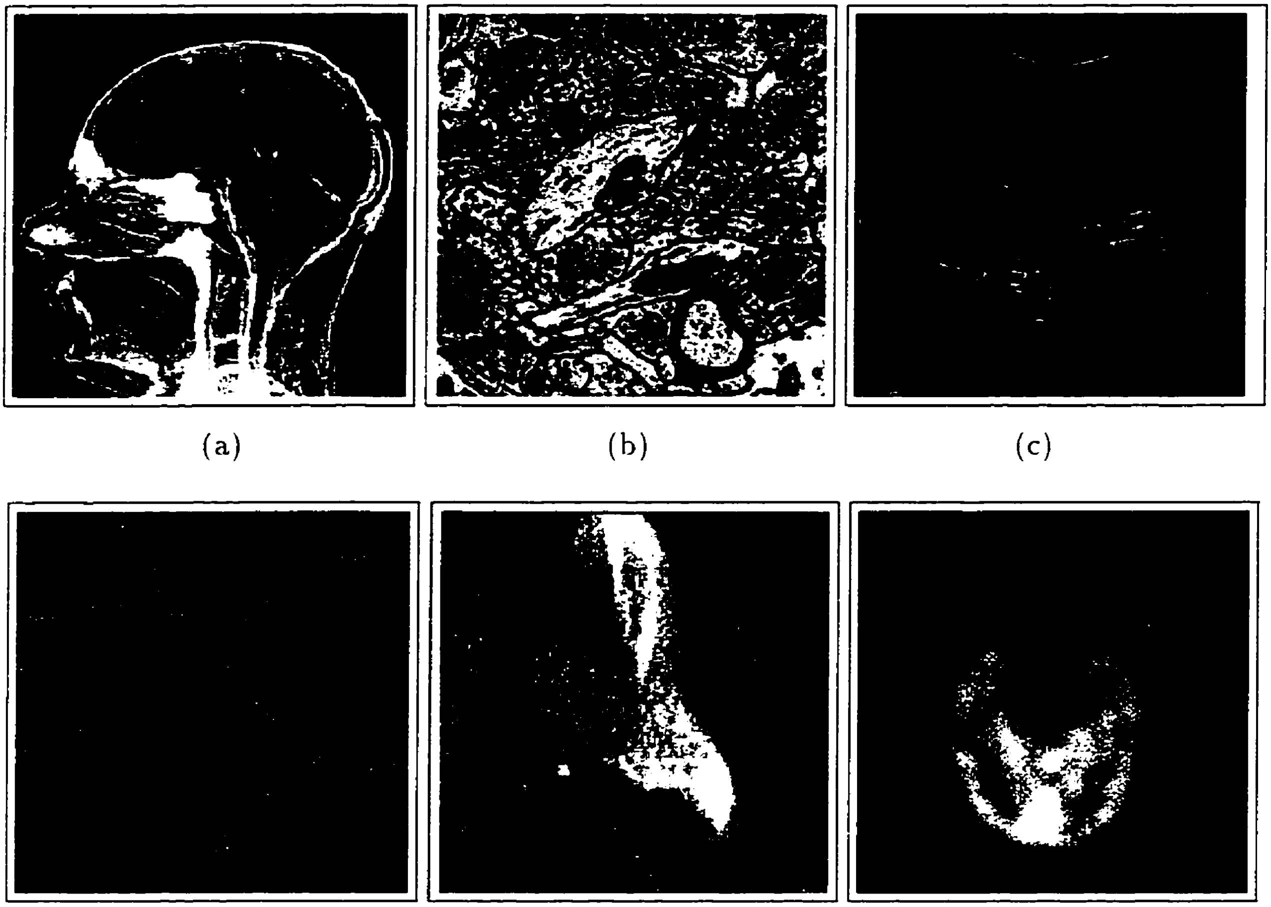
Chapter 1

Introduction

The rapid development and proliferation of medical imaging technologies is revolutionizing medicine. Medical imaging allows scientists and physicians to glean potentially life-saving information by peering noninvasively into the human body. In recent years, the role of medical imaging has expanded beyond the simple visualization and inspection of anatomic structures. It has become a tool for surgical planning and simulation, intra-operative navigation, radiotherapy planning, and for tracking the progress of disease. For example, ascertaining the detailed shape and organization of anatomic structures enables a surgeon to preoperatively plan an optimal approach to some target structure. In radiotherapy, medical imaging allows the delivery of a necrotic dose of radiation to a tumor with minimal collateral damage to healthy tissue.

With medical imaging playing an increasingly prominent role in the diagnosis and treatment of disease, the medical image analysis community has become preoccupied with the challenging problem of extracting—with the assistance of computers—clinically useful information about anatomic structures imaged through CT, MR, PET, and other modalities. Although modern imaging devices provide exceptional views of internal anatomy (Figure 1.1), the use of computers to quantify and analyze the embedded structures with accuracy and efficiency is limited. Accurate, reproducible, quantitative data must be efficiently extracted in order to support the spectrum of biomedical investigations and clinical activities from diagnosis, to radiotherapy, to surgery.

Segmenting structures from medical images and reconstructing a compact analytic



(a) (b) (c)

(d) (e) (f)

Figure 1.1: (a) MR image slice of the brain. (b) Cells in an EM photomicrograph. (c) Ultrasound image of fetus. (d) Portion of a fluorescent angiogram of the retina. (e) CT image slice of canine heart. (f) PET image slice of brain.

representation of these structures is difficult due to the sheer size of the data sets and the complexity and variability of the anatomic shapes of interest. Furthermore, the shortcomings typical of sampled data, such as sampling artifacts, spatial aliasing, and noise, may cause the boundaries of structures to be indistinct and disconnected. The challenge is to extract boundary elements belonging to the same structure and integrate these elements into a complete and consistent model of the structure. Traditional low-level image processing techniques which consider only local information can make incorrect assumptions during this integration process and generate infeasible object boundaries. As a result, these model-free techniques usually require considerable amounts of expert intervention. Furthermore, the subsequent analysis and interpretation of the segmented objects is hindered by the pixel- or voxel-level structure representations generated by most image processing operations.

A promising and vigorously researched computer-assisted medical image analysis technique is the use of *deformable models* (see the recent survey [64]). Among model-based techniques, deformable models offer a unique and powerful approach to image analysis that combines geometry, physics, and approximation theory. These continuous geometric models consider an object boundary as a single connected structure. Connectivity of the object is therefore guaranteed. Deformable models can make use of *a priori* knowledge of object shape to constrain the segmentation problem and the inherent continuity and smoothness of the models can compensate for noise, gaps and other irregularities in object boundaries. Furthermore, the parametric representations of the models can provide a compact, analytical description of object shape. Finally, deformable models support highly intuitive interaction mechanisms that, when necessary, allow medical scientists and practitioners to bring their expertise to bear on the model-based image interpretation task. These combined properties lead to a robust and elegant technique for linking sparse or noisy local image features into a complete well-defined model.

This thesis proposes, develops, and applies a new class of deformable models for the segmentation and analysis of medical images. This new class of deformable models exploits an *Affine Cell Decomposition* (ACD) of the image domain — a theoretically sound framework that significantly extends the abilities of classical deformable models such as deformable contours or *snakes* [47] and deformable surfaces [99, 105]. Affine cell decompositions divide an image domain into a collection of convex polytopes. Embedding deformable models in this framework allows the models to extract and reconstruct even the most complex biological structures. The ACD framework combined with a novel reparameterization algorithm creates a simple but elegant and powerful mechanism for multiresolution deformable curve, surface, and solid models to "flow" or "grow" into objects with complex geometries and topologies, adapting their shape to conform to the object boundaries. Model shape adaptation or *deformation* is governed by a physics-based formulation. The ACD framework enables the models to maintain the traditional properties associated with classical deformable models, such as user interaction and the incorporation of constraints through energy functions or force functions, while overcoming many of their limitations. The ACD framework also provides a convenient mechanism for the incorporation of "hard" geometric, topological and global shape constraints. These combined properties produce an effective technique for the efficient, accurate, reproducible and highly automated extraction and analysis of anatomic structures from medical images.

1.1 Problem Statement

The segmentation of anatomic structures—the partitioning of the original set of image points into subsets corresponding to the structures—is an essential first stage of most medical image analysis tasks, such as shape analysis, visualization, registration, labeling, and motion tracking. These tasks usually require anatomic structures in the original image to be reduced to a compact, analytic representation of their shapes. A primary example is the segmentation of the heart, especially the left ventricle (LV), from cardiac imagery. Segmentation and reconstruction of the left ventricle is a prerequisite for computing diagnostic information such as ejection-fraction ratio, ventricular volume ratio, heart output, and for wall motion analysis which provides information on wall thickening, etc. [91].

The manual segmentation of medical images can be extremely labor intensive and time-consuming. Consequently, semi-automatic and, ultimately, fully-automatic techniques are a desirable goal. Increasing the degree of automation can not only relieve clinicians from the labor intensive aspects of their work, but it can also increase the efficiency, accuracy and perhaps most importantly, reproducibility of the segmentation. On the other hand, since erroneous interpretations of medical images are unacceptable, any successful segmentation technique should support intuitive, efficient interactive guidance or editing by the medical expert.

With this motivation in mind, our goal is to considerably extend the capabilities of classical deformable models such as snakes, improving their performance and increasing their degree of automation, while retaining all of their traditional strengths. An important property to maintain is the ability to design energy functions or force functions to constrain and interactively guide the model; in other words, the ability to incorporate or apply *a priori* knowledge. The most significant limitations to overcome are the following:

- Snakes and other deformable models are sensitive to their initial conditions. Classical snakes were designed as interactive models and usually must be placed close to the boundary of the target object to guarantee good performance. The external energy functionals used in the variational model formulations are typically nonconvex with multiple local minima. Furthermore, gradient descent algorithms, which do not guarantee global solutions, are the most common method used for energy minimization. As a result, snakes can settle into local minimum solutions that do not represent the true boundary of the object.

- The topology of the object of interest must be known in advance since traditional deformable models are parametric and are incapable of topological transformations without additional machinery.

- The internal energy constraints of snakes, arc-length and curvature minimization, can limit their geometric flexibility and prevent them from representing long tube-like shapes or shapes with significant protrusions. The energy function of most deformable models depends on its parameterization and is not directly related to the geometry of the object. New parameterizations can change the model energy in arbitrary ways.

- Deformable models can be attracted to spurious image features during the energy minimization. This is essentially due to the local nature of boundary-based models such as snakes. Furthermore, even if global energy minimization schemes are used, there may be no relationship between the equilibrium of a snake and the real boundaries of the object. Finally, it is difficult to perform a global analysis of object shape using local boundary-based models. The incorporation of *a priori* knowledge of global object shape is essential for robust segmentation and analysis of noisy medical images.

These combined factors may ultimately limit the efficiency and degree of automation that can be achieved by classical deformable models for the segmentation and analysis of medical images.

1.2 ACD-based Deformable Models

Most classical deformable models are parametric models whose parameterization is defined initially and does not change automatically throughout the deformation process. If the topology of an object is fixed and known *a priori* and the geometry is simple (i.e. relatively compact and convex), such models may be the most appropriate since they will provide greater constraint. Implicit models on the other hand, such as the formulation proposed in [13, 14, 59], provide topological and geometric flexibility through their level set representation. They are best suited to the recovery of objects with complex shapes and unknown or complex topologies. Unfortunately, implicit models are not as convenient as parametric models in terms of mathematical formulation and for accommodating interaction or guidance by expert users and high-level control mechanisms.

In this thesis, we develop a parametric deformable model that has the power of an implicit formulation by using a superposed *affine cell complex* or grid to automatically and efficiently reparameterize the model during the deformation process. That is, we embed the traditional parametric snakes model within the framework of affine cell domain decomposition — a mathematically sound decomposition method based on the theory of algebraic topology. This framework enables our models to efficiently and effectively fulfill our goal of maintaining the traditional properties associated with snakes, such as user interaction and the incorporation of constraints through energy functions or force functions, while overcoming many of the limitations described in the previous section. Using the grid to iteratively reparameterize the model allows it to flow into complex shapes and change its topology as necessary. Multiple connected components of the model can be dynamically created or destroyed, or can seamlessly split or merge. In the following chapters of this thesis, we describe these features of ACD-based deformable models and many others that serve to demonstrate the power of this approach.

1.3 Contributions

Deformable models offer a flexible and powerful approach to medical image analysis. Nevertheless, deformable models suffer from several limitations that prevent their application to the full range of medical image analysis problems and that limit their potential for automation. We propose ACD-based deformable models as a solution to many of these limitations. As we will demonstrate in this thesis, this new class of deformable models can be used to segment, reconstruct and analyze almost any anatomic structure from images generated by almost any imaging modality. Furthermore, when combined with various constraint mechanisms, the models are capable of developing into a completely automatic image analysis technique. The contributions of the research are detailed in the following sections.

1.3.1 The Affine Cell Decomposition Framework

- We cast classical deformable models within the mathematically robust framework of domain decomposition via affine cells to yield a powerful and computationally efficient discrete geometric approach to topologically adaptable deformable models. The 2D and 3D models developed within the framework have all of the capabilities of an implicit formulation combined with the advantages of parametric models. The models are capable of segmenting and reconstructing objects with complex shapes and topologies. The framework enables our models to maintain the traditional properties associated with classical deformable models. such as user interaction and the incorporation of data-derived constraints. while overcoming many of the limitations previously described.

- The affine cell grid provides an elegant mechanism for automatic model reparameterization. allowing our models to segment and reconstruct objects with significant protrusions. tube-like objects. or objects with bifurcations. Furthermore. the grid produces model parameterizations in terms of the intrinsic local geometry of an object: there is no explicit parameterization sensitivity. Finally. the automatic and intrinsic reparameterization makes the model (relatively) insensitive to initial placement. significantly improving the efficiency and automation of the segmentation process.

1.3.2 Topologically Adaptive Deformable Models

We develop topologically adaptive snakes (T-snakes) [62. 63] and topologically adaptive deformable surfaces (T-surfaces) [66. 65]. The affine cell grid provides a mathematically sound framework for robust topological transformations. This property allows our models to seamlessly split or merge. adapting to the topology of a target object. Topological adaptability combined with broad geometric coverage significantly increases model automation. Furthermore. topological transformations are performed by efficiently tracking and recording the interior region of the models as they flow into an object. This space partitioning property provides a useful mechanism to map object topology and structure. as well as to perform interior region analysis.

1.3.3 Constraint Mechanisms

- Geometric Constraints

  Model point constraints and other geometric constraints can be easily incorporated into ACD-based deformable models either as soft constraints to be satisfied approximately or as hard constraints that must never be violated. The advantage of retaining a parametric model formulation as opposed to an implicit model formulation is that any data constraint expressed as an energy function or force function can easily be incorporated into the physics-based framework.

- Topological Constraints

  We develop a technique for imposing a global topological constraint on our ACD-based deformable models. In particular, the models maintain the topology of a sphere by evolving according to a sequence of topology-preserving deformations. This constraint enables ACD-based models to be insensitive to breaks in the object around boundaries or near narrow cavities due to noise and other imaging artifacts. This property is important in applications such as registration or labeling using a deformable anatomical atlas. Furthermore, multiple models can be simultaneously evolved under the ACD framework. The affine cell grid allows for extremely efficient collision detection and avoidance between models, again providing a potentially valuable constraint for the construction of deformable anatomical atlas models.

- Statistical Constraints

  Using the ACD framework, model evolution is performed by tracking and recording the interior region of a model as it expands or contracts under the influence of pressure forces. The strength of these pressure forces can be directly linked to the local or global statistics of the image pixel intensity values of the target object. These statistical constraints allow ACD-based models to behave as active region growing models that can effectively integrate edge information with region-based information.

- Global Shape Constraints

  We develop a technique for imposing global shape constraints on ACD-based deformable models. We use multiple superquadric shape templates that are constrained to overlap. The implicitly defined superquadric "cells" are fitted to anatomic structures using a "cell growing and splitting" procedure. The superquadrics capture the gross shape of an object and *a priori* anatomic knowledge is used to prevent the generation of infeasible shapes. Using the affine cell grid, an implicit "skin" is then wrapped around the superquadrics and acts as the initial placement of an ACD-based deformable model. The resulting closed contour or surface model is then deformed as usual to capture the shape details of the object. We can also constrain the skin, if necessary, to remain close to the underlying superquadric templates. The global shape constraints provided by the superquadrics combined with ACD-based deformable models significantly increase the potential for fully automatic medical image segmentation and analysis. For example, the deformable skin and underlying superquadrics provide a mechanism for the automatic detection of shape abnormalities.

1.3.4 Prototype Interactive System and Applications

We implement a prototype medical image analysis system that utilizes 2D and 3D ACD-based deformable models. We have used the system to segment and reconstruct a wide variety of anatomic structures with complex shapes and topologies. We use a discrete, computationally efficient implementation that incorporates a model element "cooling" process. When the "temperature" of an element falls below a "freezing point", the element is removed from the computation. This adjustable mechanism allows the system to maintain a small, manageable computational burden in many segmentation scenarios. Furthermore, we compute the vertices, edges and cells of the grid as they are needed during the evolution of the model. Consequently, the only memory requirement for the grid is one bit for each grid vertex. This "virtual" grid scheme allows us to use pixel or even subpixel resolution grids, if necessary, without incurring significant memory or computational overhead. Finally, the evolution of ACD-based deformable models is implemented using a naturally parallel geometric algorithm and we expect further significant performance gains on parallel machines.

1.4 Thesis Organization

The organization of the remainder of the thesis is as follows:

In Chapter 2 we review the basic mathematical formulation of classical deformable models. In particular, we provide a detailed description of the snakes formulation in order to illustrate with a concrete example the basic mathematical machinery that is present in many deformable models. We end the chapter with a description of the 3D generalization of snakes.

In Chapter 3 we survey the application of deformable models to two fundamental medical image analysis problems: segmentation and shape representation. We describe many of the deformable curve and surface models that have been developed and provide a brief comparison between parametric deformable models and implicit deformable models. We conclude the chapter with a survey of various mechanisms for incorporating *a priori* knowledge into a deformable model framework.

In Chapter 4 we present topologically adaptable snakes (T-snakes). We introduce the idea of using an affine cell decomposition of an image domain to iteratively reparameterize a deformable contour model and to perform topological transformations. We begin by describing the relationship of ACD-based deformable models to front propagation or curve evolution. We describe T-snakes in detail, including an overview of the theory of affine cell decomposition using a class of affine cells known as *simplicial* cells. We then provide a detailed description of the T-snake reparameterization (iterative simplicial approximation) process.

In Chapter 5 we use T-snakes to describe and demonstrate several properties of ACD-based deformable models. We then describe, with examples, the constraint mechanisms that can be imposed on these models. Geometric constraints, topological constraints, statistical constraints, and global shape constraints are presented.

In Chapter 6 we present topologically adaptable deformable surfaces (T-surfaces): a deformable closed-surface model that is a 3D generalization of topologically adaptable snakes. We extend simplicial cell decomposition to three dimensions using tetrahedra and demonstrate the fundamental similarity of the 2D and 3D algorithms.

In Chapter 7 we describe a prototype system for segmenting and reconstructing anatomic structures using T-snakes and T-surfaces and present a wide range of application examples. The real-world examples serve to demonstrate the power and flexibility of the simplicial cell decomposition framework and illustrate the properties and constraint mechanisms of the model described in earlier chapters.

Chapter 8 summarizes the thesis and suggests future research directions. We also present a comparison of ACD-based deformable models with alternative models in order to underscore the viability of our approach.

Appendix A outlines the basic mathematical theory of front propagation or curve evolution and its adaptation to the problems of shape recovery.

Appendix B provides a geometric proof of the robustness of the classification algorithm that underlies the ability of ACD-based deformable models to flow into objects with complex shapes and topologies.

Appendix C provides a detailed description of superquadrics as well as superquadric data-fitting and blending.

Chapter 2

Mathematical Foundations of Deformable Models

The mathematical foundations of deformable models represent the confluence of geometry, physics, and approximation theory. Geometry serves to represent object shape, physics imposes constraints on how the shape may vary over space and time, and optimal approximation theory provides the formal underpinnings of mechanisms for fitting the models to measured data.

Deformable model geometry usually permits broad shape coverage by employing geometric representations that involve many degrees of freedom, such as splines. The model remains manageable, however, because the degrees of freedom are generally not permitted to evolve independently, but are governed by physical principles that bestow intuitively meaningful behavior upon the geometric substrate. The name "deformable models" stems primarily from the use of elasticity theory at the physical level, generally within a Lagrangian dynamics setting. The physical interpretation views deformable models as elastic bodies which respond naturally to applied forces and constraints. Typically, deformation energy functions defined in terms of the geometric degrees of freedom are associated with the deformable model. The energy grows monotonically as the model deforms away from a specified natural or "rest shape" and often includes terms that constrain the smoothness or symmetry of the model. In the Lagrangian setting, the deformation energy gives rise to elastic forces internal to the model. Taking a physics-based

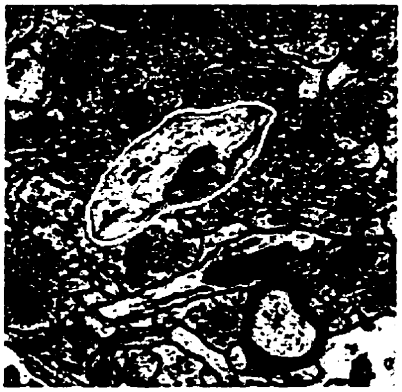


Figure 2.1: Snake (white) attracted to cell membrane in an EM photomicrograph [12].

view of classical optimal approximation, external potential energy functions are defined in terms of the data of interest to which the model is to be fitted. These potential energies give rise to external forces which deform the model such that it maximally fits the data.

Deformable curve, surface, and solid models gained popularity after they were proposed for use in computer vision [105] and computer graphics [101] in the mid 1980's. In computer vision, deformable curve and surface models were proposed as solutions to ill-posed inverse problems such as edge detection and surface reconstruction. Terzopoulos introduced the theory of continuous (multidimensional) deformable models in a Lagrangian dynamics setting [98], based on deformation energies in the form of (controlled-continuity) generalized splines [99]. The controlled-continuity spline is a generalization of a Tikhonov stabilizer [106], and can formally be regarded as *regularizing* [79] the ill-posed problems, recasting them as well-posed functional minimization problems. Early ancestors of the deformable models now in common use include Fischler and Elshlager's spring-loaded templates [33] and Widrow's rubber mask technique [116].

The deformable model that has attracted the most attention to date is popularly known as "snakes" [47]. Snakes or "deformable contour models" represent a special case of the general multidimensional deformable model theory [98]. We will review their simple formulation in the remainder of this chapter in order to illustrate with a concrete example the basic mathematical machinery that is present in many deformable models. We will end the chapter with a brief description of the deformable surface model formulation.

Snakes are planar deformable contours that are useful in several image analysis tasks. They are often used to approximate the locations and shapes of object boundaries in images based on the reasonable assumption that boundaries are piecewise continuous or smooth (Fig. 2.1). In its basic form, the mathematical formulation of snakes draws from the theory of optimal approximation involving functionals [113].

2.1 Energy-Minimizing Deformable Models

Geometrically, a snake is a parametric contour embedded in the image plane $(x, y) \in \Re^2$. The contour is represented as $\mathbf{v}(s) = (x(s), y(s))^T$, where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain. Boundary conditions may be used to specify the overall topology of the contour. For example, applying a periodic boundary condition, $v(0) = v(1)$, produces a closed snake.

The shape of the contour subject to an image $I(x, y)$ is dictated by the functional

$$\mathcal{E}(\mathbf{v}) = \mathcal{S}(\mathbf{v}) + \mathcal{P}(\mathbf{v}). \tag{2.1}$$

The functional can be viewed as a representation of the energy of the contour and the final shape and position of the contour corresponds to the minimum of this energy. The first term of the functional,

$$\mathcal{S}(\mathbf{v}) = \int_0^1 w_1(s) \left| \frac{\partial \mathbf{v}}{\partial s} \right|^2 + w_2(s) \left| \frac{\partial^2 \mathbf{v}}{\partial s^2} \right|^2 ds, \tag{2.2}$$

is the internal deformation energy. It characterizes the deformation of an elastic contour. Two physical parameter functions dictate the simulated physical characteristics of the contour: $w_1(s)$ controls the "tension" of the contour while $w_2(s)$ controls its "rigidity".[1]

We note here that the internal deformation energy defined above is a linearized version of a more general, nonlinear planar curve strain energy:

$$\mathcal{S}(\mathbf{v}) = \int_0^1 w_1(s) s^2 + w_2(s) \kappa^2 ds, \tag{2.3}$$

---

[1] The values of the non-negative functions $w_1(s)$ and $w_2(s)$ determine the extent to which the snake can stretch or bend at any point s on the snake. For example, increasing the magnitude of $w_1(s)$ increases the "tension" and tends to eliminate extraneous loops and ripples by reducing the length of the snake. Increasing $w_2(s)$ increases the bending "rigidity" of the snake and tends to make the snake smoother and less flexible. Setting the value of one or both of these functions to zero at a point s permits discontinuities in the contour at s.

where s and κ represent the arc length and curvature respectively. The linearized functional (2.2) approximates (2.3) for small deformations near the actual minimum (where higher order terms tend to 0) but is well behaved for large deformations and its quadratic form leads to significant computational benefits. In Chapter 4, we will describe our topologically adaptable snakes formulation that uses a discrete approximation to (2.3).

The second term in (2.1) consists of external energy potentials that couple the snake to the image and support user interaction. Traditionally,

$$\mathcal{P}(\mathbf{v}) = \int_0^1 P(\mathbf{v}(s))ds, \tag{2.4}$$

where $P(x,y)$ denotes a scalar potential function defined on the image plane. To couple snakes to images, external potentials are designed whose local minima coincide with intensity extrema, edges, and other image features of interest. For example, the contour will be attracted to intensity edges in an image $I(x,y)$ by choosing a potential

$$P(x,y) = -c|\nabla[G_\sigma * I(x,y)]|, \tag{2.5}$$

where c controls the magnitude of the potential, ∇ is the gradient operator, and $G_\sigma * I$ denotes the image convolved with a (Gaussian) smoothing filter whose characteristic width σ controls the spatial extent of the local minima of P.

It medical image analysis, it is essential for a user to be able to interactively control the segmentation process. Few medical images lend themselves to fully automatic processing with satisfactory results. Furthermore, the equilibrium state of the snake will, in general, represent only one local minimum solution out of a large set of possible local minima. The user must have the ability to pull the snake out of one local minimum into another. This ability is conveniently realized in the energy minimization framework through the use of user defined external constraint potentials such as interactive springs, anchored springs, and "volcanos" [47]. For example, the snake can be pulled in the direction of the mouse cursor location $(x_{\text{mouse}}, y_{\text{mouse}})$ by choosing a spring potential $P(x,y) = c((x - x_{\text{mouse}})^2 + (y - y_{\text{mouse}})^2)$, where c controls the strength of the spring. Points on the snake that are affected by the spring can be restricted to a small section of the snake closest to the mouse.

The combination of external image potentials and external constraint potentials can create a wide range of snake behavior, allowing snakes to extract and represent a broad spectrum of shapes. Furthermore, external constraint potentials are an effective, flexible means from which high-level control mechanisms can guide shape recovery, forming a sound basis for fully automatic image analysis.

In accordance with the calculus of variations, the contour $\mathbf{v}(s)$ which minimizes the energy $\mathcal{E}(\mathbf{v})$ must satisfy the Euler-Lagrange equation

$$-\frac{\partial}{\partial s}\left(w_1\frac{\partial \mathbf{v}}{\partial s}\right)+\frac{\partial^2}{\partial s^2}\left(w_2\frac{\partial^2 \mathbf{v}}{\partial s^2}\right)+\nabla P(\mathbf{v}(s,t))=\mathbf{0}. \qquad (2.6)$$

This vector-valued partial differential equation expresses the balance of internal and external forces when the contour rests at equilibrium. The first two terms represent the internal stretching and bending forces, respectively, while the third term represents the external forces that couple the snake to the image data. The usual approach to solving (2.6) is through the application of numerical algorithms (Sec. 2.1.2).

2.1.1 Dynamic Deformable Models

While it is natural to view energy minimization as a static problem, a potent approach to computing the local minima of a functional such as (2.1) is to construct a dynamical system that is governed by the functional and allow the system to evolve to equilibrium. The system may be constructed by applying the principles of Lagrangian mechanics. This leads to dynamic deformable models that unify the description of shape and motion, making it possible to quantify not just static shape, but also shape evolution through time. Dynamic models are valuable for medical image analysis, since most anatomical structures are deformable and continually undergo nonrigid motion *in vivo*. Moreover, dynamic models exhibit intuitively meaningful physical behaviors, making their evolution amenable to interactive guidance from a user (Fig. 2.2).

A simple example is a dynamic snake which can be represented by introducing a time-varying contour $\mathbf{v}(s,t)=(x(s,t),y(s,t))^{\mathsf{T}}$ with a mass density $\mu(s)$ and a damping density $\gamma(s)$. The Lagrange equations of motion for a snake with the internal energy

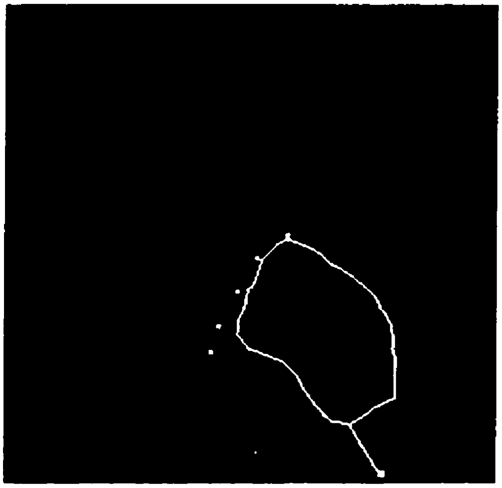


Figure 2.2: Snake deforming towards high gradients in a processed cardiac image, influenced by "pin" points and an interactive "spring" which pulls the contour towards an edge [61].

given by (2.2) and external energy given by (2.4) is

$$\mu\frac{\partial^2 \mathbf{v}}{\partial t^2} + \gamma\frac{\partial \mathbf{v}}{\partial t} - \frac{\partial}{\partial s}\left(w_1\frac{\partial \mathbf{v}}{\partial s}\right) + \frac{\partial^2}{\partial s^2}\left(w_2\frac{\partial^2 \mathbf{v}}{\partial s^2}\right) = -\nabla P(\mathbf{v}(s,t)). \tag{2.7}$$

The first two terms on the left hand side of this partial differential equation represent inertial and damping forces. Referring to (2.6), the remaining terms represent the internal stretching and bending forces, while the right hand side represents the external forces. Equilibrium is achieved when the internal and external forces balance and the contour comes to rest (i.e., $\partial\mathbf{v}/\partial t = \partial^2\mathbf{v}/\partial t^2 = 0$), which yields the equilibrium condition (2.6).

2.1.2 Discretization and Numerical Simulation

In order to compute numerically a minimum energy solution, it is necessary to discretize the energy $\mathcal{E}(\mathbf{v})$. The usual approach is to represent the continuous geometric model $\mathbf{v}$ in terms of linear combinations of local-support or global-support basis functions. Finite elements [121], finite differences [82], and geometric splines [31] are local representation methods, whereas Fourier bases [4] are global representation methods. The continuous model $\mathbf{v}(s)$ is represented in discrete form by a vector $\mathbf{u}$ of shape parameters associated with the basis functions. The discrete form of energies such as $\mathcal{E}(\mathbf{v})$ for the snake may be written as

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\top}\mathbf{K}\mathbf{u} + \mathrm{P}(\mathbf{u}) \tag{2.8}$$

where $\mathbf{K}$ is called the *stiffness matrix*, and $\mathrm{P}(\mathbf{u})$ is the discrete version of the external potential. The minimum energy solution results from setting the gradient of (2.8) to $\mathbf{0}$, which is equivalent to solving the set of nonlinear algebraic equations

$$\mathbf{K}\mathbf{u} = -\nabla \mathrm{P} = \mathbf{f} \tag{2.9}$$

where $\mathbf{f}$ is the generalized external force vector.

The discretized version of the Lagrangian dynamics equation (2.7) may be written as a set of second order ordinary differential equations for $\mathbf{u}(t)$:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \tag{2.10}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{C}$ is a damping matrix. The time derivatives in (2.7) are approximated by finite differences and explicit or implicit numerical time integration methods are applied to simulate the resulting system of ordinary differential equations in the shape parameters $\mathbf{u}$.

Many shape recovery problems do not involve time-varying data. In these situations the mass density μ is often set to zero, resulting in simplified equations of motion and a snake that comes to rest as soon as the internal forces balance the external forces:

$$\mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}. \tag{2.11}$$

Essentially, the massless snake is in a viscous medium. We can solve this first-order dynamics equation iteratively using an explicit first-order Euler method. This method approximates the temporal derivatives with forward finite differences. It updates the shape parameters $\mathbf{u}$ from time t to time $t + \Delta t$ according to the formula

$$\mathbf{u}^{(t+\Delta t)} = \mathbf{u}^{(t)} + \Delta t(\mathbf{C}^{(t)})^{-1}\left(\mathbf{f}^{(t)} - \mathbf{K}\mathbf{u}^{(t)}\right). \tag{2.12}$$

This procedure can be equivalently viewed as minimizing the snake energy $E(\mathbf{u})$ using gradient descent. Depending on the form of the external potential energy $\mathrm{P}(\mathbf{u})$ and on the form of the stiffness matrix $\mathbf{K}$, the simple gradient descent algorithm may converge very slowly to a solution. One way around this problem is to solve equation (2.11) using backward finite differences to arrive at the semi-implicit first-order Euler method

$$(\mathbf{C} + \Delta t\mathbf{K})\,\mathbf{u}^{(t+\Delta t)} = \mathbf{C}\mathbf{u}^{(t)} + \Delta t\mathbf{f}^{(t)}. \tag{2.13}$$

We can view this equation as a series of linear systems, one system for each time step. In the case of snakes (contours), the constant matrix $(\mathbf{K} + \mathbf{C})$ is pentadiagonally banded; hence, each linear system can be solved efficiently using a direct method, such as LU decomposition. The semi-implicit technique [47] is more effective than the explicit Euler method at propagating smoothness constraints along the snake.

For deformable surface models, which are introduced in the next section, the linear systems are larger and the constant matrix term exhibits a more complex banding structure, making matrix factorization techniques too costly. Instead, one can employ iterative methods, such as the conjugate gradient method. To maintain the interactivity of the deformable surface model, the number of conjugate gradient iterations per time step may be limited. Typically, only a few iterations are required to generate a sufficiently accurate solution to each linear system using the previous solution as an initial condition.

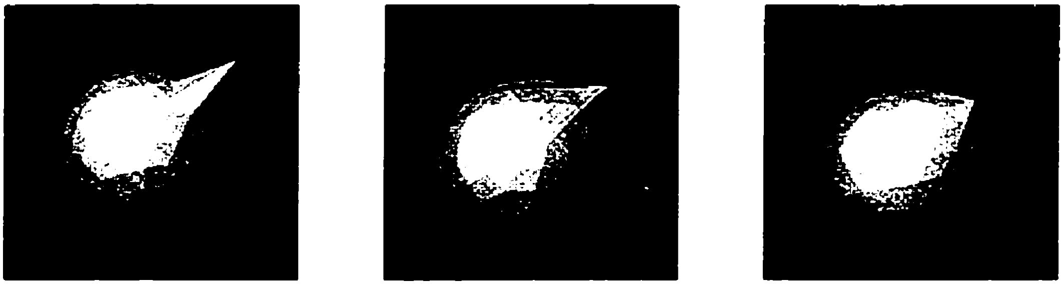
2.1.3 Deformable Surface Models

A deformable surface is represented using a vector-valued parametric representation $\mathbf{x}(u,v) = [x(u,v), y(u,v), z(u,v)]^T$ where vector $\mathbf{x}$ represents the positions of material points (u,v) relative to a reference frame in Euclidean 3-space. The parametric domain of the deformable surface is the unit square $[0,1]^2$.

The surface is constructed of the simulated thin-plate material under tension [99]. The deformation energy of this material is given by the energy functional

$$\mathcal{S}(\mathbf{x}) = \int_0^1\int_0^1 \alpha_{10}\left|\frac{\partial \mathbf{x}}{\partial u}\right|^2 + \alpha_{01}\left|\frac{\partial \mathbf{x}}{\partial v}\right|^2 + \beta_{20}\left|\frac{\partial^2 \mathbf{x}}{\partial u^2}\right|^2 + \beta_{11}\left|\frac{\partial^2 \mathbf{x}}{\partial u \partial v}\right|^2 + \beta_{02}\left|\frac{\partial^2 \mathbf{x}}{\partial v^2}\right|^2 du\,dv. \tag{2.14}$$

$\mathcal{S}$ is a controlled-continuity spline defined in [99]. The non-negative weighting functions $\alpha_{ij}(u,v)$ and $\beta_{ij}(u,v)$ control the elasticity of the material. The α_{10} and α_{01} functions control the tensions in the u and v directions, respectively, while the β_{20} and β_{02} functions



(a) (b) (c)

Figure 2.3: Closed deformable surface being pulled by a spring force showing the effect of various α_{ij} and β_{ij} settings [61]: (a) $\alpha_{ij} = 0.8$, $\beta_{ij} = 0$. (b) $\alpha_{ij} = \beta_{ij} = 0.5$. (c) $\alpha_{ij} = 0$, $\beta_{ij} = 0.8$.

control the corresponding bending rigidities, and the β_{11} function controls the twisting rigidity. Increasing the α_{ij} has a tendency to decrease the surface area of the material, while increasing the β_{ij} tends to make it less flexible.

Analogous to (2.3), a more general, nonlinear strain energy for a deformable surface is a function of the differential area and curvature at each point [103] :

$$\mathcal{S}(\mathbf{x}) = \iint \|\mathbf{G}\|_{\alpha}^{2} + \|\mathbf{B}\|_{\beta}^{2} \, du \, dv. \qquad (2.15)$$

where $\mathbf{G}$ and $\mathbf{B}$ represent the first and second fundamental matrices of the surface forms [32], and α and β weight the matrix norms. The α and β terms determine the resistance to stretching and bending respectively. In Chapter 6, we will describe a topologically adaptable surface formulation that uses a discrete approximation to (2.15).

The choice of boundary conditions can be used to specify the topology of the surface model. Natural boundary conditions specify an open surface. Periodic boundary conditions may be specified in one or both directions to yield a cylindrical and toroidal topology respectively. The addition of pole points to the cylindrical case yields a sphere.

Chapter 3

Medical Image Analysis with Deformable Models

Although originally developed for application to problems in computer vision and computer graphics, the potential of deformable models for use in medical image analysis has been quickly realized. They have been applied to images generated by imaging modalities as varied as X-ray, computed tomography (CT), angiography, magnetic resonance (MR), and ultrasound. Two dimensional and three dimensional deformable models have been used to segment, visualize, track, and quantify a variety of anatomic structures ranging in scale from the macroscopic to the microscopic. These include the brain, heart, face, cerebral, coronary and retinal arteries, kidney, lungs, stomach, liver, skull, vertebra, objects such as brain tumors, a fetus, and even cellular structures such as neurons and chromosomes. Deformable models have been used to track the nonrigid motion of the heart, the growing tip of a neurite, and the motion of erythrocytes. They have been used to locate structures in the brain, and to register images of the retina, vertebra and neuronal tissue.

In the following sections, we review and discuss the application of deformable models to two fundamental problems of medical image analysis: segmentation and shape representation.

3.1 Image Segmentation with Deformable Curves

Defining object contours is an essential first step in many medical image applications. Object contour definition, or segmentation, is currently accomplished in a number of ways: completely manually, automated first guess followed by manual editing, or manual rough delineation followed by automatic contour definition.

Most clinical segmentation is still performed using a completely manual technique known as manual slice editing. In this scenario, a skilled operator, using a computer mouse or trackball, manually traces the region of interest on each slice of an image volume. Manual slice editing suffers from several severe drawbacks. These include labor intensiveness, the difficulty in achieving reproducible results, operator bias, forcing the operator to view each 2D slice separately to deduce and measure the shape and volume of 3D structures, and operator fatigue.

Segmentation using traditional low-level image processing techniques, such as region growing, edge detection, and mathematical morphology operations [81], followed by manual editing, is still often very time consuming, requiring considerable amounts of expert interactive guidance. Furthermore, it is difficult to automate these model-free approaches because of the shape complexity and variability within and across individuals. In general, the underconstrained nature of the segmentation problem limits the efficacy of approaches that use pixel-level representations and take into account only local information. Noise and other image artifacts can cause the generation of incorrect regions or boundaries in objects recovered by these methods.

Manual rough delineation, followed by automatic contour definition using a deformable model based segmentation scheme (Fig. 3.1), can overcome many of the limitations of manual slice editing and traditional image processing techniques. Deformable models can make use of *a priori* knowledge of object shape and smoothness to constrain the segmentation problem. These continuous geometric models consider an object boundary as a single connected structure. Therefore, not only is the connectivity and smoothness of an object guaranteed, but noise, gaps and other irregularities in object boundaries can be overcome. Furthermore, the geometric model representation provides a compact,

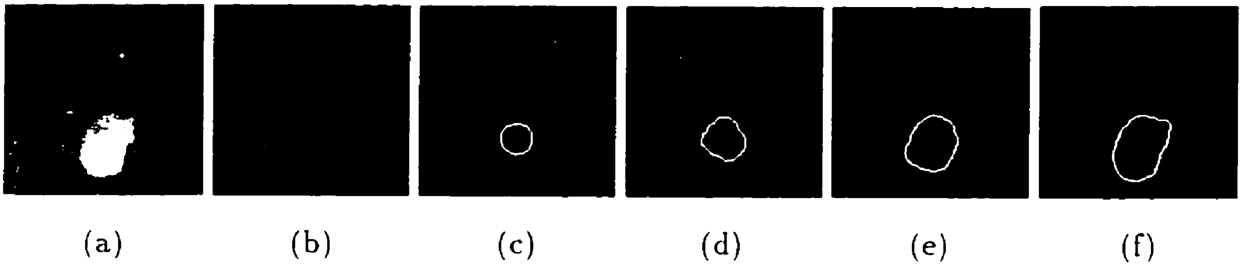


Figure 3.1: (a) Intensity CT image slice of LV. (b) Edge detected image. (c) Initial snake. (d)-(f) Snake deforming towards LV boundary, driven by "inflation" force. [61].

analytical description of object shape. These properties, combined with intuitive interactive mechanisms, lead to accurate, efficient and reproducible segmentations as well as a sound platform from which to develop automatic analysis methods.

Segmentation using deformable contour models, such as snakes [47], was among the first uses of deformable models in medical image analysis [7, 21, 108, 85, 22, 52, 12, 41, 56, 25] for both 2D and 3D data sets. Three dimensional image volumes are processed by placing a snake on an initial image slice near the target object and allowing it to deform onto the object edges. Once the user is satisfied with the result, the fitted contour model is then used as the initial object contour approximation for neighboring slices. These models are then deformed into place and again propagated until all slices have been processed. The resulting sequence of 2D contours can then be connected to form a continuous 3D surface model [54, 17, 21, 22]. This spatial propagation of the deformable contours can dramatically decrease the time taken to segment an object from an image volume [91].

Various methods have been proposed to extend, improve and further automate the deformable contour segmentation process. Cohen and Cohen [22] use an internal "inflation" force to expand a closed snake past spurious edges towards the real edges of the structure, making the snake less sensitive to initial conditions (inflation forces were also employed in [105]). Cohen *et el.* also normalized the image forces to avoid instabilities and to prevent the model from becoming trapped by spurious isolated edge points.

A primary approach to solving energy minimization problems is the calculus of variations. In this approach, finite difference or finite element techniques are used to solve the Euler-Lagrange equations derived from the energy functional. This method can be regarded as gradient descent along the potential energy surface of the deformable contour. Unfortunately, gradient descent does not guarantee finding the global minimum if the energy surface or landscape is not convex and the initial contour model is far from the target. Amini *et al.* [2] use dynamic programming (DP) to carry out a more extensive search for global minima. Although DP provides numerical stability, it does so at a high cost in computational complexity. Williams and Shah [117] propose an alternative (greedy) algorithm to DP which drastically cuts numerical costs; however, it does not guarantee numerical stability. Recently, Mortensen and Barret [70] have proposed a snake-like segmentation tool using DP which allows the user to *interactively* select the most suitable boundary from a set of all optimal boundaries emanating from a seed point.

In an alternative approach, Neuenschwander*et al.* [73] allow a user to specify only the end points of the target contour instead of a complete polygonal outline. Optimization then progresses from these end points towards the center of the snake thereby effectively propagating edge information along the snake, improving its convergence properties. Blake and Zisserman [10] propose an algorithm (Graduated Nonconvexity) which is based on iterative approximation of the energy functional by a convex function; it can bridge low ridges in the potential field and eliminate termination in local valleys.

Poon *et al.* [80] and Grzeszczuk and Levin [40] minimize the energy of active contour models using simulated annealing which is known to give global solutions and allows the incorporation of non-differentiable constraints. In general however, deterministic algorithms are usually preferred due to their faster convergence and the importance of user interaction in medical image segmentation scenarios.

Poon *et al.* [80] also use a discriminant function to incorporate region based image features into the image forces of their active contour model. The additional image features serve as a constraint for global segmentation consistency (i.e. every image pixel contributes to the discriminant function). The result is a more robust energy functional and a much better tolerance to deviation of the initial guess from the true boundaries. Others researchers [84, 16, 15, 42, 36, 46, 60] have also integrated region-based information into deformable contour models in an attempt to decrease sensitivity to insignificant edges and initial model placement.

Several limitations or problems of snake-based segmentation methods were mentioned in Chapter 1. An additional problem is that the selection of snake elasticity and rigidity parameters is often ad-hoc and optimal parameter settings are difficult to determine. Another limitation is the oscillatory behavior that can be displayed by snakes due to high intensity gradients used to push a snake towards edges. Several approaches have been suggested to deal with these difficulties. The stability of snakes has been investigated by adjusting internal parameters in [86, 8, 25]. Leymarie and Levine [53] introduced bounds on the image forces, new rules for setting the elasticity parameters, and a new terminating condition. In [67, 85], the elastic parameters of the deformable models are automatically adjusted to improve the accuracy and robustness of the shape recovery.

3.1.1 Implicit Deformable Contour Models

There are two main approaches to deformable contour models. *Parametric* models, for example snakes, are specified as parametric curves defined initially, which do not (necessarily) change during the optimization process. *Implicit* models, on the other hand do not depend on any particular parameterization.

One of the strongest limitations of parametric deformable models is the topology of the target object must be known in advance. Therefore, for images with multiple objects where some objects are contained by, or embedded in others, parametric deformable models require extensive user interaction. Recently, several researchers [13, 59, 115, 14, 88] have been developing implicit deformable contour models that allow a deformable contour to not only represent long tube-like shapes or shapes with bifurcations, but also to dynamically sense and change its topology.

In this approach, the deformable contour model is viewed as the level set of a higher dimensional surface.[1]. The evolution of the surface is defined by a Hamilton-Jacobi-like partial differential equation. There are several attractive features to this approach. First, discrete mesh points used in the numerical implementation of the PDE do not

---

[1] Appendix A provides a detailed explanation of level set evolution techniques

move, resulting in a stable computation. Topological changes in the contour are handled naturally by exploiting the property that the level set need not be simply connected: the implicitly defined surface always remains a function even if the level set changes topology. The use of evolving contours as level sets and the relevant numerical implementation was first proposed by Osher and Sethian for flame propagation [74, 90] and used by Kimia *et al.* in constructing a representation for shape in computer vision [48]. Malladi et al. [59] and Caselles *et al.* [14] illustrated the effectiveness of this approach for capturing interesting structures in medical images. Recently, several researchers have extended the level set technique to 3D [58].

There is some connection between the implicit models and parametric snakes models. The energy minimization of the snakes model involves variational principles which can be equivalently formulated as solutions to partial differential equations (Euler-Lagrange equations). Caselles [14] pointed out that setting the rigidity term to zero in the energy functional of the snake model establishes an equivalence between this parametric model and a geometric curve evolution, since they both minimize the length of the contour in the metric induced by the image.

3.2 Volume Image Segmentation with Deformable Surfaces

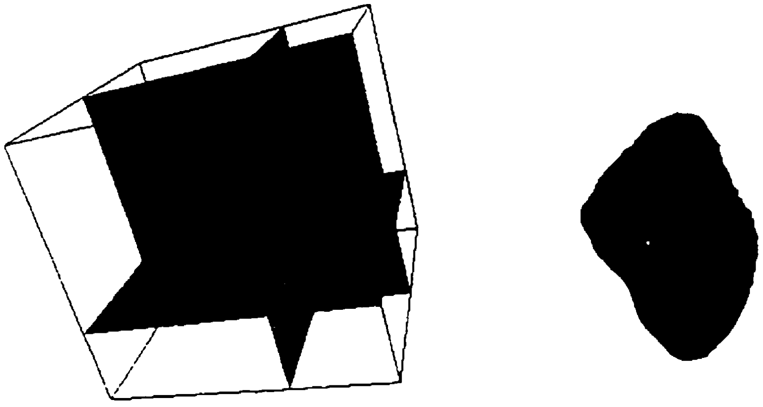
Segmenting 3D image volumes slice by slice, either manually or by applying 2D contour models, is a laborious process and requires a post-processing step to connect the sequence of 2D contours into a continuous surface. Furthermore, the resulting surface reconstruction can contain inconsistencies or show rings or bands. The use of a true 3D deformable surface model on the other hand, can result in a more efficient, robust segmentation technique which ensures a globally smooth and coherent surface between image slices.

Deformable surface models in 3D were first used in computer vision [105] and computer graphics [100] and several variants have since been developed. In the field of computer-aided design, Terzopoulos and Qin [104] recently proposed D-NURBS (Dynamic Non-Uniform Rational B-Splines), a physics-based generalization of the NURBS representation. D-NURBS allow a modeler to interactively sculpt complex shapes to required specification by applying simulated forces.

Many researchers have explored the use of deformable surface models for segmenting structures in medical image volumes. Miller [69] constructs a polygonal approximation to a sphere and geometrically deforms this "balloon" model until the balloon surface conforms to the object surface in 3D CT data. The segmentation process is formulated as the minimization of a cost function where the desired behavior of the balloon model is determined by a local cost function associated with each model vertex. The cost function is a weighted sum of three terms: a deformation potential that "inflates" the model vertices towards the object boundary, an image term that identifies features such as edges and opposes the balloon expansion, and a term that maintains the topology of the model by constraining each vertex to remain close to the centroid of its neighbors.

Cohen and Cohen [20, 22] and McInerney and Terzopoulos [61] use finite element and physics-based techniques to implement an elastically deformable cylinder and sphere, respectively. The models are used to segment the inner wall of the left ventricle of the heart from MR or CT image volumes (Fig. 3.2). These deformable surfaces are based on the thin-plate under tension surface spline (2.2), which controls and constrains the stretching and bending of the surface. The models are fitted to data dynamically by integrating Lagrangian equations of motion through time in order to adjust the deformational degrees of freedom. Furthermore, the finite element method is used to represent the models as a continuous surface in the form of weighted sums of local polynomial basis functions. The finite element method provides an analytic surface representation and the use of high-order polynomials rather than polygons means that fewer elements are required to accurately represent an object. Pentland and Sclaroff [77] and Nastar and Ayache [72] also develop physics-based models but use a reduced modal basis for the finite elements.

Staib and Duncan [93] describe a 3D surface model used for geometric surface matching to 3D medical image data. The model uses a Fourier parameterization which decomposes the surface into a weighted sum of sinusoidal basis functions. Several different surface types are developed such as tori, open surfaces, closed surfaces and tubes. Surface finding is formulated as an optimization problem using gradient ascent which attracts the surface to strong image gradients in the vicinity of the model. An advantage of the



(a) (b)

Figure 3.2: (a) Deformable "balloon" model embedded in an edge detected CT image volume deforming towards LV edges. Three orthogonal slices of the volume are shown. (b) Reconstruction of LV [61].

Fourier parameterization is that it allows a wide variety of smooth surfaces to be described with a small number of parameters. That is, a Fourier representation expresses a function in terms of an orthonormal basis and higher indexed basis functions in the sum represent higher spatial variation. Therefore, the series can be truncated and still represent relatively smooth objects accurately.

Szeliski *et al.* [96] overcome the fixed topology limitation of parametric deformable models by using a dynamic, self-organizing oriented particle system to model surfaces of objects. The oriented particles, which can be visualized as small, flat disks, evolve according to Newtonian mechanics and interact through external and interparticle forces. The external forces attract the particles to the data while interparticle forces attempt to group the particles into a coherent surface. The particles can reconstruct objects with complex shapes and topologies by "flowing" over the data, extracting and conforming to meaningful surfaces. A triangulation is then performed which connects the particles into a continuous global model that is consistent with the inferred object surface.

Leitner and Cinquin [51] have also developed a parametric deformable model, using a tensor product spline surface, that is capable of changing topology from a sphere to a torus. Other notable work involving 3D deformable surface models and medical image applications can be found in [27, 115, 97, 24] as well as several models described in the following section.

3.3 Incorporating A Priori Knowledge

In medical images, the general shape, location and orientation of objects is often known and this knowledge may be incorporated into the deformable model in the form of initial conditions, data constraints, constraints on the model shape parameters, or into the model fitting procedure. The use of implicit or explicit anatomical knowledge to guide shape recovery is especially important for robust automatic interpretation of medical images. For automatic interpretation, it is essential to have a model that not only describes the size, shape, location and orientation of the target object but that also permits expected variations in these characteristics. Automatic interpretation of medical images can relieve clinicians from the labor intensive aspects of their work while increasing the efficiency, accuracy, and reproducibility of the interpretations.

A number of researchers have incorporated knowledge of object shape into deformable models by using deformable shape templates. These models usually use "hand-crafted" global shape parameters to embody *a priori* knowledge of expected shape and shape variation of the structures and have been used successfully for many applications of automatic image interpretation. The idea of deformable templates can be traced back to the early work on spring loaded templates by Fischler and Elshlager [33]. An excellent example in computer vision is the work of Yuille *et al.* [120] who construct deformable templates for detecting and describing features of faces, such as the eye. In medical image analysis, Lipson *et al.* [55] note that axial cross sectional images of the spine yield approximately elliptical vertebral contours and consequently extract the contours using a deformable ellipsoidal template.

Deformable models based on superquadrics are another example of deformable shape templates that are gaining in popularity in medical image research. Superquadrics contain a small number of intuitive global shape parameters that can be tailored to the average shape of a target anatomic structure. Furthermore, the global parameters can often be coupled with local shape parameters such as splines resulting in a powerful shape representation scheme. For example, Metaxas and Terzopoulos [68] employ a dynamic deformable superquadric model [102] to reconstruct and track human limbs from 3D biokinetic data. Their models can deform both locally and globally by incorporating the global shape parameters of a superellipsoid with the local degrees of freedom of a membrane spline in a Lagrangian dynamics formulation. The global parameters efficiently capture the gross shape features of the data, while the local deformation parameters reconstruct the fine details of complex shapes. Using Kalman filtering theory, they develop and demonstrate a biokinetic motion tracker based on their deformable superquadric model.

Vemuri and Radisavljevic [112, 111] construct a deformable superquadric model in an orthonormal wavelet basis. This multi-resolution basis provides the model with the ability to continuously transform from local to global shape deformations thereby allowing a continuum of shape models to be created and to be represented with relatively few parameters. They apply the model to segment and reconstruct anatomical structures in the human brain from MRI data.

As a final example, Bardinet *et al.* [5, 6] fit a deformable superquadric to segmented 3D cardiac images and then refine the superquadric fit using a volumetric deformation technique known as free form deformations (FFDs). FFDs are defined by tensor product trivariate splines and can be visualized as a rubber-like box in which the object to be deformed (in this case the superquadric) is embedded. Deformations of the box are automatically transmitted to embedded objects. This volumetric aspect of FFDs allows two superquadric surface models to be simultaneously deformed in order to reconstruct the inner and outer surfaces of the left ventricle of the heart and the volume in between these surfaces. Further examples of deformable superquadrics can be found in [76, 18]

Several researchers cast the deformable model fitting process in a probabilistic framework and include prior knowledge of object shape by incorporating prior probability distributions on the shape variables to be estimated [111, 92, 119, 44]. For example, Staib and Duncan [92] use a deformable contour model on 2D echocardiograms and MR images to extract the LV of the heart and the corpus callosum of the brain, respectively.

This closed contour model is parameterized using an elliptic Fourier decomposition and *a priori* shape information is included as a spatial probability expressed through the likelihood of each model parameter. The model parameter probability distributions are derived from a set of example object boundaries and serve to bias the contour model towards expected or more likely shapes.

Szekely *et al.* [95] have also developed Fourier parameterized models. Furthermore, they have added elasticity to their models to create "Fourier snakes" in 2D and elastically deformable Fourier surface models in 3D. By using the Fourier parameterization followed by a statistical analysis of a training set, they define mean organ models and their eigen-deformations. An elastic fit of the mean model in the subspace of eigenmodes restricts possible deformations and finds an optimal match between the model surface and boundary candidates.

Cootes *et al.* [23] and Hill *et al.* [43] present a statistically based technique for building deformable shape templates and use these models to segment various organs from 2D and 3D medical images. The statistical parameterization provides global shape constraints and allows the model to deform only in ways implied by the training set. The shape models represent objects by sets of landmark points which are placed in the same way on an object boundary in each input image. For example, to extract the LV from echocardiograms, they choose points around the ventricle boundary, the nearby edge of the right ventricle, and the top of the left atrium. The points can be connected to form a deformable contour. By examining the statistics of training sets of hand-labeled medical images, and using principal component analysis, a shape model is derived that describes the average positions and the major modes of variation of the object points. New shapes are generated using the mean shape and a weighted sum of the major modes of variation. Object boundaries are then segmented using this "point distribution model" by examining a region around each model point to calculate the displacement required to move it towards the boundary. These displacements are then used to update the shape parameter weights.

Chapter 4

Topologically Adaptable Snakes

In this chapter we describe an extension to the classical snakes paradigm using a space partitioning or decomposition technique to create topologically adaptable snakes (T-snakes). We begin by introducing the idea of using an affine cell decomposition of an image domain to iteratively reparameterize a snakes model and to perform topological transformations. We then describe the implementation of the model in detail, including an overview of the theory of affine cell decomposition and approximation using a special type of affine cell known as a simplex. We conclude the chapter with an overview of the complete algorithm as well as a brief discussion of potential limitations of T-snakes.

4.1 Model Overview

We define our T-snakes model as a closed 2D contour consisting of a set of nodes connected in series. The viscoelastic-like T-snake is a discrete approximation to a traditional snakes model and retains many of the snakes properties. In our implementation, an "inflation" force is used to push the model towards image edges until it is opposed by external image forces. The internal forces act as a smoothness constraint and users can interact with the model using spring forces and other constraints. The deformation of the model is governed by discrete Lagrangian equations of motion.

Unlike traditional snakes, the set of nodes and interconnecting elements of a T-snake does not remain constant during its evolution. That is, we decompose the image domain into a grid of discrete cells. As the model moves under the influence of external and internal forces, we reparameterize the model with a new set of nodes and elements by efficiently computing the intersection points of the model with the superposed grid. We also keep track of the interior region of the model by "turning on" any grid vertices the T-snake passed over during its motion. By reparameterizing the model at each iteration of the evolutionary process, we create a simple, elegant and automatic model subdivision technique and the grid provides a framework for robust topological transformations. This allows the model to be relatively independent of its initial placement and "flow" into complex shapes with complex topologies in a stable manner. Conversion to the traditional parametric snakes model representation is simply a matter of disabling the grid at any time during the evolutionary process. By providing a boundary representation as well as a representation of the interior region of an object, this hybrid snakes model combines the space partitioning, intrinsic parameterization and topological flexibility properties of an implicit formulation with the boundary properties of a parametric model.

4.1.1 Relationship to Front Propagation

The motion of a T-snake is analogous to the motion of a propagating front, as in flame propagation (see Appendix A). There exist several traditional numerical techniques used to solve the equations of motion for this problem. The first set of techniques, as discussed in [90], parameterizes the moving front and discretizes the parameterization into a set of marker particles or nodes. The normal direction to the front, the curvature, and the stretch are approximated by discrete derivatives at the marker nodes. The motion of the marker nodes is then governed by approximation to the equations of motion. This *Lagrangian*[1] formulation reportedly suffers from instability problems and topological changes in the front connectivity are difficult to achieve.

The second set of techniques, known as "volume of fluid" techniques [19], track the motion of the interior region rather than the boundary. These algorithms discretize the interior region by superposing a grid of cells on the domain and assigning to each cell a "volume fraction" corresponding to the amount of interior fluid currently in the cell. The front is moved by constructing local polygonal approximations to the front in each cell

---

[1] A *Lagrangian* formulation or description of particle motion gives the velocity of each particle (i.e. a particle is tracked with respect to its original coordinates). An *Eulerian* description gives the velocity at fixed points (i.e. a particle is tracked with respect to its final coordinates).

based on neighboring volume fractions. This Eulerian technique is stable and topological changes are easily handled. It may be difficult, however, to calculate front properties, such as curvature and normals, from the coarse boundary representation used without resorting to very fine resolution grids.

A third technique is the level set approach of Osher and Sethian [74] previously introduced in Chapter 3. They model the front as the level set of an evolving higher-dimensional surface, where the motion of the surface is described by a scalar Hamilton-Jacobi equation with parabolic right hand side. An Eulerian formulation provides stability and topological changes are handled naturally in the higher dimensional space.

A T-snake is a hybrid model that contains aspects of all three techniques and attempts to combine their strengths. Between reparameterization stages, a T-snake behaves as a classical snake and evolves according to Lagrangian dynamics. This Lagrangian formulation phase allows any data-derived or user-defined force to guide the snake. During the reparameterization phase, the snake is reparameterized in terms of the simplicial grid and the fixed grid points are used to track the interior of the closed contour model, creating a space partitioning similar to that of an implicit function. This Eulerian formulation phase provides stability, intrinsic parameterization, and topological flexibility.

4.2 Model Description

A T-snake is a discrete form of the classical snake described in Chapter 2. It is defined as a set of N nodes indexed by $i = 1, \ldots, N$, where the nodes are connected in series. We associate with these nodes time varying positions $\mathbf{x}_i(t) = [x_i(t), y_i(t)]$, along with tension forces $\boldsymbol{\alpha}_i(t)$, rigidity forces $\boldsymbol{\beta}_i(t)$, inflation forces $\boldsymbol{\rho}_i(t)$, and external forces $\mathbf{f}_i(t)$ that act in the image plane. A periodic boundary condition is applied $\mathbf{x}_1(t) = \mathbf{x}_N(t)$ to produce a closed contour model.

The behavior of a T-snake is governed by a discrete and simplified form of equation (2.7). The result is a set of first-order ordinary differential equations of motion

$$\gamma_i \dot{\mathbf{x}}_i + \boldsymbol{\alpha}_i + \boldsymbol{\beta}_i = \boldsymbol{\rho}_i + \mathbf{f}_i, \tag{4.1}$$

where $\dot{\mathbf{x}}_i$ is the velocity of node i, γ_i is a damping coefficient that controls the rate of dissipation of the kinetic energy of the nodes, and $\mathbf{f}_i$ and $\boldsymbol{\rho}_i$ are external forces that attract model nodes toward salient image edges. Since the model has no inertia, it comes to rest as soon as the applied forces balance the internal forces (i.e. $\dot{\mathbf{x}}_i = 0$).

4.2.1 Internal Forces

We connect the nodes in series using nonlinear springs which we also refer to as model elements. Let l_i be the given reference length of the spring connecting node i to node $i+1$, and let $\mathbf{r}_i(t) = \mathbf{x}_{i+1}(t) - \mathbf{x}_i(t)$ be the separation of the nodes. We want the spring to resist expansion or compression only when its actual length $\|\mathbf{r}_i\|$ is greater or less than l_i respectively. Hence, given the deformation $\epsilon_i(t) = \|\mathbf{r}_i\| - l_i$, we define the tension force

$$\boldsymbol{\alpha}_i = a_i \epsilon_i \hat{\mathbf{r}}_i - a_{i-1}\epsilon_{i-1}\hat{\mathbf{r}}_{i-1} \tag{4.2}$$

where the a_i is the tension parameter for node i and the caret denotes a unit vector. Since the set of nodes and springs of our deformable contour model does not remain constant during its evolution, we define the rest-lengths of the springs at time t in terms of the length of the springs at time $t - \Delta t$. This gives our model the behavior of a viscoelastic material.

The main objective of the rigidity forces are to minimize the local curvature of the contour. There exist several discrete approximations to local curvature [117, 75]. For example, one can define the local curvature at a vertex to be the difference between the directions of the two normalized edge segments that join at that location. We use an approximation that measures the distance between a vertex and the centroid of its neighbors [69]. Consequently, the rigidity force attempts to minimize this distance and is defined as

$$\boldsymbol{\beta}_i(t) = b_i(\mathbf{x}_i(t) - \frac{1}{2}(\mathbf{x}_{i-1} + \mathbf{x}_{i+1})). \tag{4.3}$$

where b_i is the rigidity parameter. This force can be made scale invariant by dividing the right hand side of (4.3) by the distance between the neighbor nodes. A force in the opposite direction, multiplied by 1/2, is also be applied to each of the neighbor nodes.

4.2.2 External Forces

An inflation force is used to push the model towards intentity edges in the image $I(x, y)$, until it is opposed by the image forces. The inflation force takes the form

$$\boldsymbol{\rho}_i = qF(I(x, y))\mathbf{n}_i, \tag{4.4}$$

where $\mathbf{n}_i$ is the unit normal vector to the contour at node i, and q is the amplitude of this force. The binary function F links the inflation force to the image data

$$\begin{aligned} F(I(x, y)) &= +1, \quad I(x, y) \geq T, \\ F(I(x, y)) &= -1, \quad I(x, y) < T, \end{aligned} \tag{4.5}$$

where T is an image intensity threshold. The functional F makes the T-snake contract when $I(x, y) < T$.

To stop the contour at significant edges, we construct the force

$$\mathbf{f}_i = p\nabla P(\mathbf{x}_i), \tag{4.6}$$

where the weight p controls the strength of the force and the potential P is defined by (2.5). The weights p and q are usually chosen to be of the same order, with p slightly larger than q so that a significant edge will stop the inflation, but with q large enough so that the model will pass through weak or spurious edges.

To calculate a continuous image function $I(x, y)$ for either (4.6) or (4.6) we compute the intensity at an arbitrary point (x, y) by bilinearly interpolating the intensities at the four pixels surrounding (x, y).

4.2.3 Numerical Integration

We integrate equation (4.1) forward through time using an explicit first-order Euler method. This method approximates the temporal derivatives with forward finite differences. It updates the positions of the model nodes from time t to time $t + \Delta t$ according to the formula

$$\mathbf{x}_i^{(t+\Delta t)} = \mathbf{x}^{(t)} - \frac{\Delta t}{\gamma}\boldsymbol{\alpha}_i^{(t)} - \boldsymbol{\beta}_i^{(t)} + \mathbf{f}_i^{(t)} + \boldsymbol{\rho}_i^{(t)}. \tag{4.7}$$

While the explicit Euler method is known to be unstable unless small time steps are used, it is simple, efficient and, in our experience, a very reasonable range of time step sizes can be found that produce stable T-snake behavior resulting in accurate segmentations. However, more sophisticated numerical integration techniques such as Runge-Kutta [82] or semi-implicit methods [103] could also be used and would result in better convergence and larger time steps, at the expense of a more complicated numerical implementation.

4.3 Simplicial Cell Decomposition

The intuitive idea of a space decomposition is to subdivide space into a collection of disjoint connected subsets. Subdivision of a domain into simpler subdomains, along with a structure that links these subdomains, allows us to obtain valuable information about the geometry and the topology of the space. One common subset is a k-dimensional cell (i.e. a set which is homeomorphic to an open disk of some dimension k), where the boundary of each cell is defined to be a finite union of lower dimensional cells. The subdivision of space using such subsets is known as a *cellular complex* [71]. The framework of cellular complexes is a powerful tool for constructing definitions and proofs for image topology [49].

Affine cell decompositions are examples of special cases of cell decompositions obtained by restricting the geometry of the cells to that of a *convex polytope*. There are two main types of affine cell decomposition methods: non-simplicial and simplicial. Most nonsimplicial methods employ a rectangular tessellation of space. The marching cubes algorithm [57] is an example of this type. These methods are fast and easy to implement but they cannot be used to represent the boundaries of an implicitly defined object unambiguously without the use of a disambiguation scheme. Although non-simplicial cell decomposition methods can be used to reparameterize a T-snake, we use *simplicial* cell decomposition [71, 110]. Simplicial cell decomposition is a theoretically sound decomposition method that relies on classical results from algebraic topology related to the piecewise linear structure of a *simplicial complex*. Computational work on this type of method was pioneered by Allgower [1]. Throughout the remainder of this thesis, we will use the terms "affine cell" and "simplicial cell" interchangeably.

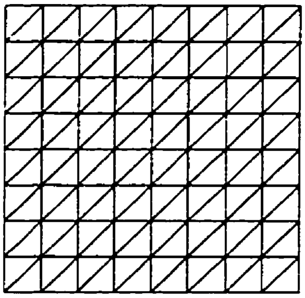


Figure 4.1: Freudenthal triangulation.

In a simplicial cell decomposition — also known as a *triangulation* — space is partitioned into cells defined by open simplices, where an n-simplex is the simplest geometrical object of dimension n: e.g., a triangle in 2D or a tetrahedron in 3D. It is desirable to have a small number of cell types, e.g. *congruent cells*, that differ only by orientation or reflection. If all the cells are identical, computations can be made very simple and efficient. The simplest triangulation of Euclidean space $\Re^n$ with this property is the *Coxeter-Freudenthal* triangulation (Fig. 4.1). It is constructed by dividing space using a uniform cubic grid and the triangulation is obtained by subdividing each cube into $n!$ simplices.

More formally, given a set of $k+1$ points $\{v_0, v_1, \ldots, v_k\} \in \Re^m$, the set is called *affinely independent* if the vectors $\{v_1 - v_0, v_2 - v_0, \ldots, v_k - v_0\}$ are linearly independent (i.e. they do not all belong to the same k-dimensional hyperplane).

DEFINITION 4.1. The set

$$\sigma = [v_0, \ldots, v_k] = \left\{ v \in \Re^n : v = \sum_{i=0}^{k} \lambda_i v_i \mid \sum_{i=0}^{k} \lambda_i = 1, \lambda_i \geq 0 \right\} \tag{4.8}$$

is called a $k-$*simplex* having vertices $\{v_i\}_{i=0}^{k}$. In other words, the simplex σ^k is a *convex linear hull* of the points $\{v_i\}_{i=0}^{k}$. The coefficients λ_i are usually called the barycentric coordinates of v. If $\{w_i\}_{i=0}^{l}$ is a nonempty subset of $\{v_i\}_{i=0}^{k}$, the l-simplex $\tau = [w_0, \ldots, w_l]$ is called an *l-dimensional face* of σ. In general, the simplex σ_i^{k-1} is called the ith $(k-1)$-dimensional face of the simplex σ^k. Thus, a k-dimensional simplex has $k+1$ faces of dimension $k-1$. A zero-dimensional face is a vertex, a one-dimensional face is an edge, and a two-dimensional face is a triangle. If σ is a k-simplex, the *barycenter* of σ is defined by $b(\sigma) = \frac{1}{k+1} \sum_{i=0}^{k} v_i$. It is convenient to use a face of σ which is formed from the convex

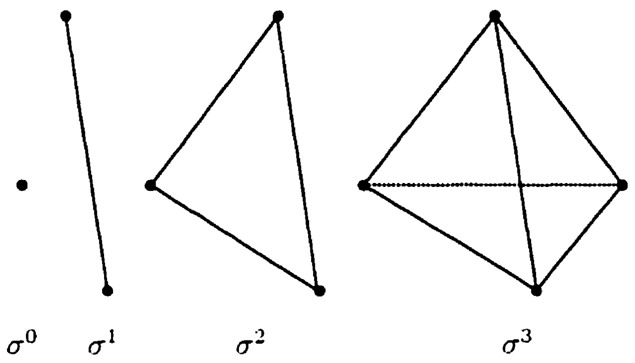


Figure 4.2: A 3-simplex (far right) and examples of its $k-1$-dimensional faces, where (from left to right) $k = 0, 1$, and 2 respectively.

combination of the vertices of σ with one or more specific vertices omitted (Figure 4.2).

DEFINITION 4.2. A ***simplicial complex*** or triangulation $\mathcal{T}$ over a domain $D \in \Re^{n+1}$ is a family of simplices with the following properties:

1. $D = \cup_{\sigma \in \mathcal{T}}$.
2. If $\sigma_1, \sigma_2 \in \mathcal{T}$, then $\sigma_1 \cap \sigma_2$ is either empty or a common face (lower dimensional simplex) of both σ_1, σ_2.
3. If D is a compact subset of $\Re^{n+1}$ then it intersects only finitely many simplices.

DEFINITION 4.3. The ***meshsize*** of a triangulation $\mathcal{T}$ is defined by $\delta = \sup_{\sigma \in \mathcal{T}} \text{diam } \sigma$. The meshsize depends on the space norm, and its value for different norms differ by a constant. Using the Euclidean norm the meshsize of the unitary Freudenthal triangulation in $\Re^n$ is $\sqrt{n}$.

4.3.1 Simplicial Approximation

Medical image volumes represent the sampling, typically at the vertices of a cubical lattice of voxels, of an intensity function $I(x, y, z)$

$$\{(\mathbf{v}_{ijk}, I_{ijk}) | \ 1 \leq i \leq L, \ 1 \leq j \leq M, \ 1 \leq k \leq N\}. \tag{4.9}$$

The boundaries of anatomic structures embedded in these images can be viewed as isovalue surfaces S of this implicit function $I(x, y, z) = I_S$ (or $I(x, y) = I_C$, an isovalue

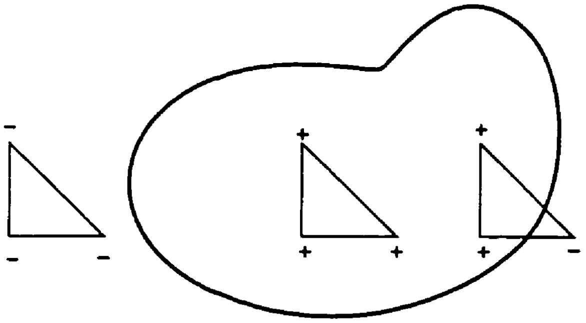


Figure 4.3: Simplex classification.

contour in 2D).[2]

Simplicial cell decompositions provide a framework for the creation of robust, consistent local polygonal (affine) approximations of the boundary contours (or boundary surfaces in 3D) of these implicitly defined structures. In 2D, an anatomic structure partitions an image into two open sets of dimension 2 (the interior and exterior points) and one open set of dimension 1 (the boundary points). A simplex σ^2 can be classified in relation to this partitioning of space by testing the "sign" of its vertices. If the signs are the same for all vertices, the simplex must be totally inside or outside the structure. If the signs are different, the boundary of the structure must intersect the simplex (Fig. 4.3). In a k-simplex, the negative (inside) vertices can always be separated from the positive (outside) vertices by a single plane; thus an unambiguous polygonalization of the simplex always exists. Furthermore, by the definition of a simplicial complex, a consistent polygonization of the entire boundary contour (surface) will result.

In a 2D image, the set of grid triangles (2-dimensional simplices) that intersect the boundary contour of the anatomic structure (the boundary triangles) are called the *transverse* simplices. These boundary triangles form a two dimensional combinatorial manifold that has as its dual a one dimensional manifold that approximates the contour. The one dimensional manifold is constructed from the intersection of the object boundary contour with the edges of each boundary triangle. The intersection points result in one line segment (1-dimensional simplex) approximating the boundary contour inside each bound-

---

[2] Due to sampling artifacts, spatial aliasing, and noise, I_S is rarely well-defined.

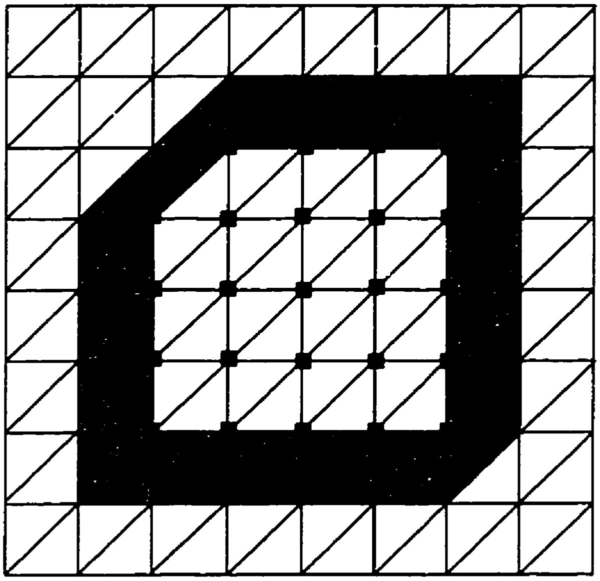


Figure 4.4: Simplicial approximation (dashed-line) of an object contour (solid-line) using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded.

ary triangle (Fig. 4.4). Each line segment intersects a boundary triangle at two distinct edges, separating the inside vertices from the outside vertices. The set of all these line segments constitute the combinatorial manifold that approximates the boundary contour of the object.

Restating the above more formally, let $H : \Re^{n+1} \rightarrow \Re^n$ be a map. Given a triangulation $\mathcal{T}$ of $\Re^{n+1}$, we are approximating the components of $H^{-1}(0)$ (we shall use the isovalue 0 without loss of generality) by using the values of H on the vertices of $\mathcal{T}$. This leads to the following formal definitions and theorems [1]:

DEFINITION 4.4. For any map $H : \Re^{n+1} \rightarrow \Re^n$, the piecewise linear approximation $H_\mathcal{T}$ to H relative to the triangulation $\mathcal{T}$ of $\Re^{n+1}$ is the map which is uniquely defined by

1. $H_\mathcal{T}(v) = H(v)$ for all vertices of $\mathcal{T}$.

2. For any $(n+1)$-simplex $\sigma = [v_1, v_2, \ldots, v_{n+2}] \in \mathcal{T}$, the restriction $H_\mathcal{T}|_\sigma$ of $H_\mathcal{T}$ to σ is an affine map. Consequently, if $u = \sum_{i=1}^{n+2} \lambda_i v_i$ is a point in σ, then its barycentric coordinates λ_i satisfy $\sum_{i=1}^{n+2} \lambda_i = 1$ and $\lambda_i \geq 0$ for $i = 1, \ldots, n+2$, and since $H_\mathcal{T}$ is

affine, we have

$$H_{\mathcal{T}}(u) = H\left(\sum_{i=1}^{n+2} \lambda_i v_i\right) = \sum_{i=1}^{n+2} \lambda_i H(v_i). \tag{4.10}$$

3. The simplicial approximation of H is obtained as the union

$$H_{\mathcal{T}} = \cup_{\sigma \in \mathcal{T}} H_\sigma. \tag{4.11}$$

where $H_{\mathcal{T}}(v) = H_\sigma(v)$ for $v \in \sigma$ and $\sigma \in \mathcal{T}$.

Implicit Function Theorem 1 *Let* $H : \Re^{n+k} \to \Re^n$ *be a smooth map such that* $0 \in$ *range*(H)*. Then* $M = \{x \in \Re^{n+k} | H(x) = 0, x$ *is a regular point of* $H\}$ *is a smooth* k*-dimensional manifold.*

For $k = 1$, the manifold is a curve and for $k = 2$, the manifold is a surface

Theorem 1 *Let* 0 *be a regular value of* $H_{\mathcal{T}}$*. If* $\sigma \in \mathcal{T}$ *has a non-empty intersection with* $H_{\mathcal{T}}^{-1}(0)$ *(i.e. it is transverse), then* $M_\sigma := \sigma \cap H_{\mathcal{T}}^{-1}(0)$ *is a* k*-dimensional polytope, and the family*

$$M_{\mathcal{T}} := \{M_\sigma | \sigma \in \mathcal{T}, \sigma \cap H_{\mathcal{T}}^{-1}(0) \neq \emptyset\} \tag{4.12}$$

is a k*-dimensional piecewise linear manifold approximating* M.

Proposition 1 *Assume that* H *is Lipshitz continuous (i.e. there exists a constant* $\alpha > 0$ *such that* $\|H(u) - H(v)\| \leq \alpha \|u - v\|$ *holds for all* $u, v \in \Re^{n+k}$*) and triangulation* $\mathcal{T}$ *has meshsize* $\delta > 0$*. Then* $\|H(x) - H_{\mathcal{T}}(x)\| \leq \frac{1}{2}\alpha\delta^2$ *for* $x \in \Re^{n+k}$.

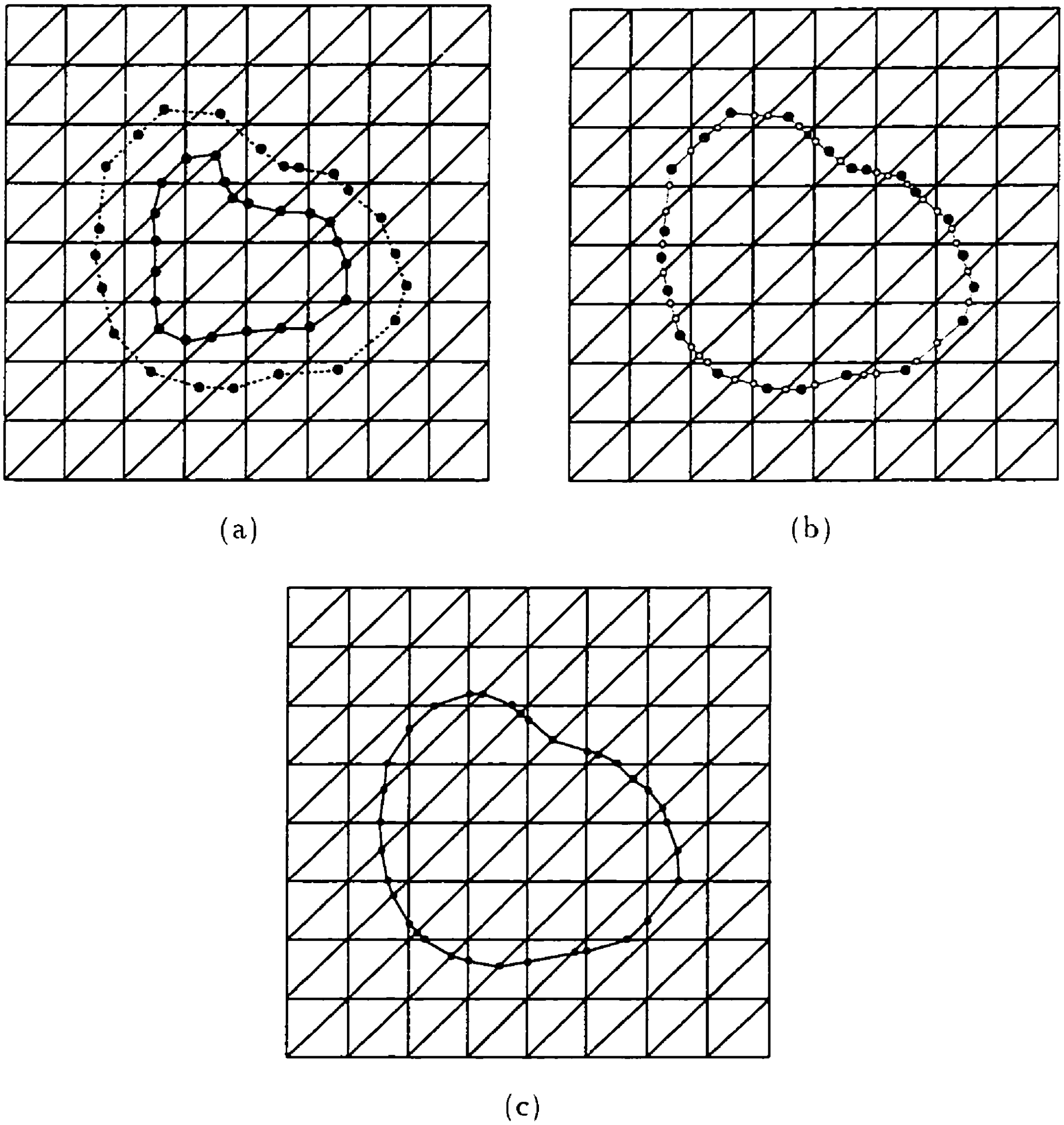
The theorem and proposition are not proved here; the reader is referred to [1].

4.3.2 Iterative Reparameterization

During N time steps of equation (4.7) (referred to as a *deformation step*) a T-snake moves from its current position to a new position.[3] At the beginning of the deformation step, the model nodes are defined in terms of the edges of the grid boundary triangles. At the end of the deformation step, the nodes have moved "off" of the grid triangle edges

---

[3] The T-snake is either expanding or shrinking, but not both. See Section 4.4 for a discussion of T-snake deformation modes.


(a)

(b)

(c)

Figure 4.5: Phase 1 of T-snake reparameterization: (a) T-snake expands and moves off grid during deformation step, (b) new model nodes are computed, (c) new T-snake nodes and elements.

(Figures 4.5a). We then reestablish the correspondence of the model with the grid by computing a new simplicial approximation of the deformed T-snake. This new simplicial approximation is computed using a novel two-phase "reparameterization" algorithm. In phase one, we perform a local search and intersection test for each model element. That is, for each element connecting two nodes, we compute a bounding box of the element at its new position. Using this bounding box, we determine which grid triangle edges may potentially intersect the model element. For each of the edges within the bounding box, an intersection test is performed with the model element. If an intersection point is found, it is recorded and may become a node of the updated model (Figures 4.5b,c). If an intersection point for this edge already exists, we can either use the latest intersection point or the point closest to the "outside" grid vertex of this edge. This first phase of the reparameterization process is simple and efficient and is inherently parallel; each model element can be processed independently.

During the deformation step, a T-snake "passes over" a set of grid triangle vertices (Figure 4.6a). Using the analogy of front propagation, specifically flame propagation, these grid vertices have been "burned".[4] We are able to determine and track the interior region of the T-snake by identifying and recording these burned grid vertices during the deformation step. Furthermore, these interior grid vertices unambiguously define the boundary of the model; they are used to continuously track the grid boundary triangles throughout the evolution of the model and hence determine the set of new model nodes used to form the T-snake boundary. By maintaining an interior region representation as well as a boundary representation of a T-snake, we are able to construct a characteristic function of an object $\mathcal{O}$. The characteristic function $\chi\mathcal{O} : \Re^n \rightarrow \{0, 1\}$ of an object $\mathcal{O}$, is defined as $\chi\mathcal{O}(p) = 1$ if $p \in \mathcal{O}$ and $\chi\mathcal{O}(p) = 0$ if $p \ni \mathcal{O}$.

Determining the set of grid vertices that have been burned during a deformation step uses a simple, robust and original classification algorithm. Each model element may have

---

[4] The inflation force described in section 4.2 pushes a T-snake in a direction normal to the snake at each model node. Curves evolved in this manners can develop shocks or singularities such as corners and self-intersections (Appendix A). Once these singularities develop, it is not clear how to evolve the curve beyond the singularities. We solve this problem by mimicing the physically correct behavior for a propagating flame front. This behavior is selected by adhering to a so-called *entropy condition* [74]: *if the T-snake is viewed as a burning flame, then once a particle is burnt it stays burnt.*

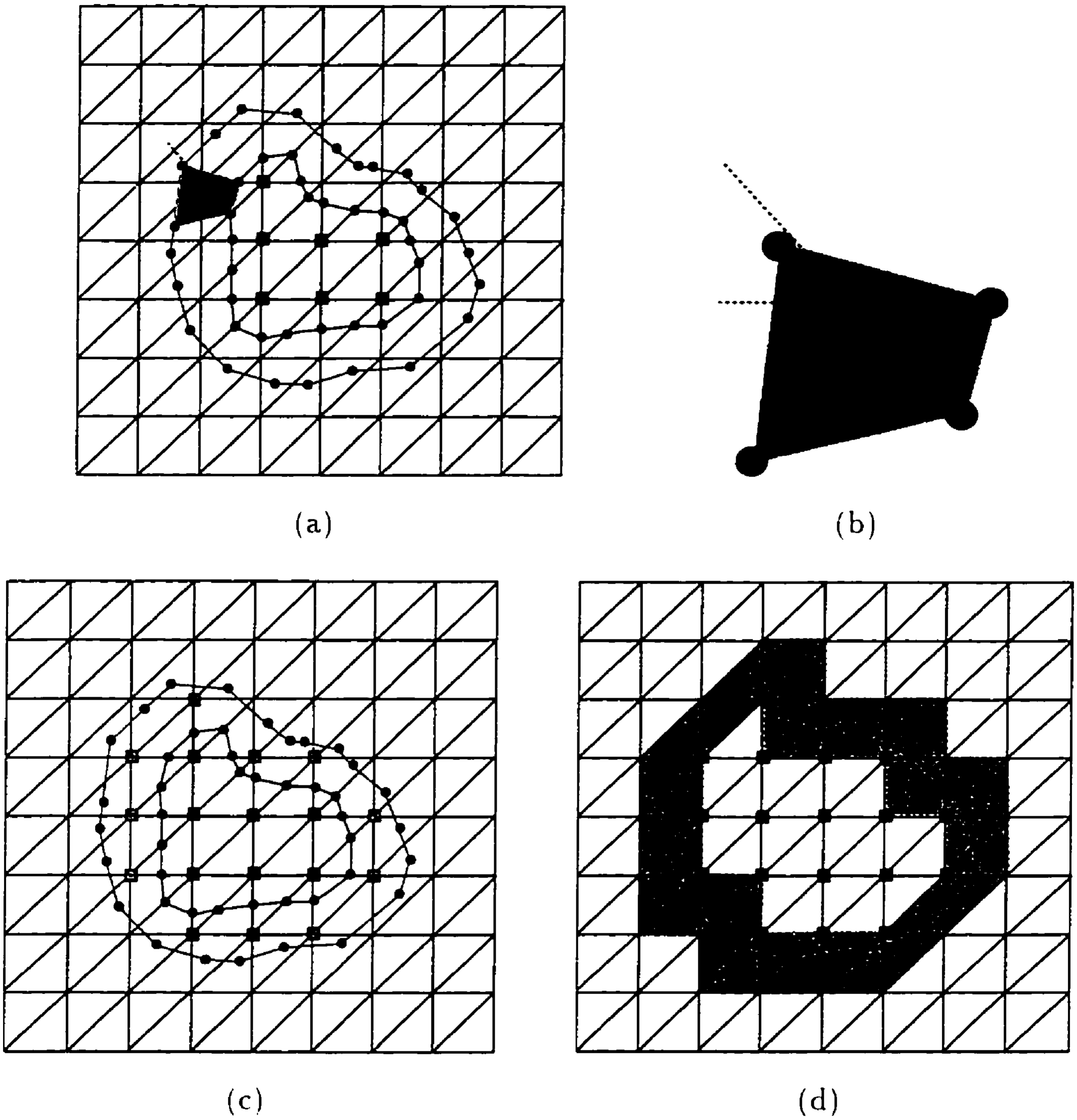


Figure 4.6: Phase 2 of T-snake reparameterization: (a) a model element of an expanding T-snake may pass over a grid vertex, (b) subspace partitioning (c) new grid vertices (light-shaded), (d) new T-snake.

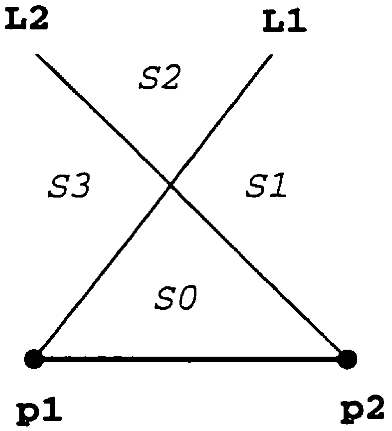


Figure 4.7: Formation of subspaces used in the labeling of a grid vertex.

passed over zero, one, or several grid vertices during a deformation step. For each model element, we form a bounding box around the current position and the new position of the element. The bounding box allows us to quickly determine the subset of grid vertices that may have been burned. For each of the grid vertices within the bounding box, we partition the image domain into four subspaces by forming two half-lines (Fig. 4.6a,b). The lines are formed by joining the two model nodes of a model element at their current position to a grid vertex. We then classify the two model nodes at their new positions into one of the four subspaces. This classification allows us to quickly determine whether the grid vertex in question has been burned. The classification algorithm essentially consists of several dot products and is extremely efficient and inherently parallel. Figures 4.6a-c, illustrate the second phase of the reparameterization process. In Figure 4.6c, the new grid vertices are shown in light gray. Finally, Figure 4.6d shows the new T-snake after both reparameterization phases have completed.

Classification Algorithm

This section describes a robust, efficient algorithm for determining if a model element has passed over or "burned" a grid vertex v during one deformation step. We begin by partitioning the bounded image domain into four subspaces (Figure 4.7) by forming two half-lines $L1$ and $L2$. Line $L1$ is formed using node $p1$ of the model element and v, and line $L2$ is formed using node $p2$ and v. During one deformation step, $p1$ and $p2$

move to a new location, $p1n$ and $p2n$ respectively, approximately in the direction of the normal defined at the model nodes $p1$ and $p2$. We restrict the maximum movement of a model node during one deformation step to be much less than the dimensions of the image domain. This restriction allows us to assume the movement follows a straight line path and prevents the formation of degenerate subspaces. We then classify $p1n$ and $p2n$ into one of the four subspaces, depicted in Figure 4.7. There are 2 model nodes and 4 subspaces for a total of 16 possible classification combinations (Figure 4.8(1)-(16)). The four points $p1, p2, p2n, p1n$ form a closed polygon Q. This polygon may be nonconvex in which case it can be considered to consist of two convex (triangular) pieces.

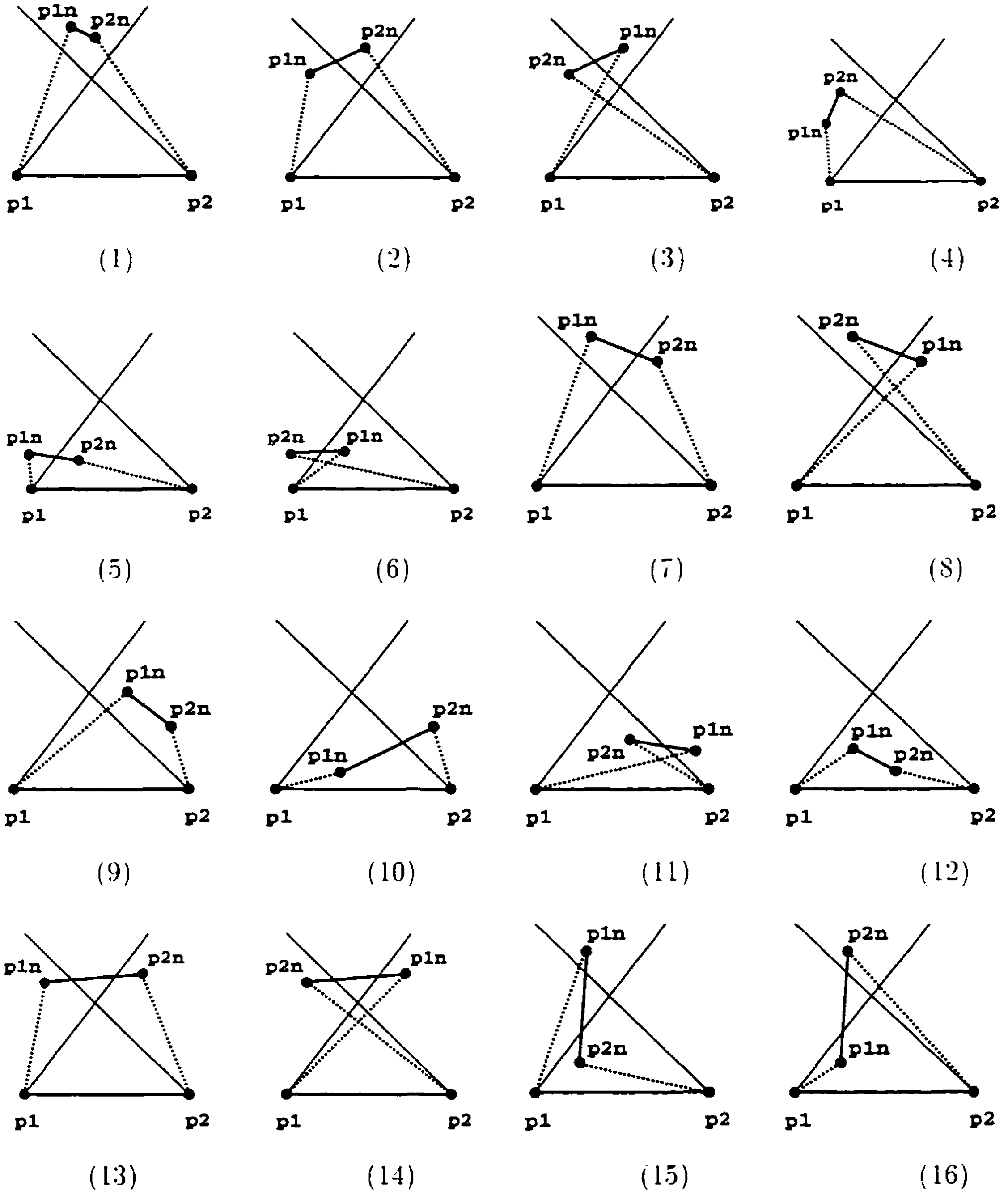
DEFINITION 4.5. A point p is inside polygon Q if a ray cast from p intersects exactly one edge of Q or exactly three edges of Q.

This definition is invalid when the ray passes exactly through an endpoint of an edge of polygon Q. We will assume that this situation is detectable and one edge of the polygon can be shortened slightly.

DEFINITION 4.6. A grid vertex v is labeled as "burned" if it is inside polygon Q.

According to these definitions, from Figure 4.8(1)-(16), cases 1, 2, and 7 would label the v burned, cases 3-6 and 8-12 would label v not-burned. Thus, by simply classifying $p1n$ and $p2n$ into one of the four subspaces, we can immediately label v for the majority of the cases. The classification algorithm consists of two inside-outside half-space tests each for $p1n$ and $p2n$. A half-space test essentially consists of a dot product between a point p and the *point-normal* equation of a line and is extremely efficient. Furthermore, a neighbor element of $\overline{p1p2}$ shares one of the half-lines, $L1$ or $L2$ (which are formed from the fixed grid vertex point v), and the maximum element movement restriction mentioned earlier guarantees that degenerate subspaces cannot be formed. This process results in stable numerical computations.

Cases 13-16 are ambiguous and require one additional test. For each of these cases, we perform an inside-outside half-space test for v using a line formed by the model element at its new location $\overline{p1np2n}$ to define the half-spaces. For cases 13 and 14, v must lie in the same half-space as $p1$ and $p2$ for v to be labeled as burned. For cases 15 and 16, v must lie in the same half-space as $p1$ and $p2$, respectively.



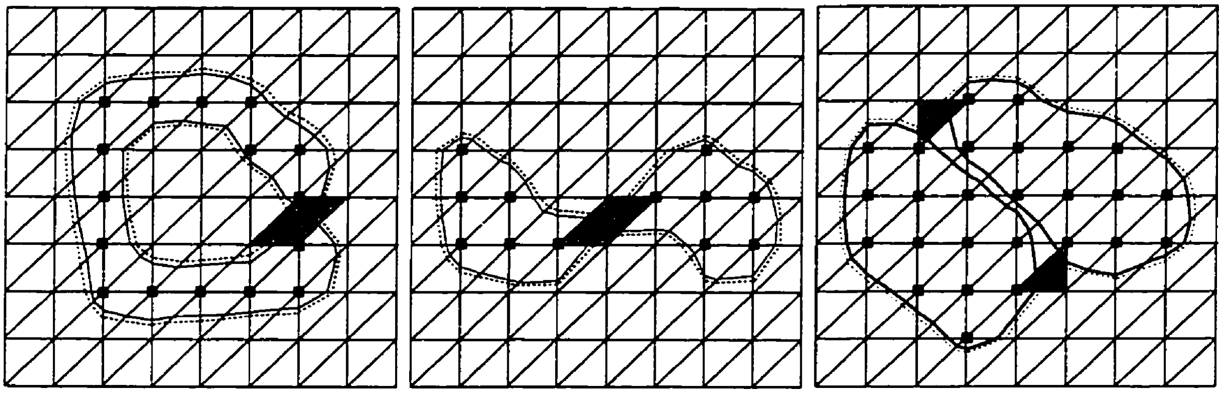
(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11) (12)

(13) (14) (15) (16)

Figure 4.8: Classification combinations.



(a) (b) (c)

Figure 4.9: Examples of T-snake topological transformations: (a) self-intersection, (b) shrinking and splitting, (c) merging. The resulting T-snake(s) after the transformations are shown as the dotted line(s). Node reconnections occur automatically in the shaded triangles so that inside and outside grid vertices are separated by a model element.

We claim that the classification algorithm is sufficient to unambiguously label a grid vertex point v as burned or not-burned. We present a simple, geometric proof of this claim in Appendix B.

4.3.3 Topological Transformations

When a T-snake collides with itself or with another T-snake, or when a T-snake breaks into two or more parts, a topological transformation must take place. In order to effect consistent topological changes, consistent decisions must be made about disconnecting and reconnecting T-snake nodes. The simplicial grid and the reparameterization process provides us with an automatic and unambiguous mechanism to perform reconnections. By tracking the interior grid vertices (and hence the boundary grid triangles), adhering to the entropy condition, and reestablishing the correspondence of the model with the grid after a deformation step, we can always unambiguously determine the boundary or "iso-contour" of the new T-snake(s). We simply compute new model elements from the signs of the grid vertices in each boundary triangle and from the intersection points computed in the first phase of reparameterization, such that the inside and outside grid vertices of these triangles are separated by the model element (Fig. 4.9). Thus, by mimicing the evolving level set of an implicit function, the simplicial grid and the reparameterization process guarantees that topological transformations are handled automatically, consistently and efficiently.

4.3.4 The T-Snake Algorithm

Unlike the level set evolution techniques which accede control to a higher dimensional implicit function, T-snakes retain an explicit parametric model formulation. The parametric formulation allows us to track and control the evolution of the T-snake. Consequently, reparameterizations can be performed very efficiently and constraints can be easily imposed on the model. The entire process is essentially reduced to computing a new model "region" — a boundary segment and interior grid vertices — generated by the movement of one model element during a deformation step. The union of these regions defines the new T-snake boundary and updates the interior of the T-snake. This process is directly controllable and parallelizable. The T-snake algorithm is as follows:

For each deformation step (N time steps):

1. For each time step:
   (a) Compute the external and internal forces acting on model nodes.
   (b) Update the node positions using equation (4.7).
2. Reparameterization Phase 1: Using the procedure described in Section 4.3.2, compute the grid intersection points for all model elements. These intersection points will become the new model nodes.
3. Reparameterization Phase 2: For each model element, compute and record the grid vertices "burned".
4. Using this set of burned grid vertices, determine the corresponding set of boundary grid triangles and compute all new model elements and nodes. Model nodes contain pointers to neighboring nodes. Model node neighbors (as well as element neighbors) are easily computed via adjacent boundary grid triangles.

5. For all current model elements, determine if the model element is still valid. A model element is valid if its corresponding grid triangle is still a boundary triangle. Discard invalid model elements and unused model nodes.

A T-snake is considered to have reached its equilibrium state when all of the model elements have been inactive for a specified number of deformation steps. Model element activity or movement is measured via the grid again using a flame propagation analogy. Model elements are assigned a "temperature" based on the number of deformation steps the element (and its corresponding boundary grid triangle) has remained valid. An element is considered inactive when its temperature falls below a pre-set "freezing point". Once a T-snake has reached equilibrium the simplicial grid can be discarded, if desired, and the model run as a classical snake. The internal energy constraints will then create more evenly spaced model nodes (Section 5.1.4).

4.4 Limitations of T-Snakes

Section 4.3 described the mechanism by which a T-snake evolves. In particular, to propagate a T-snake past singularities or shocks we adhere to an entropy condition: once a particle (in our case, a grid vertex) is burned, it stays burned. While this policy mimics physically correct behavior of a propagating front (which is consistent with our physics-based framework) and provides robust topological transformation abilities, the implication of this policy is that a T-snake can only expand or shrink as it evolves, but not both. We typically seed a T-snake inside an object and allow it to expand towards the object boundaries. Conversely, we can also surround an object with a T-snake and allow it to shrink and wrap itself around the object. This *deformation mode* limitation can affect the degree of user interaction. If a T-snake is expanding, the user cannot exert a force on the snake in an "inward" direction. In practice however, this restriction is rarely problematic. Typically user interaction with T-snakes involves "dragging" model nodes in the direction of evolution to pull the T-snake off of a spurious edge or into a narrow cavity partially "blocked" by an edge. Furthermore, a possible solution to this deformation limitation would be to reverse the direction of evolution occasionally for a

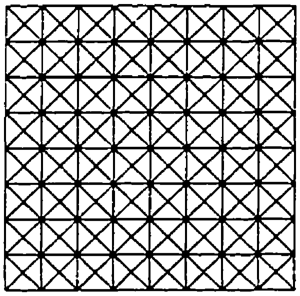


Figure 4.10: J_1 triangulation.

small time interval.

The resolution of the simplicial grid controls the degree of geometric flexibility of a T-snake. If an object of interest contains a narrow protrusion, the grid resolution must be fine enough to allow the T-snake to flow into it. On the other hand, the "body" of the object may be relatively smooth (i.e. other than the protrusion, the object boundary may not exhibit rapid shape variations). This situation results in an excessive number of model elements used to recover and describe the object boundary. One solution to this problem is to remove or merge model elements in regions of low shape variation in a post-processing phase. This topic has been addressed in 3D by several researchers [107, 38] and is not explored in this thesis.

The decomposition of a rectangular grid into triangles (Fig. 4.1) introduces diagonal edges on the rectangles. The orientation of these diagonal edges is arbitrarily chosen and affects the connectivity of a T-snake. For this reason, topological *correctness* (i.e. faithfulness to the geometry of the object boundary) of the T-snake is not guaranteed, only topological *consistency* (i.e. closed contours, without artifacts such as holes, are always generated). Typically, this orientation bias does not present a problem since the resolution of the grid is set fine enough to accommodate object feature size. Alternatively, the J_1 triangulation (Fig. 4.10) can be used to ameliorate this orientation bias problem (although it does not entirely eliminate it) at the expense of an increased number of model elements and algorithmic complexity. As mentioned previously, any affine cellular complex could be used as the T-snake grid. Topological disambiguation schemes may be required in some cases while others may increase algorithmic complexity or perhaps decrease the number of model elements. We have found a simplicial complex, using the Coxeter-Freudenthal triangulation, to be an excellent balance of mathematical theory, algorithmic complexity, and model compactness.

As mentioned previously, if a target object contains a very narrow feature, the grid resolution must be fine enough to allow the T-snake to flow into it. Furthermore, the grid orientation bias mentioned above also affects the minimum feature width that can be segmented. In general, image features that are orthogonal to the diagonal edges of the grid are the limiting factor. These features must be wider than the length of a diagonal edge (i.e. the grid meshsize δ) to allow their segmentation. Another factor that affects the minimum segmentable feature width w is the ratio of the inflation force strength to the strength of the internal forces. As this ratio decreases, w increases. Finally, other factors affecting w are the width of the (Gaussian) smoothing filter σ used in the extraction of image edge features (see section 2.1), image noise, and the local variation of the image intensity threshold T. Given a noise-free image and T-snake parameter values within a normal range, then in our experience, the minimum segmentable feature width w is approximately equal to 1.5δ.

Chapter 5

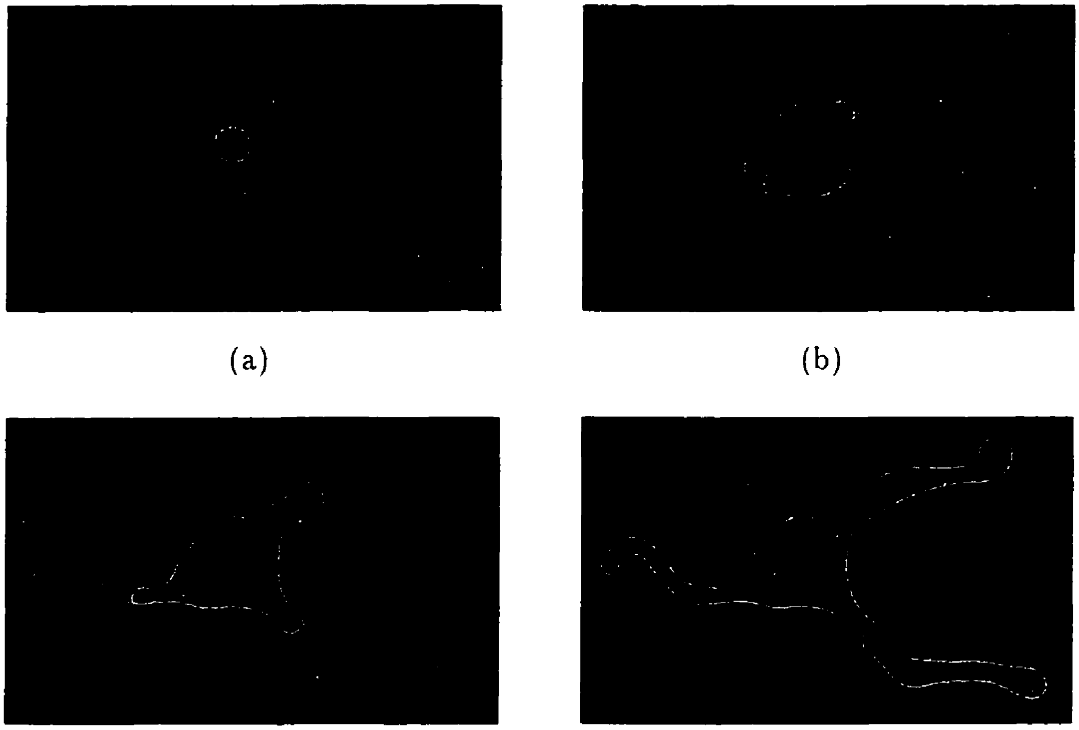
T-Snakes Features and Constraints

This chapter describes and demonstrates several features of T-snakes. In particular we illustrate, with examples, the ability of T-snakes to flow into complex shapes and adapt to object topology, the interactive capabilities of T-snakes, and several additional features arising from the simplicial grid. We then describe the various constraint mechanisms that can be imposed on T-snakes. Geometric constraints, topological constraints, statistical constraints, and global shape constraints are presented.

5.1 T-Snakes Features

5.1.1 Geometric Coverage and Intrinsic Parameterization

Using the simplicial grid to reparameterize a T-snake at each iteration of the evolution process creates a simple, elegant and automatic model subdivision technique. This process allows a T-snake to segment and reconstruct objects with significant protrusions, tube-like objects, or objects with bifurcations (Figures 5.1, 5.2). Furthermore, manifolds such as curves and surfaces have certain shape properties that are independent of any particular parameterization; the *intrinsic geometry* of these objects depends only on their shape in the domain. Embedding and defining T-snakes in terms of the simplicial grid removes any explicit parameterization and instead parameterizes the fitted T-snake in terms of this intrinsic local geometry. This property can significantly increase segmentation efficiency and reproducibility by making the T-snake relatively insensitive to its initial placement. A T-snake may be seeded practically anywhere within or surrounding a target object and still produce similar final segmentations with similar intrinsic



(a) (b)

(c) (d)

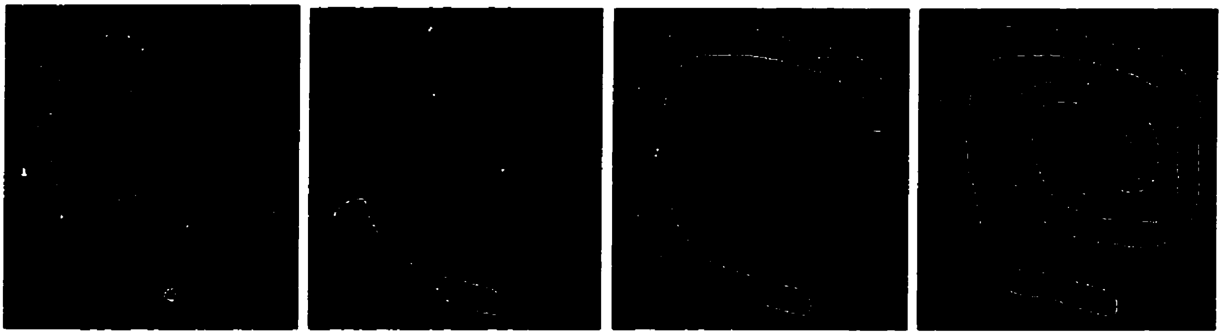
Figure 5.1: T-snake flowing into object with protrusions.

parameterizations (Figure 5.3).[1] Efficiency is increased in many scenarios because the time-consuming task of initializing a snake near the boundary of the target object is mitigated.

This automatic and intrinsic reparameterization property of T-snakes is important in the design of completely automatic segmentation techniques. It is possible to design explicitly parameterized snakes models that are capable of automatic global or local reparameterization [56. 46]. This can be acheived, for example, by checking along the entire length of the snake for any snake element that has become larger than some predefined maximum length and then dividing these elements into shorter ones of equal length [56]. In semi-automatic segmentation scenarios, this scheme can be quite effective since the snake is typically placed near the boundary of the target object. In automatic scenarios however, since the explicit parameterization of the snake is not directly related

---

[1] Noise and spurious image features can still affect T-snake behavior and, consequently, affect the result of the segmentation.



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

Figure 5.2: T-snake flowing along a spiral tube-like object.

to the geometry of the target object, different segmentation results may be generated by snakes placed in different initial positions, especially for objects with very complex shapes.

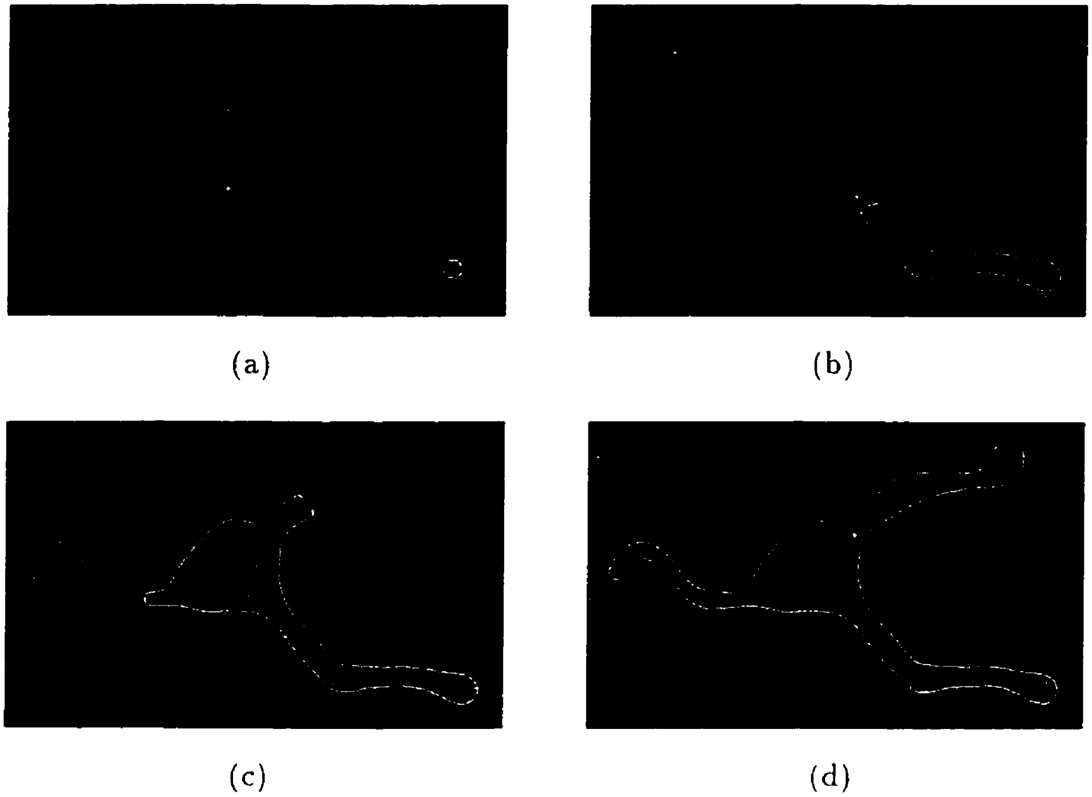
5.1.2 Topological Adaptability

As described in Chapter 4, the simplicial grid provides a mathematically sound framework for robust topological transformations. This feature allows a T-snake to seamlessly split or merge and adapt to the topology of the target object (Figures 5.4, 5.5). Topological adaptability combined with geometric flexibility can potentially significantly increase the automation of the segmentation process.

5.1.3 Multi-resolution Capability

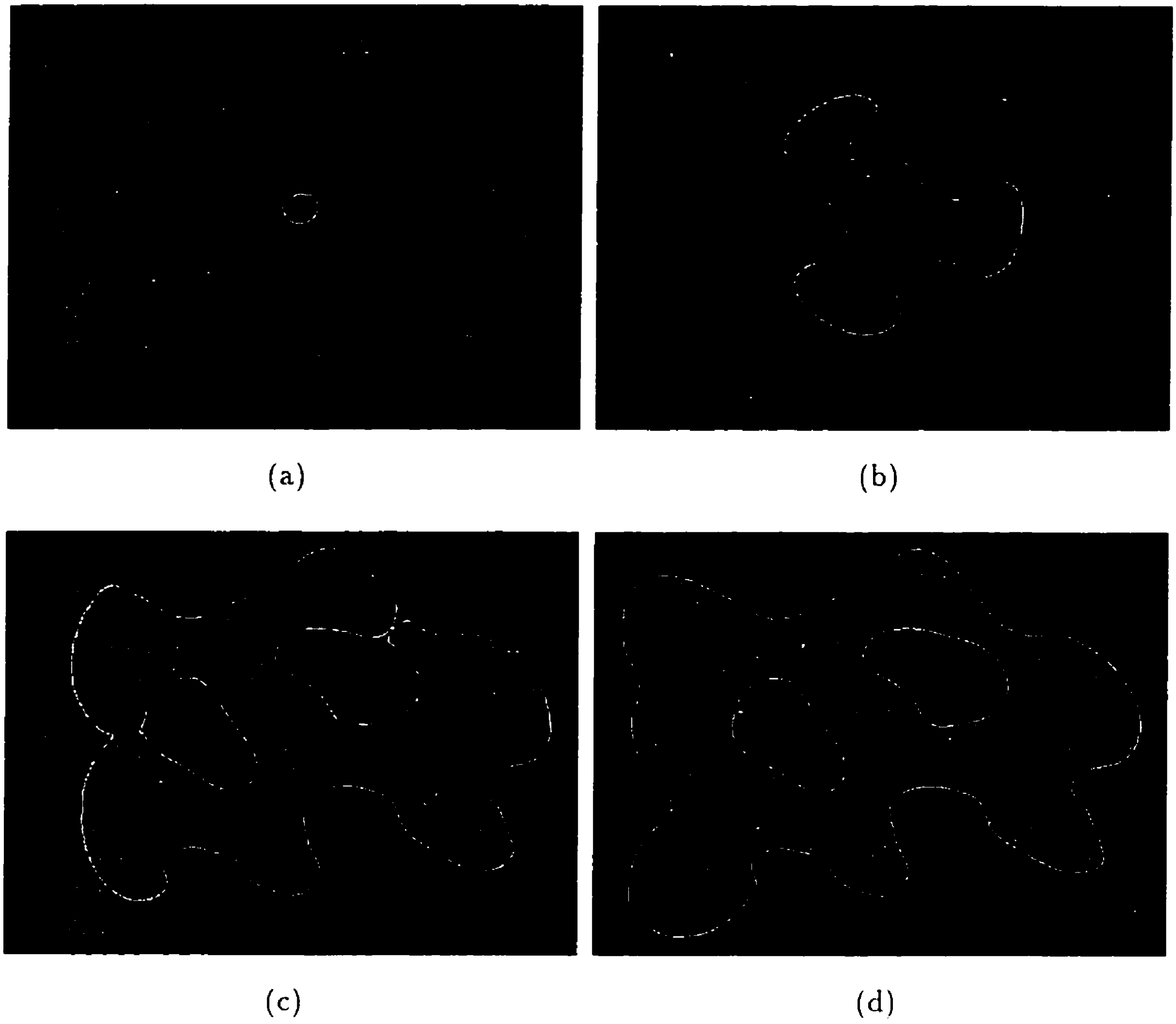
ACD-based deformable models are inherently multi-resolution. The size and number of model elements is directly determined by the grid resolution (Figure 5.6). Furthermore, if the grid resolution is restricted to multiples of an initial coarse resolution, the finer grids preserve the model nodes of the coarse resolution grids (Figure 5.7).

The multi-resolution feature of T-snakes and T-surfaces can be exploited in several ways. The grid resolution can be adjusted to generate a model tailored to expected image feature size or to generate a desired accuracy. We can also compute an image pyramid that provides a set of images at decreasing scales. Coarser resolution models can then be run on large scale images and the result can be used as input to finer scale images.



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

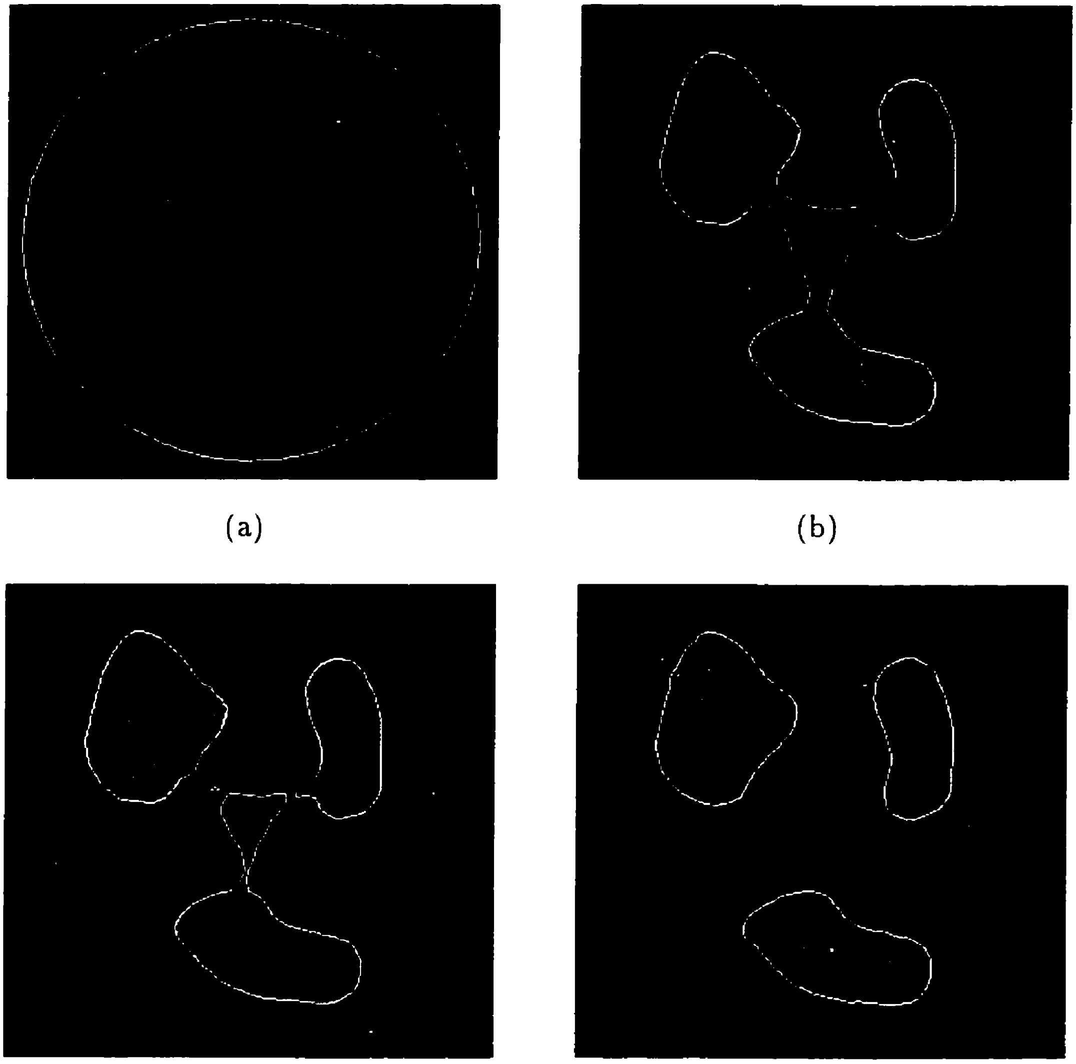
Figure 5.3: Initialization insensitivity of T-snakes. A T-snake may be seeded essentially anywhere within or surrounding an object.



(a) (b)

(c) (d)

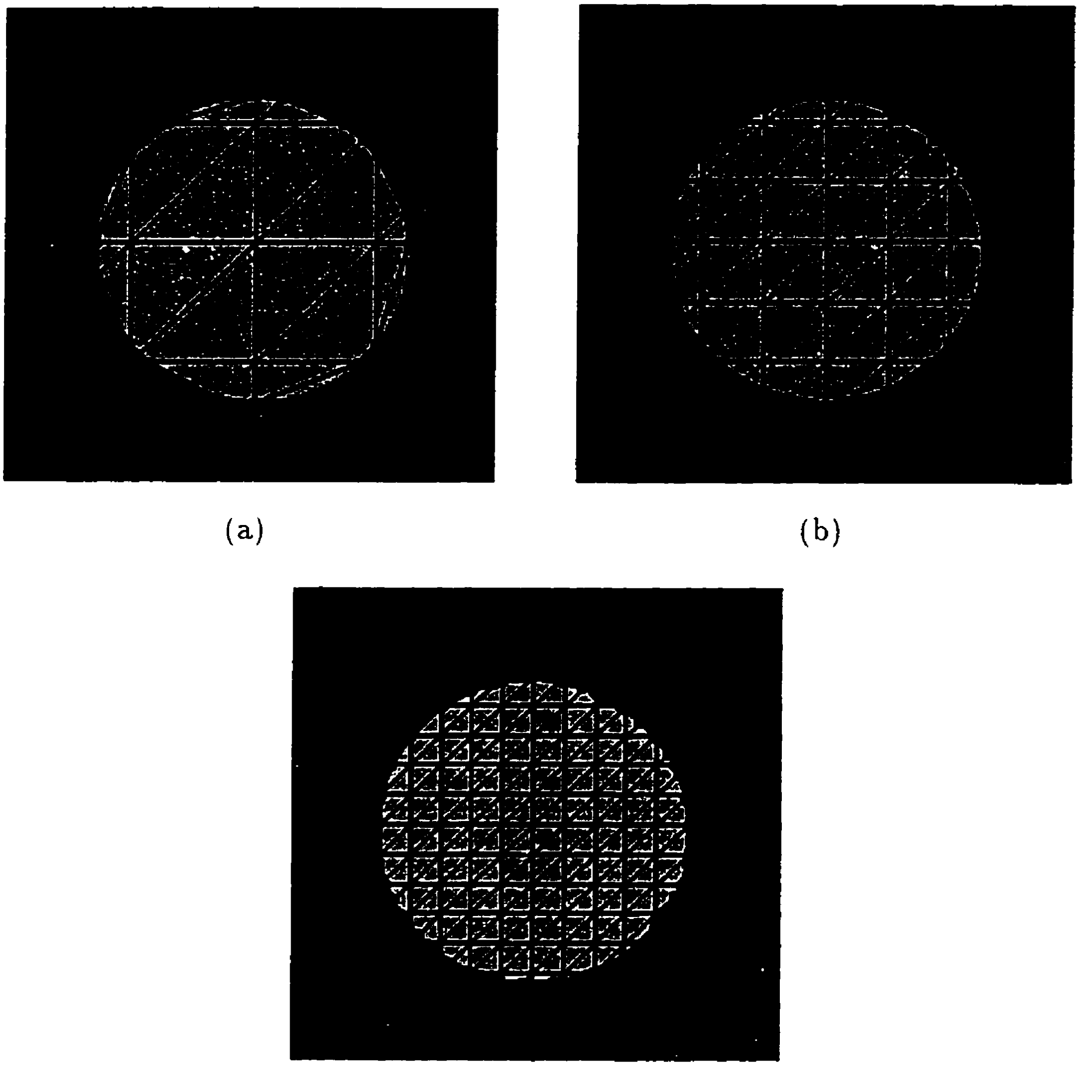
Figure 5.4: T-snake flowing and splitting to segment object with complex topology.



(a) (b)

(c) (d)

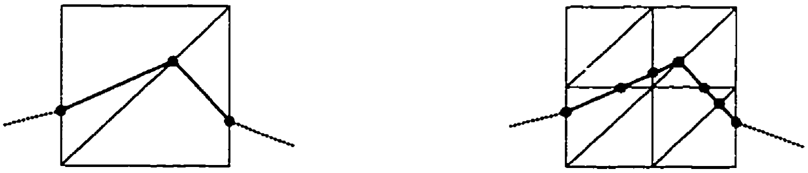
Figure 5.5: T-snake shrinking, wrapping and splitting around multiple objects.



(a) (b)

(c)

Figure 5.6: T-snake representation of circular object using 3 grid resolutions: (a) 4 × 4 (b) 8 × (8 c) 16 × 16.



(a) (b)

Figure 5.7: T-snake nodes on coarse-resolution grid are preserved on finer resolution grid.

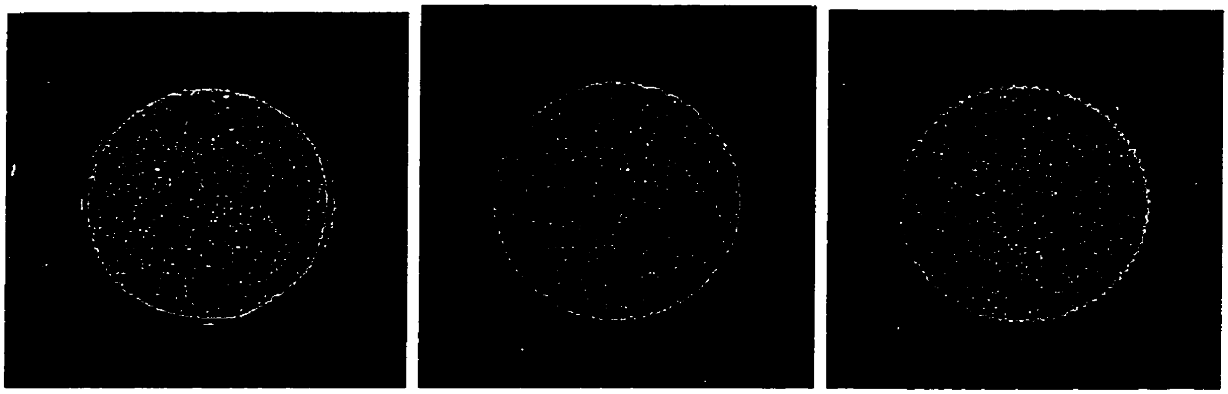

(a) (b) (c)

Figure 5.8: Redistribution of snake nodes after discarding simplicial grid. Grid resolution: (a) 4×4 (b) 8×8 (c) 16×16.

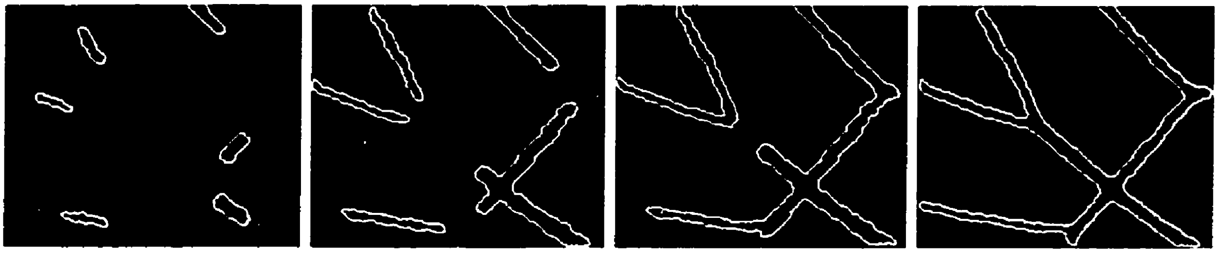
Although this procedure is not explored further in this thesis, we believe it has potential for significant improvements in segmentation efficiency for many applications.

5.1.4 Conversion to/from Classical Snakes

Converting a T-snake to the traditional parametric snakes model is simply a matter of discarding the simplicial grid and disabling the reparameterization process. This conversion can be performed at any time during the model evolution. The internal forces of the snake will then act to redistribute the model nodes to produce a more even distribution. (Figure 5.8).

Converting a closed snake back to a T-snake requires the specification of a point inside the snake as well as the superposition of the grid and the resumption of the reparameterization process. Without the specification of an inside point it is possible that the conversion will produce an incorrect T-snake. This situation arises when a closed snake crosses itself. Conversion from a conventional snake to a T-snake can also be carried out at any time during the model evolution.

The T-snake conversion feature can be very useful in some interactive segmentation scenarios. The viscoelastic T-snake can be "slippery" when the user attempts to pull T-snake nodes with the mouse. It is also possible to break off small pieces of a T-snake if excessive interactive force is used. The "rubber-band"-like classical snake exhibits more



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

Figure 5.9: Segmentation of the blood vessels in angiogram of retina using using multiple T-snakes.

predictable behavior under interactive control. Converting to and from the two models allows a user to dynamically fine-tune sections of the T-snake fit.

5.1.5 Multiple T-Snakes

Multiple T-snakes can be statically or dynamically created (or destroyed) (Figure 5.9). This feature can be useful in several scenarios. Firstly, multiple T-snakes can evolve concurrently on parallel machine architectures to improve segmentation efficiency. Secondly, users can seed T-snakes on several objects or on part of an object that is blocked by an edge to the object body. Finally, target objects can be automatically identified, seeded with multiple T-snakes, and then automatically segmented.

Collision Detection and Avoidance

As described in Chapter 4, we keep track of the interior region of a T-snake by "turning on" any grid vertices the T-snake passed over during its evolution. This feature can be used to perform efficient collision detections among multiple T-snakes. When segmenting multiple anatomic structures with common or adjacent boundaries using multiple T-snakes, an efficient collision detection and avoidance scheme could be utilized as an extra constraint in the segmentation process. By assigning different values to the grid vertices of each T-snake, we effectively give the T-snakes a unique identifier (Fig. 5.10). As a T-snake deforms, it may attempt to move into boundary grid triangles containing "on" vertices. A simple check can then be performed to determine the identity of the "on"

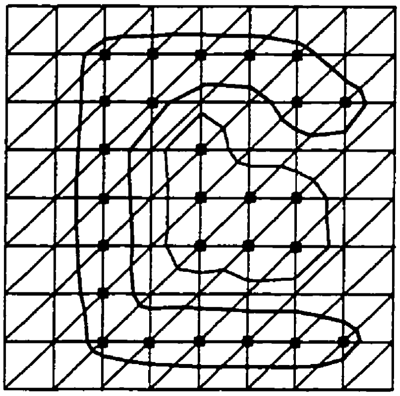
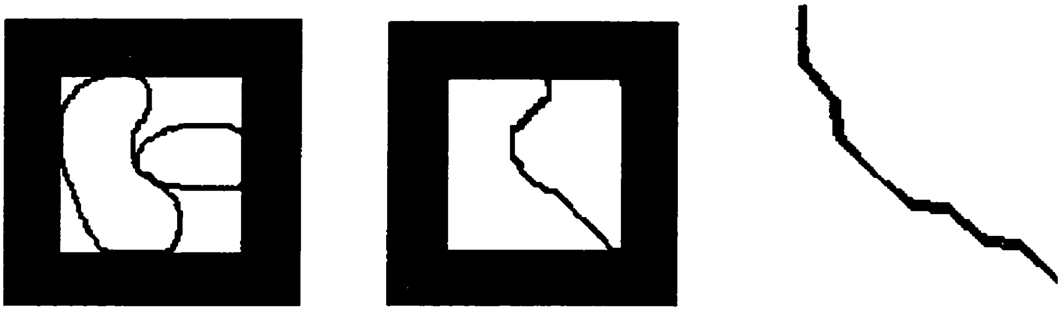


Figure 5.10: Collision detection and avoidance using model interior region "identifiers". A T-snake cannot move into "territory" already occupied by another T-snake.



(a) (b) (c)

Figure 5.11: Two models colliding, detecting and avoiding intersection.

vertices. If the grid vertices belong to another T-snake, various "avoidance" strategies can be used such as applying a repelling force to the intersecting nodes of the colliding T-snake to push it out of the occupied region (Figure 5.11).

We demonstrate the use of the model collision detection feature with a simple example. A sagittal slice of a preprocessed MR brain image was manually segmented into four anatomical regions and the pixels in each region were assigned a constant intensity value. We initialized two T-snakes in regions 2 and 4 and recovered the shapes of these regions (Figure 5.12). We then initialized two T-snakes in regions 1 and 3 and allowed the snakes to evolve, using only a minimum intensity threshold as the external image force. When these T-snakes attempt to flow into the brighter regions 4 and 3 they collide with the initial T-snakes and are forced to assume the shape of the common boundary regions

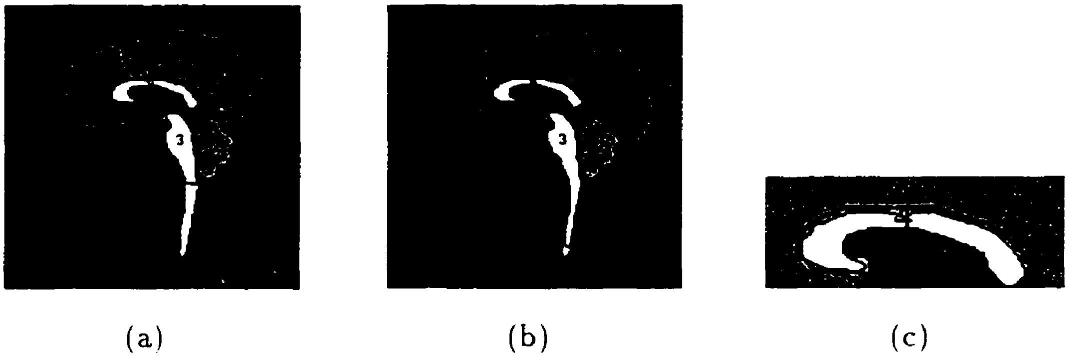

(a) (b) (c)

Figure 5.12: Reconstruction of preprocessed anatomical regions of the brain using multiple models. The collision detection feature was used as an extra constraint when reconstructing regions 1 and 3.

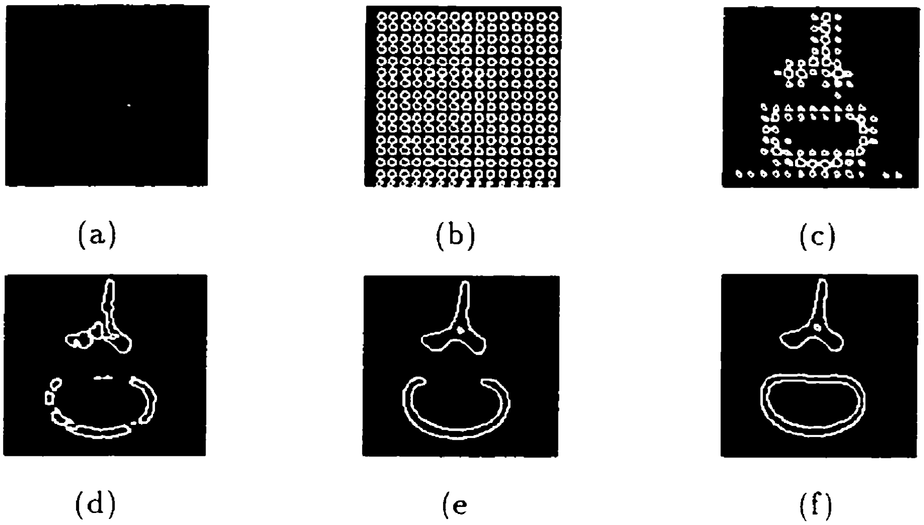
(Fig. 5.12c).

T-snake "Seeds"

We can make use of the multiple T-snakes feature in an interesting approach to automatic segmentation. The automatic segmentation of images containing multiple objects of interest, objects embedded inside other objects, or objects containing holes creates an initialization problem that cannot be resolved using a single T-snake. If the T-snake is initialized such that it surrounds the objects of interest and then shrinks and split around these objects, only the outer boundaries will be recovered. Conversely, to grow a T-snake from within each object requires prior knowledge of, or automatic identification of the objects. In our approach, inspired by [97], we uniformly distribute a set of small T-snake "seeds" over the image domain. These T-snakes then progressively expand, shrink, merge, and/or split to recover larger and larger regions of each object, including both the inner and outer boundaries (Figs. 5.13a–f). If there is no object of interest to attract a T-snake, it will shrink and eventually disappear. This technique results in a single stage process and requires no user interaction.

5.1.6 Model Element "Cooling" Process

We incorporate a model element "cooling" process into the T-snake implementation. As described in Section 4.3.4, model elements are assigned a temperature based on the num-



(a) (b) (c)

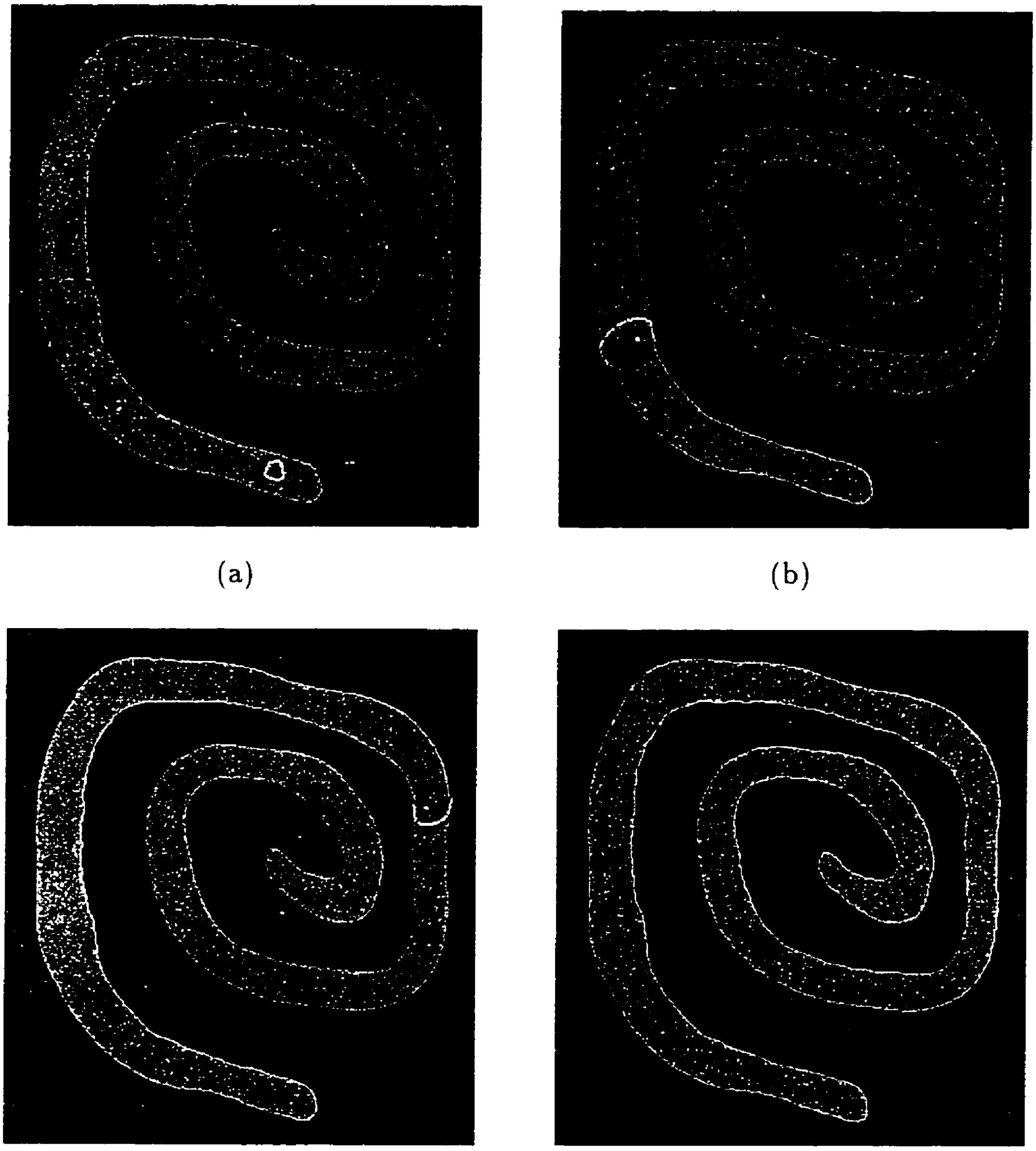
(d) (e (f)

Figure 5.13: T-snakes seeds growing, shrinking, merging, splitting and disappearing to automaticlly recover vertebral phantom parts.

ber of deformations steps the element (and its corresponding boundary grid triangle) has remained valid. The temperature attribute provides an excellent measure of model element activity or movement. When the temperature of an element falls below a user-set freezing point, the element is removed from the computational process and stored in a table. This adjustable mechanism allows the system to maintain a small, manageable computational burden for many segmentation scenarios. Figure 5.14 illustrates the cooling process. As the T-snake flows along the spiral-shaped object, only a few model elements of the "flame" are active (hot) at one time. The remaining elements are inactive (frozen) and do not contribute to the computational load. In this example, a 40 × 40 cell grid is used to segment the spiral from the 300 × 340 pixel image. The segmentation completes in a few seconds on a Silicon Graphics Indigo workstation.

5.1.7 Interactive Control

The affine cell decomposition framework allows T-snakes to maintain the intuitive interactive capabilities associated with classical snakes. Users can exert attraction or repulsion forces using "volcanos", "magnets", and "anchored springs" [47]. For example, figure 5.15 shows a T-snake being pulled in the direction of the mouse. As mentioned previously, the viscoelastic T-snake can be "slippery" when using mouse driven forces. In the example



(a) (b)

(c) (d)

Figure 5.14: T-snake flowing along a spiral tube-like object. The active part of the T-snake is in white, the inactive part is black.

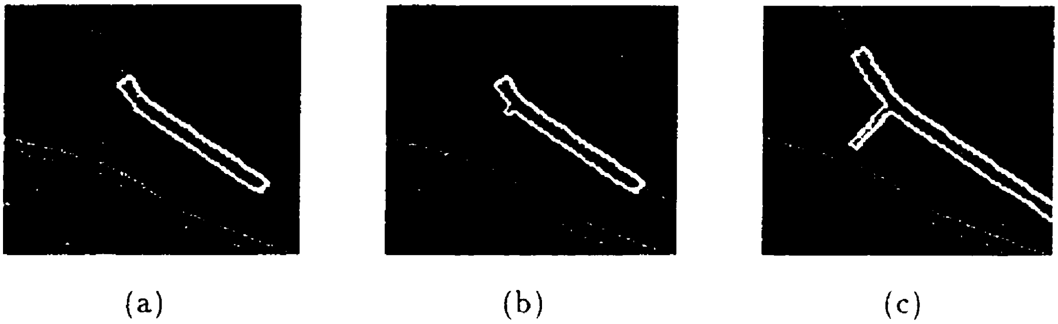


Figure 5.15: T-snake flowing along retinal artery is "pulled" into smaller branch by user.

presented, the mouse spring has slipped a little along the T-snake during the application of the spring force as the T-snake begins to flow into the smaller branch. Other useful T-snake interaction mechanisms are described in Section 5.2.1.

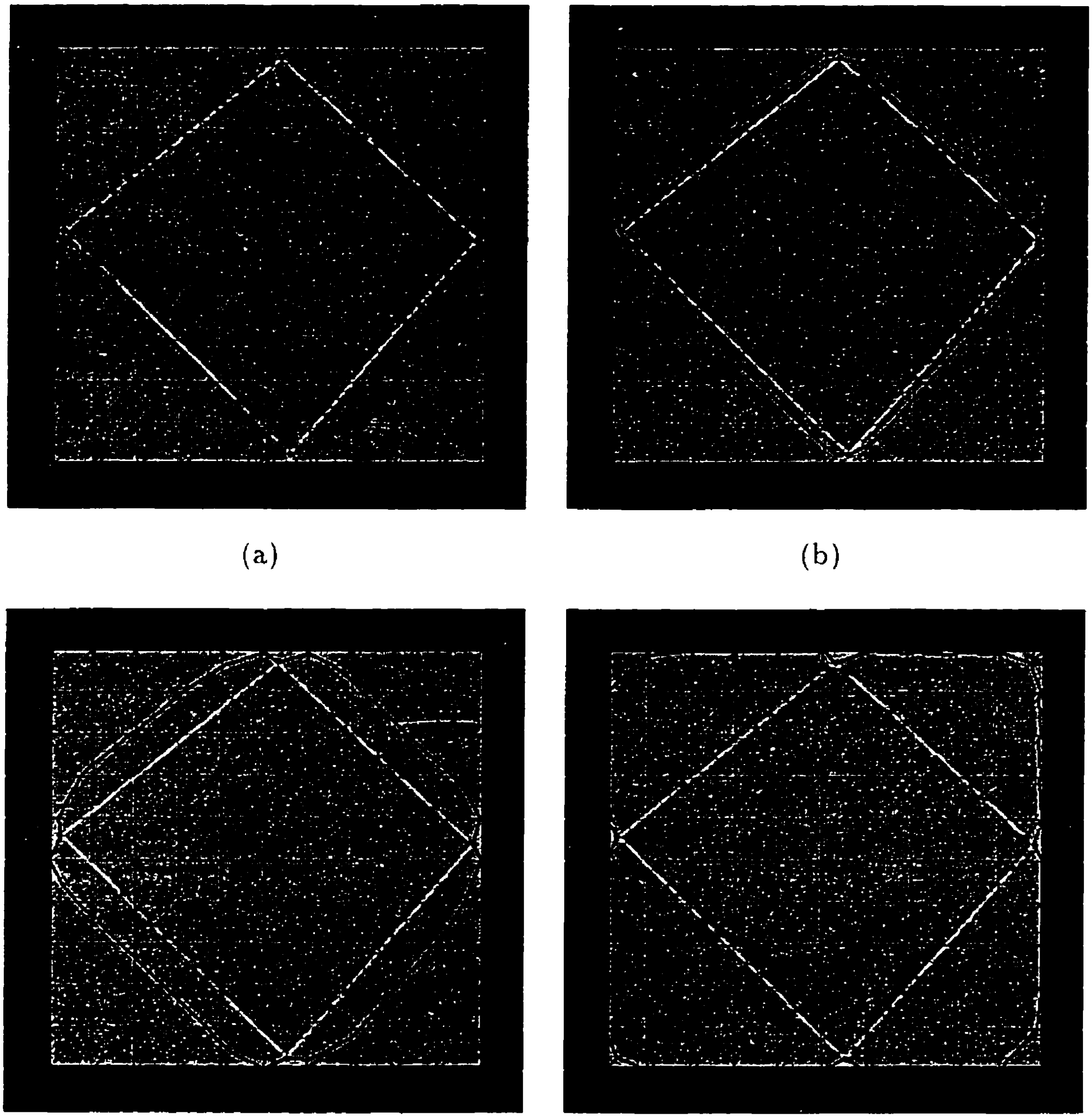
5.2 T-Snakes Constraints

The T-snakes features described above combine to create a powerful, semi-automatic segmentation technique. The addition of the constraints detailed in the following section provides T-snakes with the potential for completely automatic medical image analysis. Soft constraints are incorporated through the T-snakes physics-based formulation. Hard constraints are easily incorporated through the ACD framework and through the parametric model formulation that maintains direct control of T-snake evolution.

5.2.1 Geometric Constraints

Geometric constraints are incorporated into the T-snakes model either as soft constraints to be satisfied approximately or as hard constraints that must never be violated. Soft constraints are incorporated into the T-snakes physics-based formulation as energy functions or force functions. An example of a soft point constraint was described in the previous section through the mouse-driven spring force.

Hard fixed-point constraints are easily incorporated into the ACD framework. The user begins by specifying a series of points with the mouse or some other input device on or near the boundary of the target object (Figure 5.16a). These points are connected to form a closed polygon and the polygon is converted to an initial T-snake. Each of these



(a) (b)

(c) (d)

Figure 5.16: Example of geometric "hard" fixed point constraints.

Figure 5.17: CT image slice of LV and edge-detected version.

fixed points lies within a grid boundary triangle. Instead of forming one model element in each of these boundary triangles the fixed points are used to create two elements. using a fixed point to form a shared model node for the two model elements. Model nodes created from fixed points are frozen (i.e. unresponsive to forces). To ensure that boundary triangles containing fixed points remain valid boundary triangles throughout the T-snake evolution. at least one vertex of these boundary triangles must remain "off" or not "burned". By enforcing this simple policy we ensure that the fixed points always remain valid model nodes of the T-snake. Figures 5.16b-d show a T-snake constrained to pass through four fixed points as it flows and conforms to a square-shaped object. The four points were positioned slightly away from the boundary of the object for illustrative purposes. Currently we restrict the number of fixed points per grid triangle to one.

Other useful hard geometric constraints include barriers or "forbidden zones". For example. the user can place a circle or ellipse in the path of a T-snake. If a T-snake node enters one of these implicitly defined zones. it can either be instantly frozen or the model element cooling process will quickly freeze elements connected to the node. This constraint forces the T-snake to take on the shape of the impacted region of the circle or ellipse. allowing the user to "shore-up" object boundary sections with sparse edges.

5.2.2 Statistical Constraints

Section 4.2.2 described the external forces used to drive a T-snake towards object boundaries and lock it onto object edges. Unfortunately it is often the case that image edge forces (equation 4.6) are not quite strong enough to overcome the inflation forces, result-

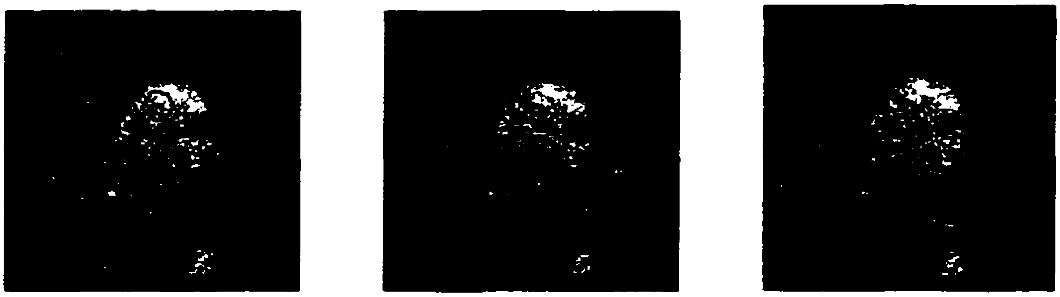
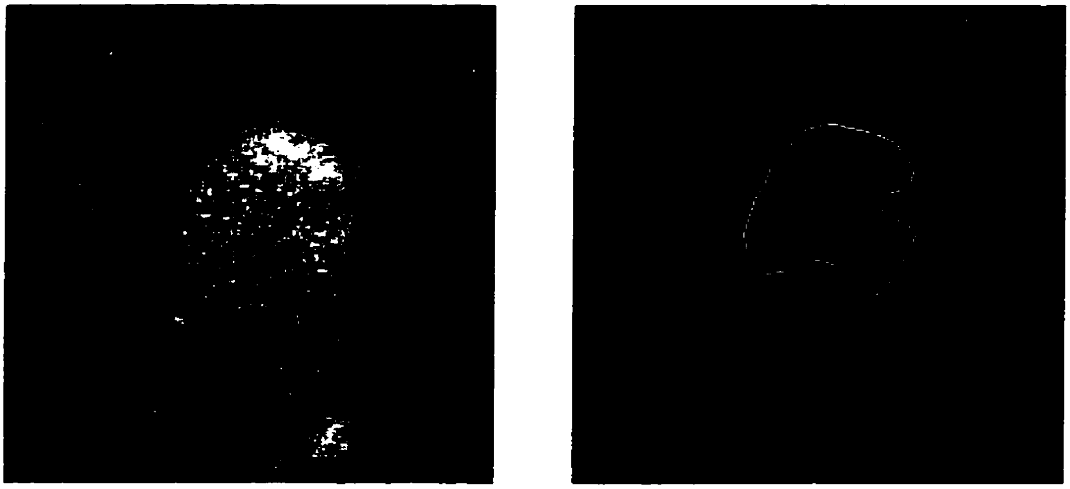

(a) (b) (c)

Figure 5.18: T-snake segmenting LV image slice using statistically-weighted inflation force and image edge forces.

ing in a "leak" of the T-snake into regions surrounding the target object. Consequently, interactive guidance or intervention may be required to guarantee a good result. Using region based statistics to weight the inflation force can effectively address this problem and thereby increase model automation. Region based statistics can be incorporated into the inflation force by extending the functional F of equation 4.6

$$\begin{aligned} F(I(x,y)) &= +1, \quad |I(x,y)-\mu| \leq k\sigma, \\ F(I(x,y)) &= -1, \quad |I(x,y)-\mu| > k\sigma. \end{aligned} \tag{5.1}$$

where μ is the mean image intensity of the target object, σ the standard deviation of the object intensity and k is a user defined constant. The statistical values, μ and σ, are typically computed or known *a priori*. Figure 5.17a shows a CT image slice of a canine heart. The bright region is the left ventricle (LV). In this particular slice, notice that the LV intensity diminishes considerably in some regions resulting in very weak edges (Figure 5.17b). Figures 5.18 a-d show the T-snake segmenting the LV using a statistically weighted inflation force. Figures 5.2.2a,b provide an enlarged view. Notice that the area around the papillary muscles (the indentation in the upper right part of the T-snake) has been correctly segmented whereas this area has been smoothed- over when only a single threshold value is used to weight the inflation force (Figure 5.20).



(a) (b)

Figure 5.19: Close-up view of LV segmentation: (a) image intensity slice, (b) edge-detected image slice.

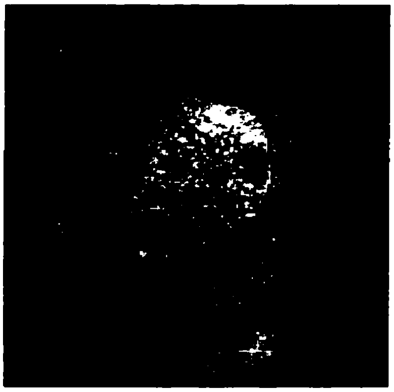


Figure 5.20: Segmentation of LV without incorporating image intensity statistics.

5.2.3 Topological Constraints

A promising technique for the analysis and interpretation of medical images is skeletonization. Automating the skeletonization process is difficult however because thinning algorithms are highly sensitive to noise. Providing a smooth segmented object to the thinning algorithms which is guaranteed to have the topology of the real object can potentially result in more robust skeletonizations. Classical parametric snakes models can produce smooth object segmentations but it is difficult to guarantee that no self-intersections of the fitted snake model have occurred. Consequently during discretization. incorrect object topologies can be generated. The ACD framework. on the other hand. provides a robust mechanism to identify and prevent self-intersections (and therefore possible topology changes) of the T-snake. The result is a simple. efficient technique for imposing a global topological constraint on a T-snake that guarantees the topology of the fitted T-snake matches that of the real object.

Topology Preserving Deformations

We implement the global topological constraint using topology-preserving T-snake deformations. The T-snake has an initial topology of a circle and the evolution of the model is performed using a sequence of these deformations. We are able to efficiently perform the topology-preserving deformations since our model is defined in terms of a simplicial cell complex.

During one deformation step. a T-snake is deformed (expanded or contracted) and a finite number of grid vertices will be burned. The addition of these grid vertices to the current set of burned vertices defines a new T-snake boundary. We would like to ensure that these additional grid vertices cannot change the T-snake topology. Therefore. before a model element l burns a grid vertex v during a deformation step. we validate the deformation of l. If the tentative deformation of l is deemed invalid. then the simplest strategy is to cancel its deformation. However. this strategy may undermine the dynamic properties of the T-snake since some model elements are temporarily frozen for potentially several deformation steps. A repulsion force can therefore be calculated as the sum of internal and *reversed* inflation forces. If the deformation of the element is still invalid

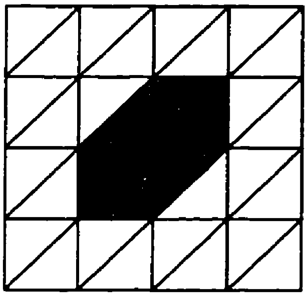


Figure 5.21: At least 6 T-snake model elements are needed to contain a grid vertex.

despite the addition of the repulsion force, the deformation is cancelled.

We begin the validation process by assuming that none of the grid vertices connected to the target ("un-burned") vertex v by a grid triangle edge will be burned during the current deformation step. We observe that there exists a path of grid triangle edges from an unburned grid vertex to all other unburned grid vertices, where no grid edge intersects a T-snake element. Otherwise there would exist an "island" of unburned grid vertices, indicating that a topology change of the T-snake had already occurred (all burned grid vertices are connected in a similar manner). We can therefore assume that some — but not all — of the vertices connected to v may already have been burned.

Given any grid vertex, the smallest number of model elements needed to form a closed polygon containing the vertex is 6 (Figure 5.21). Therefore we check the 6 grid triangles surrounding v and extract any and all model elements present in these triangles. Model element l will, of course, be a member of this set. We then perform a breadth first traversal of these extracted elements. That is, beginning with a seed element (l) we visit its two neighboring elements. If either of these elements is a member of the set of extracted elements, it is placed on a queue. We then take the first element from the queue, visit its two neighbor elements (if they have not already been visited) and again, place any neighbor on the queue that is a member of the extracted element set. This process is repeated until the queue is empty or all of the extracted model elements have been visited.

If all of the extracted model elements were marked as visited during the traversal, then these elements are all neighbor elements; they form a contiguous section of the T-snake with a maximum path "length" of 6 model elements. In this case, the defor-

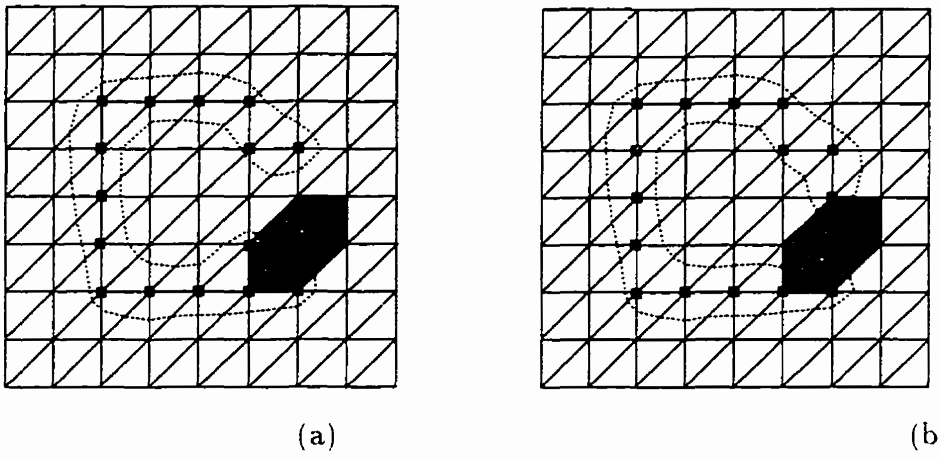


Figure 5.22: Example of topology-preserving deformation. In (a), the model elements in triangles A, B, C, F form a contiguous section of the T-snake and cannot cause a topology change when the grid vertex (light-shaded square) is burned. In (b), if element A, for example, burns the grid vertex, a topology change will occur, forming two T-snakes (one inner T-snake and one outer).

mation of l during this deformation step is deemed valid since the extracted elements cannot form a closed circuit around v (Figure 5.22a). This observation follows from our assumptions that at least one of the grid vertices directly connected to v remains unburned. If the extracted elements do not form a contiguous set then the deformation of l is deemed invalid. In this case, a topology change of the T-snake will occur (Figure 5.22b). Since we do not actually update model element positions until after phase 2 of the reparameterization process (Section 4.3.2), the validation process is deformation step dependent. For example (Figure 5.23) during one deformation step model element A may validly burn grid vertex 1 while model element B validly burns vertex 2. This would result in a topology change, with a small T-snake surrounding vertices 3 and 4. For this reason, we define two model element deformation restrictions that ensure a model topology preserving deformation:

1. A grid vertex cannot be burned during the current deformation step if any of the 6 grid vertices connected to it have been burned during this deformation step.

2. A grid vertex v cannot be burned by a model element A if

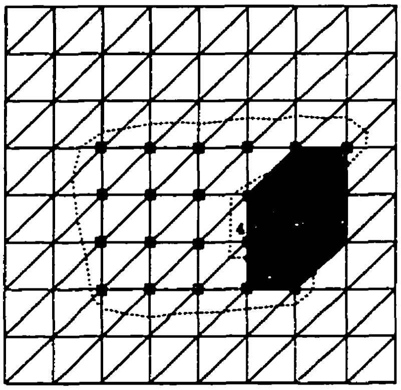


Figure 5.23: Example used to illustrate the time dependency of the model element validation proces. During one deformation step, element A may validly burn vertex 1 and element B may validly burn vertex 2. A topology change will occur resulting in two T-snakes, one inner surrounding vertices 3 and 4, and one outer T-snake.

(a) there exists another model element B whose grid triangle is one of the 6 grid triangles surrounding v,

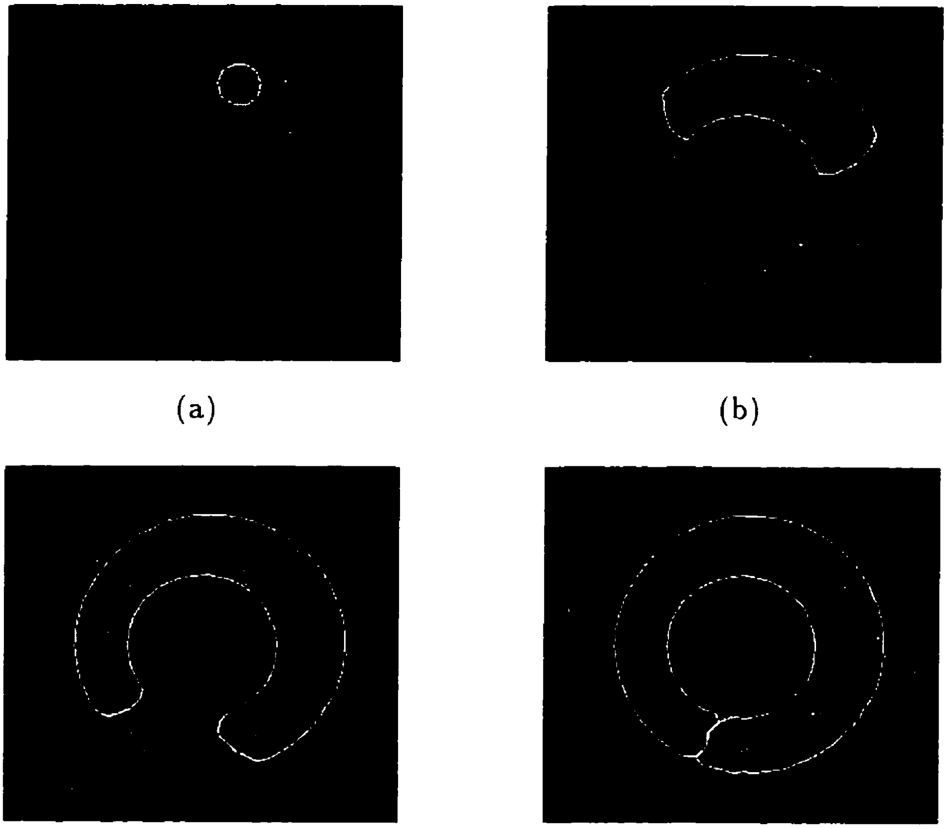
(b) there does not exist a connected path from A to B through all other model elements extracted from the 6 grid triangles surrounding v.

Topological Constraint Examples

Figures 5.24-5.26 demonstrate the topological constraint mechanism with a series of synthetic examples. In the first series (Figures 5.24a-d) a model is seeded within a donut shaped object. The model flows into the object, collides with itself, and is unable to interpenetrate. Figure 5.25a shows a closeup of the collision region. Figure 5.25b shows the same object but using a finer resolution grid. Figures 5.26a-d,e-h depict two more examples using two synthetic data sets presented in Section 5.1.2 for comparison.

5.2.4 Global Shape Constraints

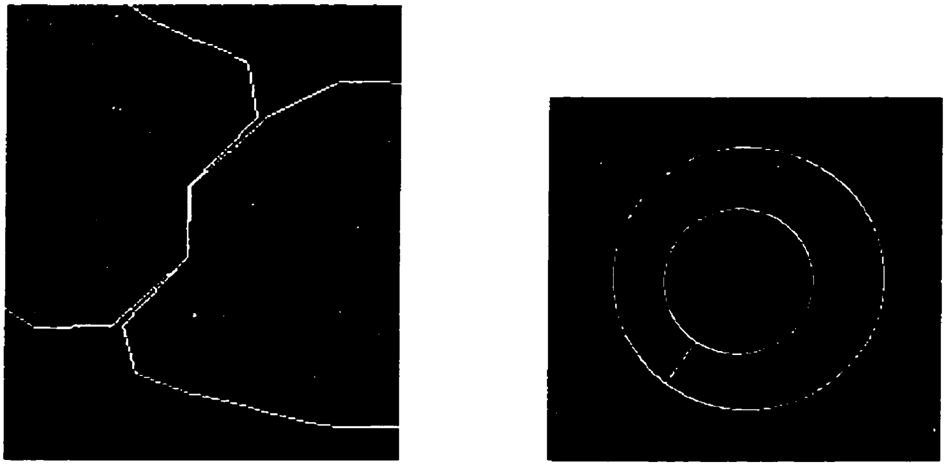
In this section we present a technique for imposing global shape constraints on T-snakes. We use multiple deformable superquadric shape templates (referred to as "cells") that are fitted to pieces of the target object and are constrained to overlap (Figure 5.27). In Chapter 3 we described the recent work of several researchers involving superquadric



(a) (b)

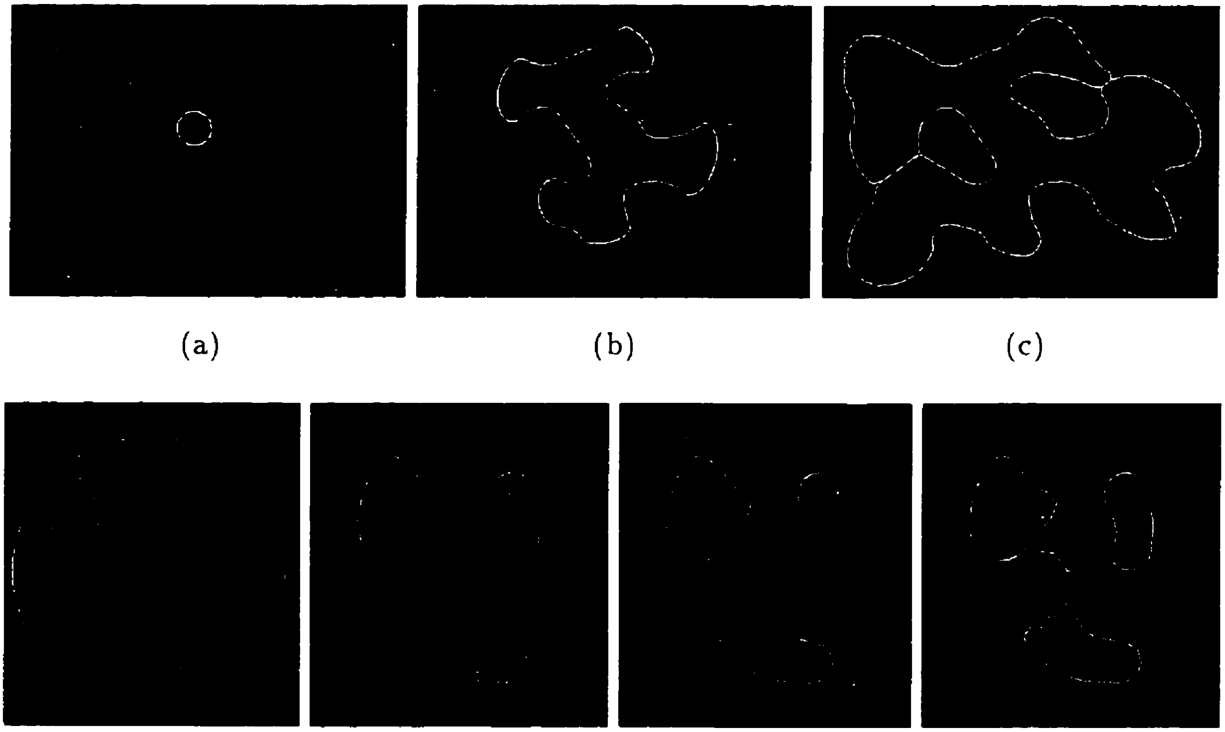
(c) (d)

Figure 5.24: Example of T-snake topological constraint. The T-snake flows around the donut-shaped object recovering its shape. It then collides with itself and is unable to interpenetrate.



(a) (b)

Figure 5.25: A close-up view of the self-collision zone. In (b) the same example is repeated with a finer resolution grid.



(a) (b) (c)

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

Figure 5.26: Examples demonstrating T-snake topological constraint.

Figure 5.27: Superquadric "cells" fitted to tubular-shaped object.

shape templates as a method of incorporating *a priori* knowledge into shape recovery problems. The deformable superquadric cells described here are modified versions of the dynamic deformable superquadric models developed by Metaxas and Terzopoulos [102, 68]. Appendix C provides a detailed description of our superquadric cells as well as the cell-fitting and cell-blending process.

Like T-snakes, superquadrics have both an implicit and parametric representation. Consequently, we can incorporate both region and boundary information into the superquadric cell-fitting process. Furthermore, we have developed a cell *growing* and *splitting* process. In the first phase of the process, a seed cell is initially placed within a region of the target object. Initial cell placement is currently manual but automatic mechanisms could be easily created for many objects (using, for example, bounding boxes). The cell then globally deforms and adapts to the coarse shape of that region (Figure 5.28a,b). In the second phase, the parametrically-defined "skin" of the fitted cell is locally deformed for a preset number of iterations (Figure 5.28c). We then examine the displacement of the skin from the underlying cell to find sections of the skin that have achieved a local maximum displacement. The average position of a local maximum section is used as a seed site for a new "child" superquadric cell. Seed sites could also be preset near "parent" cell boundaries for objects with relatively simple shapes. Points on the parent cell closest this seed site are used as a constraint point; a "child" cell is constrained to stay within a small distance of this point (Figure 5.29a-d). In addition, to determine a good initial orientation of non-circular cells, child cells are "born" as circles and their growth or deformation is restricted to global scaling and translation for a fixed time period. During this preliminary growth phase, the child cell attempts to center itself among local image edge features. The orientation of the cell is then computed from its center position and the constraint point on the parent cell (i.e. along an axis formed from these points).

(a) (b) (c)

Figure 5.28: (a),(b) Superquadric cell growing and fitting to section of object. (c) Parametric "skin" contour of cell deforming away from cell. Maximum displacement regions are used as seed sites for "child" cells.

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

Figure 5.29: Superquadric cells growing and "splitting" to capture coarse shape of object.

Once the cell orientation has been computed, the cell squareness parameter is set to a predetermined value that closely matches cell shape to object shape and the normal growth of the cell is initiated.

A priori knowledge of a specific shape recovery problem can be incorporated into the cell growing/splitting process either statically or dynamically. For example, limits on the global shape parameters of an individual cell can be preset: upper and lower limits for the scale, bend, taper, and eccentricity of a cell can ensure that infeasible cells are not generated. The minimum and maximum number of cells can also be statically determined. Conversely, "child" cells can dynamically "inherit" shape characteristics from their parent during the cell splitting process. Child cells can also be "informed" of current region and edge information by a parent. For example, it is often the case in angiograms of arterial structures that the intensity of pixels diminishes as the distance from the root of the artery "tree" increases. This property makes it difficult to find a single intensity threshold value. By performing local region and edge analysis, a parent cell can detect changes to the image intensity and pass this information to child cells. The result is a robust adaptive threshold setting technique.

The cell growing/splitting process terminates when no further seed sites can be found

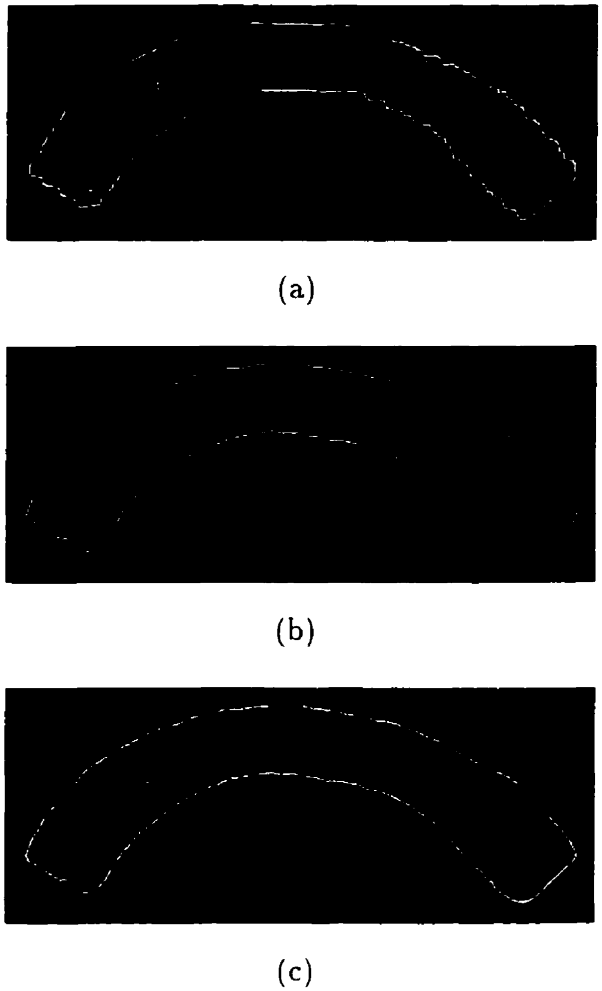


Figure 5.30: (a) Superquadric cells blended to form initial T-snake. (b) Final fitted T-snake (c) Fitted T-snake and underlying cells.

or the preset maximum number of cells has been reached. Using the implicit form of the superquadric cells and the simplicial grid, the implicit cell functions are blended resulting in an isocontour closely surrounding the cells (Figure 5.30a). This implicit skin is converted to a T-snake. We then allow the T-snake to deform as usual to capture the shape details of the object (Figure 5.30b). Using anchored springs, we can also constrain the distance between the T-snake nodes and the initial implicit "skin". We can then use the displacement of the final T-snake from the underlying superquadrics as a mechanism for the automatic detection of shape abnormalities (Figure 5.30c).

Superquadric cells possess several desirable properties for noisy data fitting:

- A small number of intuitive global shape parameters that can generate and concisely describe a wide range of shapes.
- An implicit and parametric representation. The parametric form provides a well-defined boundary representation that can be used to perform efficient cell fitting and inter-cell constraints. The implicit form provides a well-defined cell interior and the ability to add or blend cells together.
- Image region and edge information can be integrated over the cell boundary and/or cell area. The effect of missing or noisy image features is therefore greatly minimized.

These features, combined with the locally deformable T-snakes, result in a very robust shape recovery technique that is "tailorable" to a specific image analysis task. In Chapter 7 we present several examples using the superquadric cell/T-snake system to demonstrate its potential as a widely applicable and fully automatic medical image analysis technique.

Chapter 6

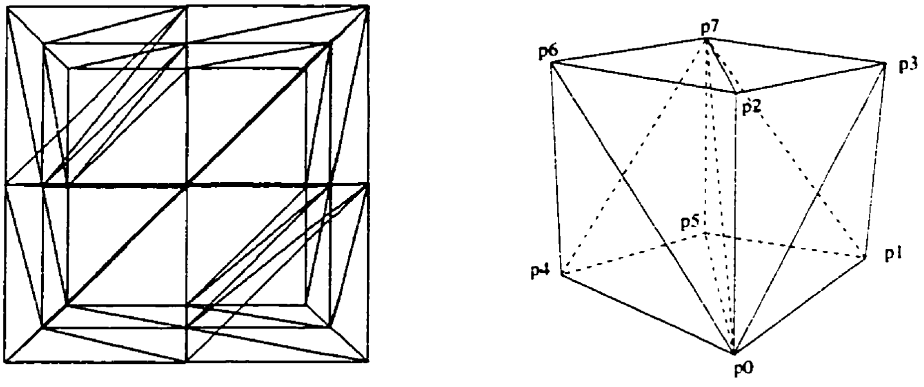
Topologically Adaptable Surfaces

This chapter presents topologically adaptable surfaces (T-surfaces) — a deformable closed-surface model that is a 3D extension of T-snakes. We begin this chapter with a description of the physics-based model formulation, highlighting the differences of the 2D and 3D models. We then extend the simplicial cell decomposition framework to three dimensions using tetrahedral cells and demonstrate the fundamental similarity of the 2D and 3D algorithms. We conclude the chapter with illustrative examples of T-surface features and constraints.

6.1 Model Description

We define our discrete deformable surface model as a closed oriented triangular surface mesh. A T-surface is a three dimensional generalization of a T-snake. Furthermore, analogous to the T-snake, a T-surface is a discrete form of a classical deformable model (i.e. a closed form of the deformable surface described in section 2.1.3). The vertices of the T-surface triangles, the model nodes, act as a dynamic particle system where the particles are interconnected by discrete spring units [109]. Like the T-snake, we associate with these nodes time varying positions $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]$, along with "tension" forces $\boldsymbol{\alpha}_i(t)$, "rigidity" forces $\boldsymbol{\beta}_i(t)$, "inflation forces" $\boldsymbol{\rho}_i(t)$, and external forces $\mathbf{f}_i(t)$.

The behavior of the T-surface is governed by a 3D form of equation (4.1). Similarly, 3D forms of the tension forces (4.2), inflation forces (4.4), and external image forces (4.6) are used. Like the T-snake, the rigidity forces attempt to minimize the local curvature



(a) (b)

Figure 6.1: (a) 3D Freudenthal triangulation. (b) Cubic cell division into 6 tetrahedra: $\rho_0 = (p_0, p_1, p_3, p_7)$, $\rho_1 = (p_0, p_1, p_5, p_7)$, $\rho_2 = (p_0, p_2, p_3, p_7)$, $\rho_3 = (p_0, p_2, p_6, p_7)$, $\rho_4 = (p_0, p_4, p_5, p_7)$, $\rho_5 = (p_0, p_4, p_6, p_7)$.

of the model by keeping a model node at the centroid of its neighbors [69]

$$\beta_i(t) = b_i \left(\mathbf{x}_i(t) - \frac{1}{n} \sum_{j \in \mathcal{N}} \mathbf{x}_j \right), \tag{6.1}$$

where n is the number of neighbors. This force can be made scale invariant by dividing the right hand side of (6.1) with the maximum distance between neighboring nodes.

6.1.1 3D Simplicial Cell Decomposition

Simplicial cell decomposition in 3D uses tetrahedra. The 3D Coxeter-Freudenthal triangulation (i.e. tetrahedrization) (Figure 6.1a) is constructed by dividing Euclidean space using a uniform cubic grid and subdividing each cube into 6 tetrahedra (Figure 6.1b).

The set of grid tetrahedra that intersect the boundary surface of the anatomic structure (the boundary tetrahedra) form a three dimensional combinatorial manifold that has as its dual a two dimensional manifold that approximates the object boundary surface. The two dimensional manifold is constructed from the intersection of the object boundary surface with the edges of each boundary tetrahedra. The intersection points result in one triangle or one quadrilateral (which can be subdivided into two triangles) approximating the surface inside each boundary tetrahedra (Fig. 6.2), where each triangle or quadrilateral intersects a tetrahedron on three or four distinct edges, respectively.

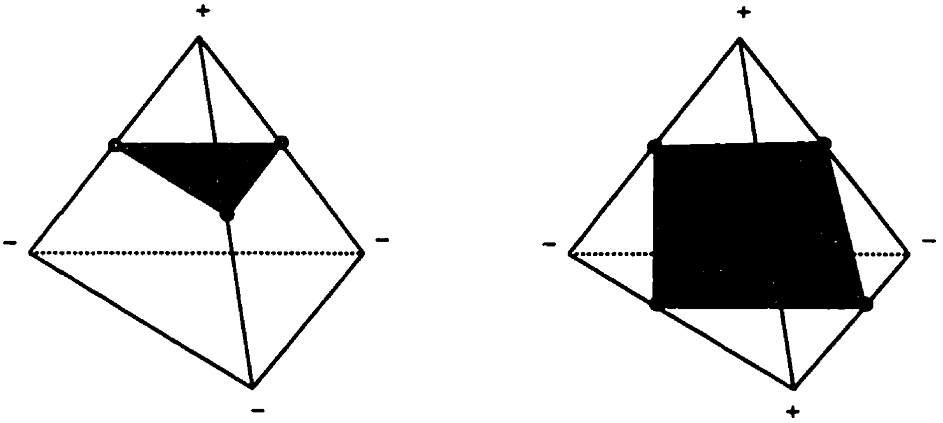


Figure 6.2: Intersection of object boundary with grid tetrahedra. Either one triangle is generated or one quadrilateral which can be divided into two triangles. The "signs" of the tetrahedra vertices indicate whether the vertex is inside or outside the object.

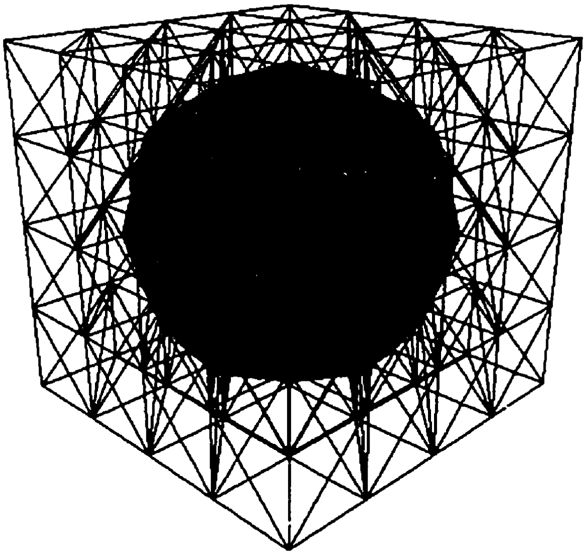
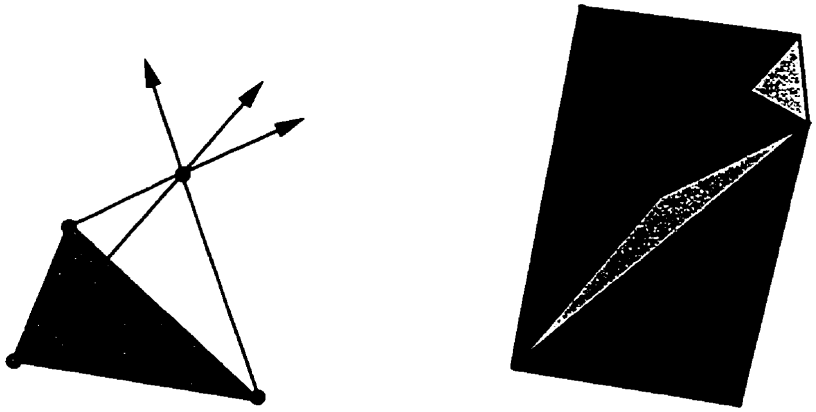


Figure 6.3: Approximation of a sphere using a 4 × 4 × 4 grid.



(a) (b)

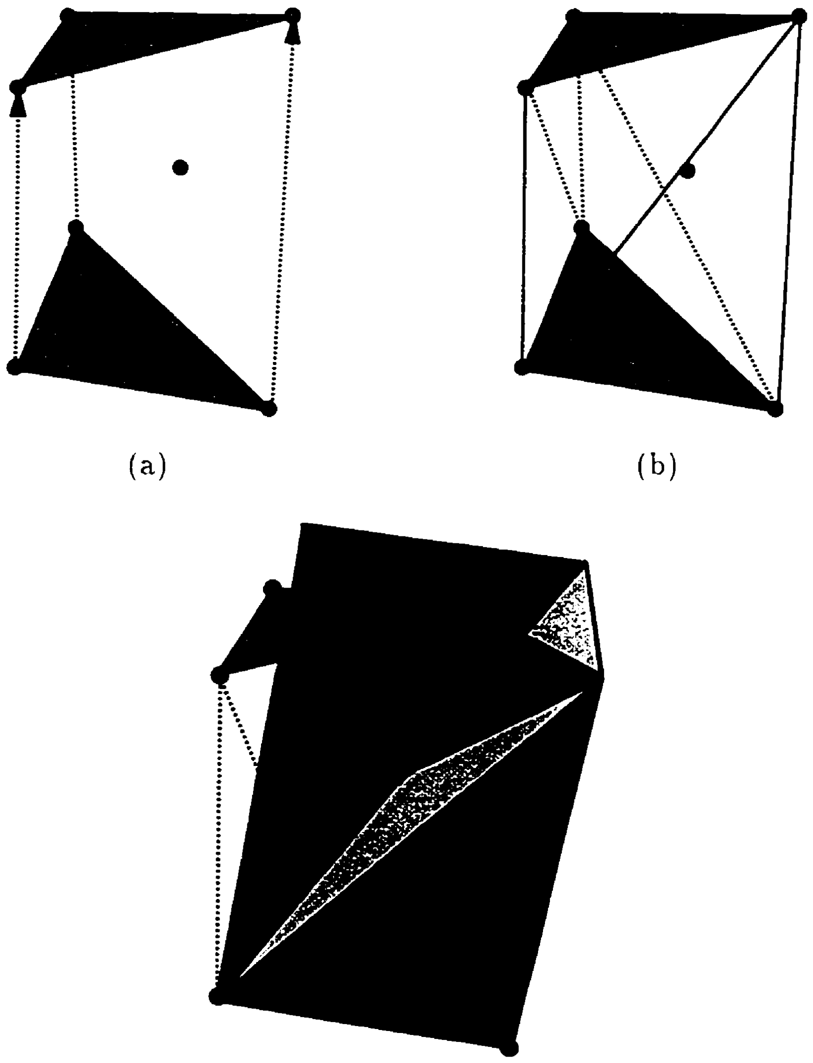
Figure 6.4: Formation of subspaces used in the classification algorithm. (a) Three planes are formed by joining the two nodes of a model triangle edge with a grid vertex. (b) The three planes partition space into 8 subspaces.

The triangle (or quadrilateral) separates the positive vertices of the tetrahedron from the negative vertices. The set of all these triangles constitute the combinatorial manifold that approximates the object boundary surface (Figure 6.3).

6.1.2 Iterative Reparameterization

The T-surface reparameterization process is identical to the T-snakes case. In phase 1. a bounding box is formed around each model element (triangle) at its new position and grid intersection points are computed. The intersection points are used to create the new model nodes.

In phase 2. each model triangle may have passed over zero. one. or several grid vertices during a deformation step (Figure 6.5a). The model triangle at its current and new positions form a prism-like polyhedron. The sides of this prism may be nonplanar so they are divided into two triangles forming an 8-sided polyhedron (Figure 6.5b) (there are two possible triangulations of the prism sides). To determine if a grid vertex is burned by a model triangle during a deformation step, we must determine if the grid vertex is contained by (inside) the polyhedron. The polyhedron may be twisted and/or inverted depending on the path followed by the model element nodes during the deformation step. Rather than deal with the potentially complex-shaped polyhedron directly, we use



(a)

(b)

(c)

Figure 6.5: (a) A model triangle moves to a new position during a deformation step and potentially burns a grid vertex. (b) The 8-sided polyhedron formed from the model triangle at current and new positions. (c) The 8 subspaces used to classify a model triangle at its new position and determine if the grid vertex was burned.

a simple, robust classification algorithm that depends on the fixed positions of the grid vertex and the well-defined current position of the model triangle nodes $p1, p2, p3$. The classification algorithm is a 3D extension of the T-snakes algorithm described in Section 4.3.2. That is, we partition the image domain into eight subspaces by forming three planes. The three planes are formed by joining the two nodes of each model triangle edge at their current position to the target grid vertex (Figure 6.4a,b). We then classify the three nodes of the model triangle at its new position $p1n, p2n, p3n$ into one of the eight subspaces (Figure 6.5c). The classification algorithm is then used to quickly determine whether the grid vertex in question has been burned. The following section outlines the basic principles used to construct the algorithm.

3D Classification Algorithm

As mentioned above, we begin by classifying $p1n, p2n, p3n$ into one of the 8 subspaces. This classification operation consists of 3 inside-outside half-space tests for each model element node. A 3D half-space test essentially consists of a dot product between a point p and the *point-normal* equation of a plane and is extremely efficient. There are 3 model nodes and 8 subspaces for a total of 512 possible classification combinations.

Examination of the 2D classification algorithm reveals essentially two underlying principles. For cases 1-12 (Figure 4.8), if $p1n$ lies to the left of $L1$ (defined as the inside half-space), and $p2n$ lies to the right (in the inside half-space) of $L2$, then v is labeled as burned. For the ambiguous cases (13, 14), this principle also holds; although in case 14, it is reversed. A second principle can then be derived which states that v and either of $p1$ or $p2$ must lie in the same half-space of the half-spaces defined by $\overline{p1np2n}$. For case 15 (or 16) a similar principle states that v must lie in the same half-space as $p1$ ($p2$) where the half-spaces are again defined by $\overline{p1np2n}$.

These two simple principles carry over to the 3D classification algorithm. We first note that in the 2D case each point, $p1n, p2n$, has its own "constraint" line $L1, L2$ respectively, against which it is classified. In 3D, each point $p1n, p2n, p3n$ is associated with *two* constraint planes. For example, $p1n$ is associated with the two planes defined by points $p1, p2, v$ and $p1, p3, v$. The first principle, extended to 3D, states the following:

- Point pin, $i = 1,2,3$, is in the inside half-space of one of its two constraint planes (i.e. there is at least one point $(p1n, p2n, p3n)$ in the inside space of each constraint plane).
- There is at least one point pin, $i = 1,2,3$, in the inside space of all constraint planes.

If these two conditions are met, then v is labeled as burned.

For the ambiguous cases in 3D analogous to cases 13, 14 in 2D (i.e. no point $(p1n, p2n, p3n)$ is inside all 3 constraint planes or outside all 3 planes), a second principle is used:

- Vertex v lies in the same half-space as $p1$ or $p2$ or $p3$, of the half-spaces defined by the plane formed from model triangle $p1n, p2n, p3n$.

If this condition and the two conditions from the first principle are met, v is labeled as burned. For the 3D cases equivalent to the 2D cases 15, 16, the second principle is modified slightly:

- Vertex v lies in the same half-space as all points $p1, p2, p3$ that have a corresponding point $p1n, p2n, p3n$ in the inside space of all 3 constraint planes (where the half-spaces are again defined by the plane formed from the model triangle $p1n, p2n, p3n$.

6.1.3 Topological Transformations

Topological transformations are handled as in the T-snakes case. That is, by tracking the interior grid vertices (and hence the boundary grid tetrahedra) and reestablishing the correspondence of the T-surface with the grid after each deformation step, the boundary or "isosurface" of the new model(s) is uniquely determined.

6.2 T-Surface Features

Chapter 5 described various features of T-snakes. These features are inherent to both 2D and 3D ACD-based deformable models. The following illustrative examples will reiterate several of these features with T-surfaces.

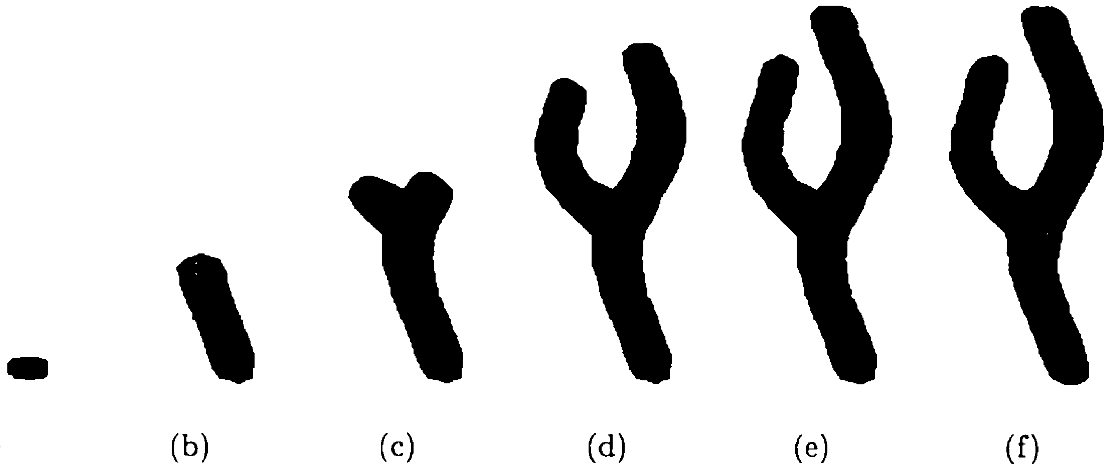


Figure 6.6: T-surface flowing into an object with a bifurcation. The dark shaded regions are "frozen" and have been removed from the computation.

6.2.1 Geometric Coverage and Topological Adaptability

In the first example, we use a synthetic data set of a branching structure to demonstrate the automatic model element subdivision provided by the reparameterization process (Figure 6.6(1)-(6)). This feature provides a T-surface with broad geometric coverage. This example also illustrates the model element cooling process described in section 5.1.6. The darker shaded regions of the T-surface indicate frozen model elements. Only a relatively small number of model elements are active throughout the T-surface evolution in this example. Consequently, the segmentation and reconstruction requires only approximately 1 minute on an SGI Indigo 2 workstation (grid resolution: $32 \times 32 \times 32$ cubic cells, each divided into 6 tetrahedra, data set: $128 \times 128 \times 128$ voxels). Figure 6.6f shows the T-surface converted into a classical deformable surface model. Like the T-snake, conversion to a classical deformable model is simply a matter of discarding the grid.

In the second set of examples we demonstrate the topological adaptability of T-surfaces. Figures 6.7, 6.8, and 6.9 show a T-surface growing (or shrinking) and merging with itself (or splitting) to segment several synthetic data sets. In Figures 6.8e and 6.9f, the T-surface has been converted to a classical deformable surface model.

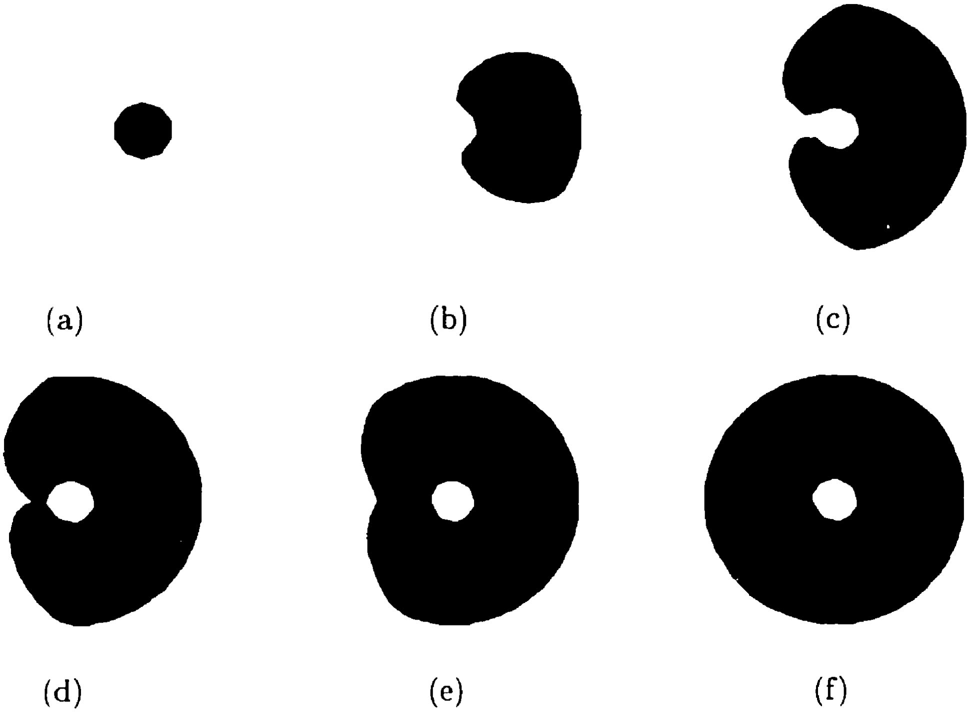


Figure 6.7: T-surface deforming and merging with itself to recover toroidal object.

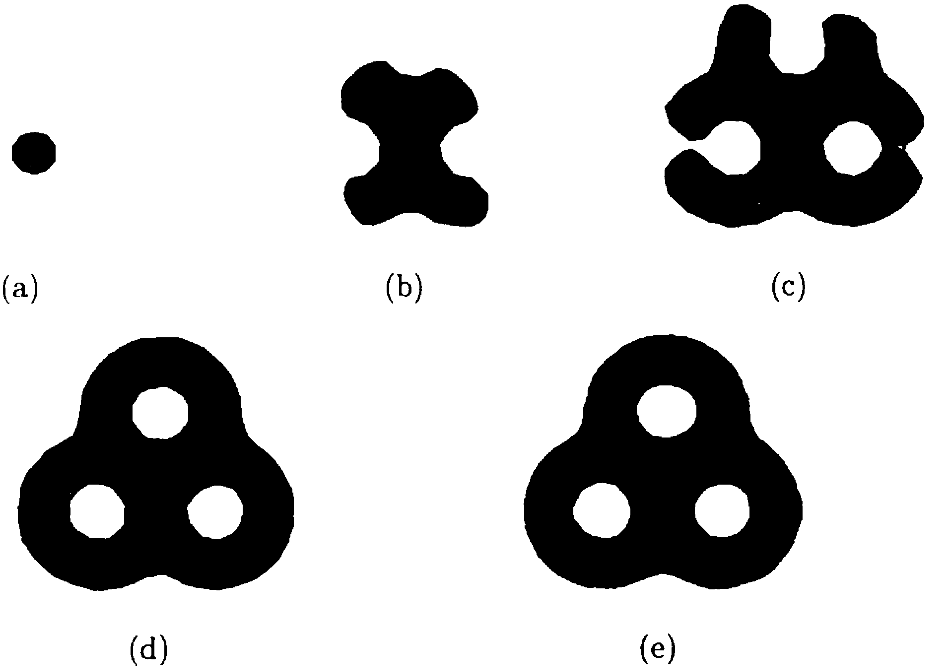


Figure 6.8: T-surface recovering shape of object with complex topology.

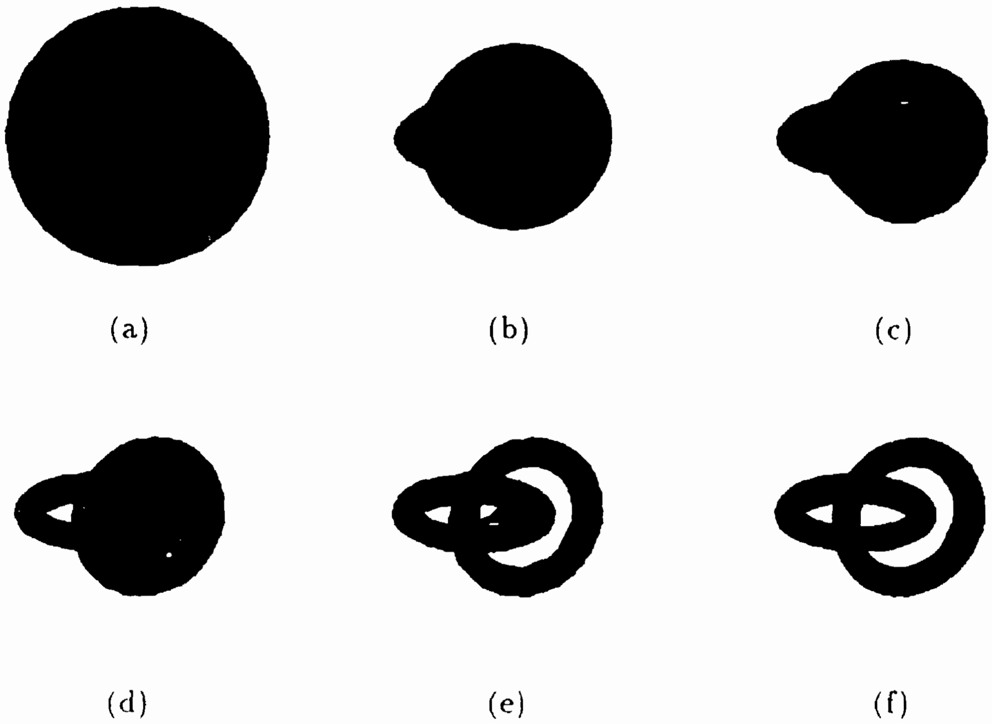
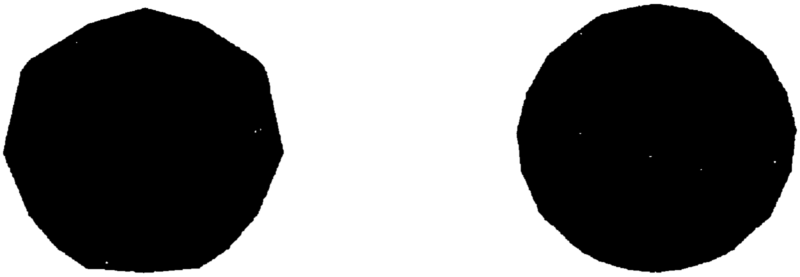
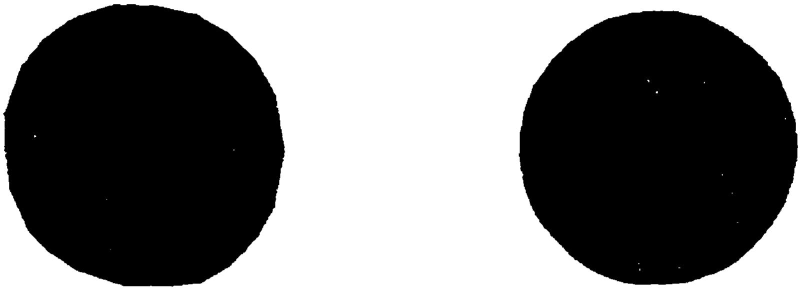


Figure 6.9: T-surface shrinking, wrapping, and splitting to recover interlocked ring objects.



(a) (b)

Figure 6.10: Approximation of a sphere using (a) 4 × 4 × 4 grid and (b) 8 × 8 × 8 grid.



(a) (b)

Figure 6.11: Conversion of T-surface sphere approximation to classical deformable surface model using (a) 4 × 4 × 4 grid and (b) 8 × 8 × 8 grid. Note the redistribution of model nodes from the action of the internal forces.

6.2.2 Multiresolution Capability

Figure 6.10 shows a sphere approximated with a T-surface at two grid resolutions. Like the T-snake, if the grid resolution is restricted to multiples of an initial coarse resolution, the finer grids preserve the T-surface model nodes of the coarse resolution grids.

As we have mentioned, converting a T-surface to a classical deformable surface model is simply a matter of discarding the grid. This action results in a redistribution of the surface model nodes due to the internal forces (Figure 6.11).

6.3 T-Surface Constraints

We have incorporated several of the constraint mechanisms described in Section 5.2 into the T-surface model. While it is relatively straightforward to include hard geometric point constraints into the 3D ACD framework (the procedure is essentially identical to that of T-snakes). this type of constraint has not been implemented. Statistical constraints are also not described in this section (the principle is identical to the 2D case). They are. however. used in several of the application examples presented in Chapter 7.

6.3.1 Topological Constraints

As described in Section 5.2.3. the ACD framework provides a robust. efficient mechanism to identify and prevent model self-intersections. allowing us to impose a global topological constraint on T-snakes and T-surfaces. In the 3D case. the T-surface has an initial topology of a sphere. A series of topology-preserving deformations is then used to deform the model.

The principle of the 3D topology-preserving deformation is similar to the 2D case. To reiterate. a model element l is allowed to burn a grid vertex v during a deformation step only if the deformation of the element is deemed valid. The validation check is performed by extracting any and all model elements from the 24 grid tetrahedra surrounding v. including model element l. Like the 2D case. we then perform a breadth first traversal of these extracted elements to determine if they form a contiguous section of the model. In this case. any element of the extracted set can be reached from any other element through neighboring elements. where all neighbor elements are also members of the extracted set (two model elements are neighbors if they share at least one model node).

If the extracted model elements do form a contiguous section of the T-surface. then the deformation of l during this deformation step is deemed valid since the extracted elements cannot form a closed circuit around v. If the extracted elements do not form a contiguous set then the deformation of l is deemed invalid. Like T-snakes. since we do not actually update model element positions until after phase 2 of the reparameterization process (Section 6.1.2). the validation check is time-step dependent. For this reason a grid vertex cannot be burned during the current deformation step if any of the 26 grid vertices connected to it have been burned during this deformation step.

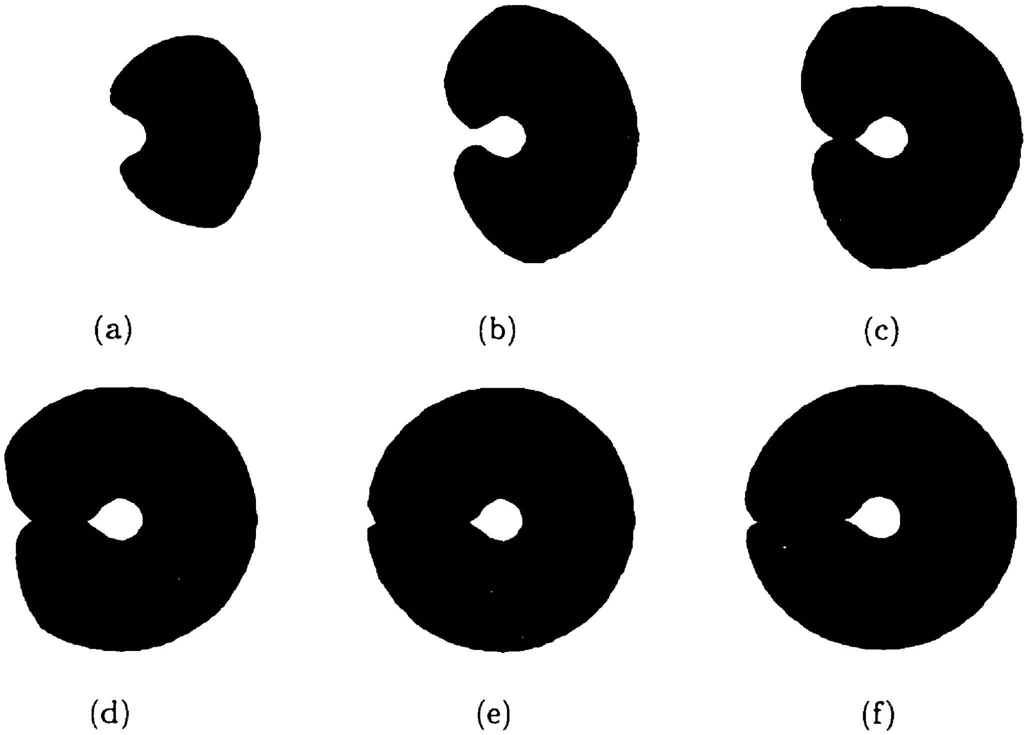


Figure 6.12: T-surface flowing around torus. The topology constraint prevents the model from merging with itself. It collides but cannot not penetrate.

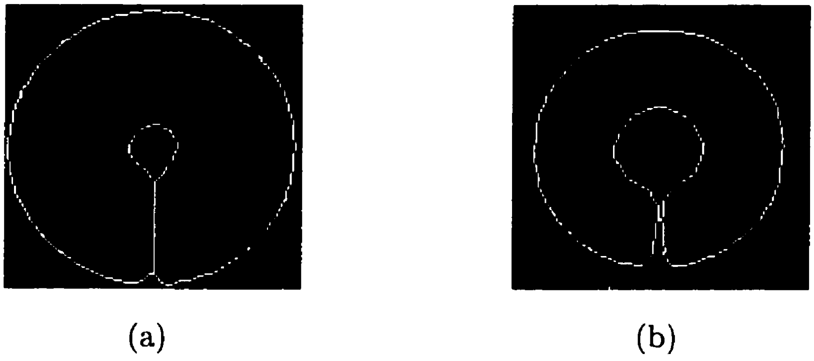


Figure 6.13: Cross-sectional slices of the fitted T-surface.

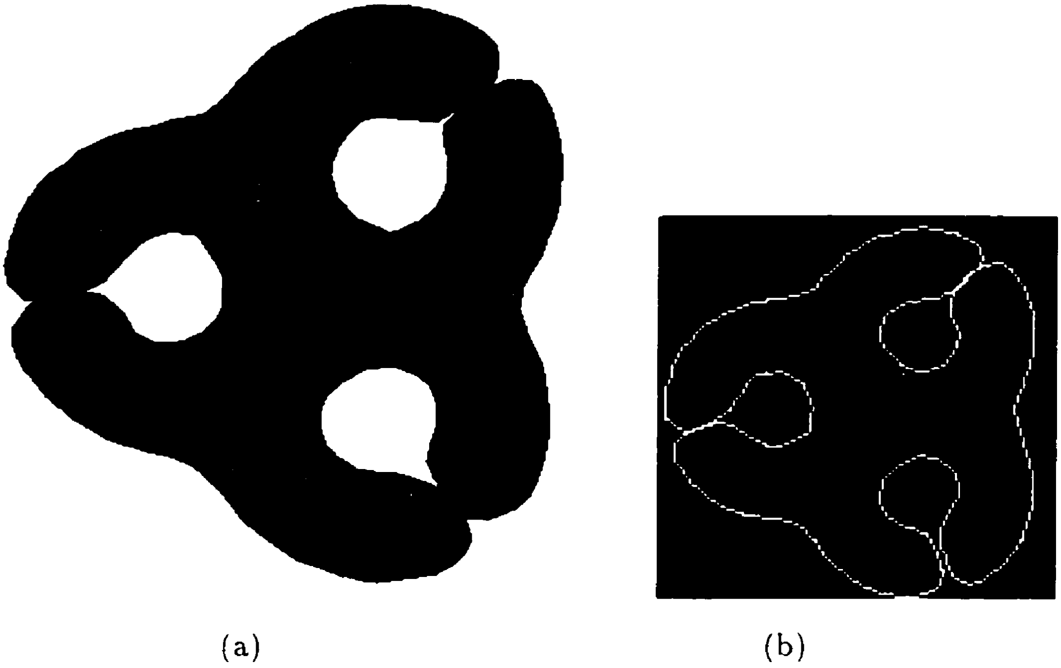


Figure 6.14: T-surface fit to object with complex topology. The topology constraint prevents the T-surface from merging with itself. (b) Cross-section of the fitted model illustrating that no topology change has taken place.

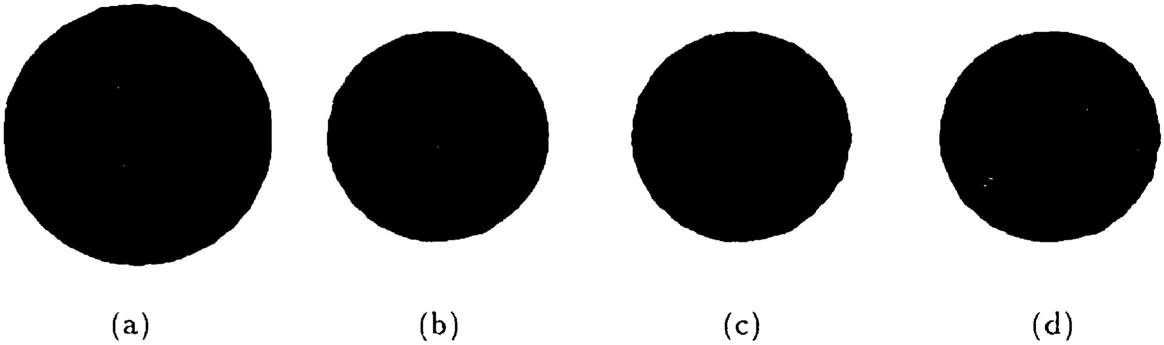


Figure 6.15: T-surface shrinking and wrapping around torus. The topology constraints prevents the T-surface from splitting.

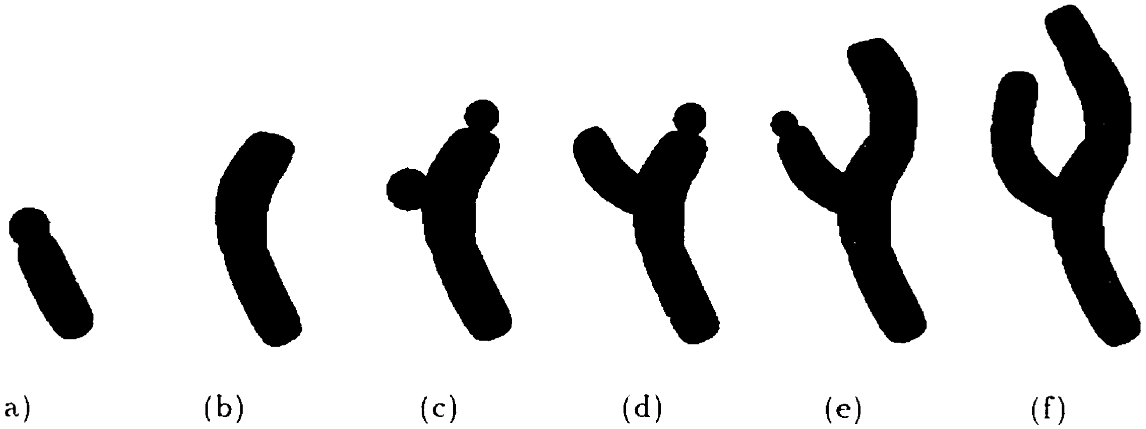


Figure 6.16: Superquadric cells growing and splitting to recover object with bifurcation.

6.3.2 Global Shape Constraints

We have extended the superquadric cell growing/splitting procedure to 3D using superellipsoidal cells (Figure 6.16a-f). As in the 2D case, the superellipsoids are able to scale, bend and taper, and are constrained to overlap. Once a cell has completed its growing phase, a parametric "skin" surface deforms away from the cell. Maximum skin displacement regions are then determined and child cell seed-sites computed. Child cells are given an initial spherical shape for a fixed number of deformations to establish a good orientation. They are then converted to cylinders (in this example) for the remainder of the growing phase. Upon termination of the cell growing/splitting process, the implicit form of the deformed superquadric ellipsoids are blended resulting in an isosurface closely surrounding the cells (Figure 6.17a). This implicit skin is converted to a T-surface which deforms to capture the shape details of the object (Figure 6.17b).

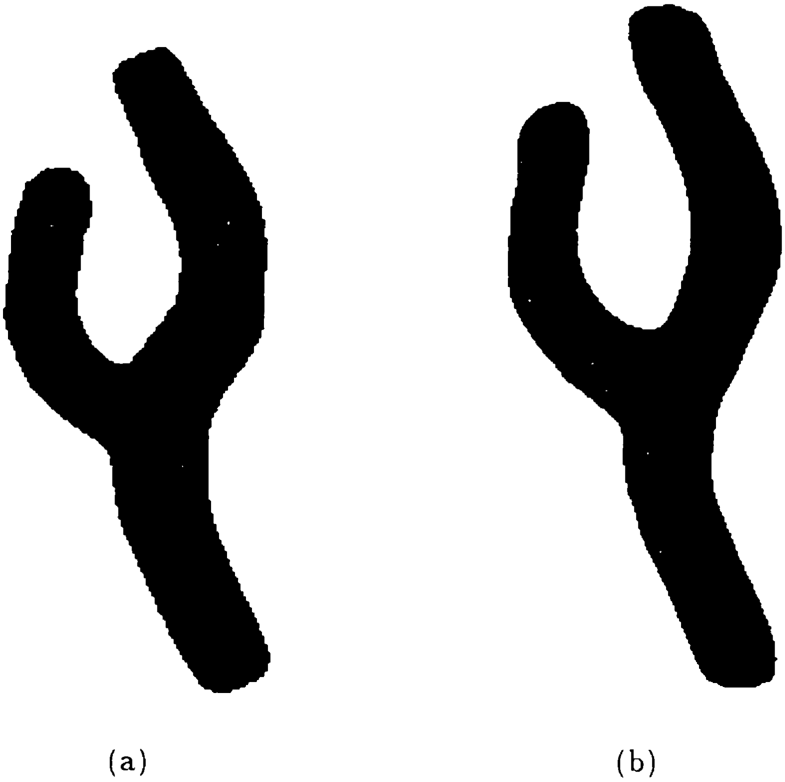

(a) (b)

Figure 6.17: (a) Result of blend of superquadric cells to form initial T-surface. (b) Final fit of T-surface.

Chapter 7

Segmentation of Medical Images with T-Snakes and T-Surfaces

This chapter presents 2D and 3D segmentation examples using T-snakes and T-surfaces. A wide range of anatomic structures with complex shapes and topologies are extracted from a variety of medical images. Examples range from segmenting neuronal cells from an EM photomicrograph to segmenting cerebral vasculature from massive Magnetic Resonance Angiogram (MRA) image volumes. The examples serve to demonstrate the power and flexibility of simplicial cell decomposition based deformable models for segmentation, reconstruction and analysis. This chapter also describes our prototype ACD-based interactive segmentation system.

7.1 Prototype Medical Image Processing System

We have developed prototype interactive segmentation systems based on 2D and 3D versions of our ACD-based deformable models. The systems are written in C and are implemented on Silicon Graphics workstations. User interaction is made available through a graphical user interface and an input device such as a mouse. The remainder of this section will briefly describe the 3D system.

The 3D system provides views of the data and model in separate windows to facilitate the initialization, interaction and visualization of the data and the T-surface. One window displays a 3D view of the model which can be interactively rotated or scaled. Another window displays a 2D image slice with the corresponding cross sectional contour of the model overlaid. Three standard view planes are available to display the image slices (XY, XZ, YZ). The user can quickly scan through the image volume slices using the keyboard arrow keys.

Two control panels are used for interaction and display. The first control panel is used in combination with the mouse to initialize the deformable model. Initialization is performed using an implicitly defined superquadric function. The superquadric "seed" can be quickly scaled, bent and tapered, if desired, using the control panel, placed inside the object of interest (or scaled such that it contains the object) and is then converted to a T-surface representation. Multiple "seeds" can be dynamically created and placed throughout the object. Seed placement is performed with the mouse and the 2D image slice window. A cross-sectional view of the seed is displayed in the image slice window and the user can employ the mouse to move the seed into position. This process is repeated for the two other orthogonal views of the image volume.

The second control panel controls model evolution and display. The model can be stopped, started, and restarted, and model evolution and display parameters can be dynamically changed. Furthermore, the user can also interact with the model in the 2D image slice window by positioning the mouse at some point in the window and depressing a mouse button. The mouse position is determined and the closest model point on the cross-sectional contour is calculated. A spring force is then applied to the model in the direction of the vector from the model point to the mouse position. The force is applied as long as the mouse button is depressed and its direction can be changed by dragging the mouse button to a new position in the window. The user can also interact with the model by positioning the mouse and depressing a mouse button to specify a "pin" point. Pin points apply a sustained spring force to the closest model node on the cross-sectional contour. These sustained forces "pin" the model to the point (i.e. force the model to closely approximate the pin point). This mechanism allows the user to reinforce or create an object edge.

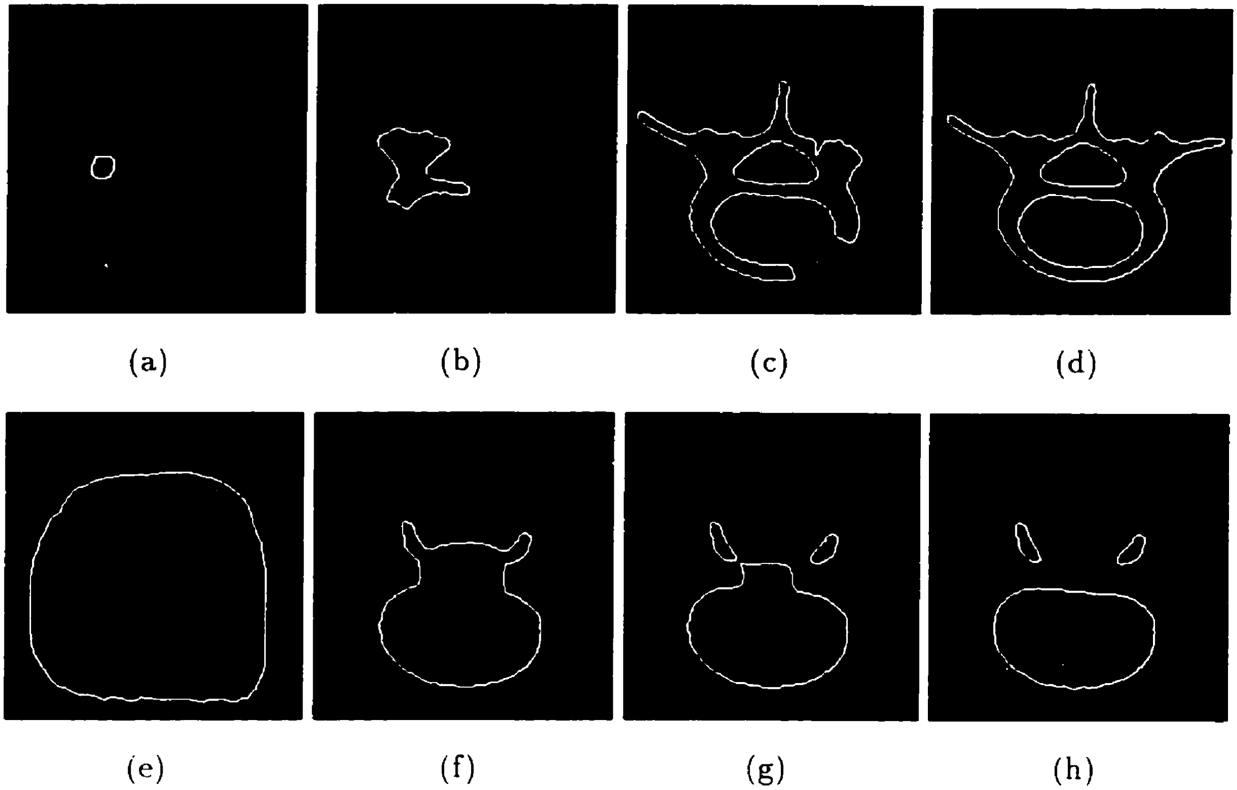


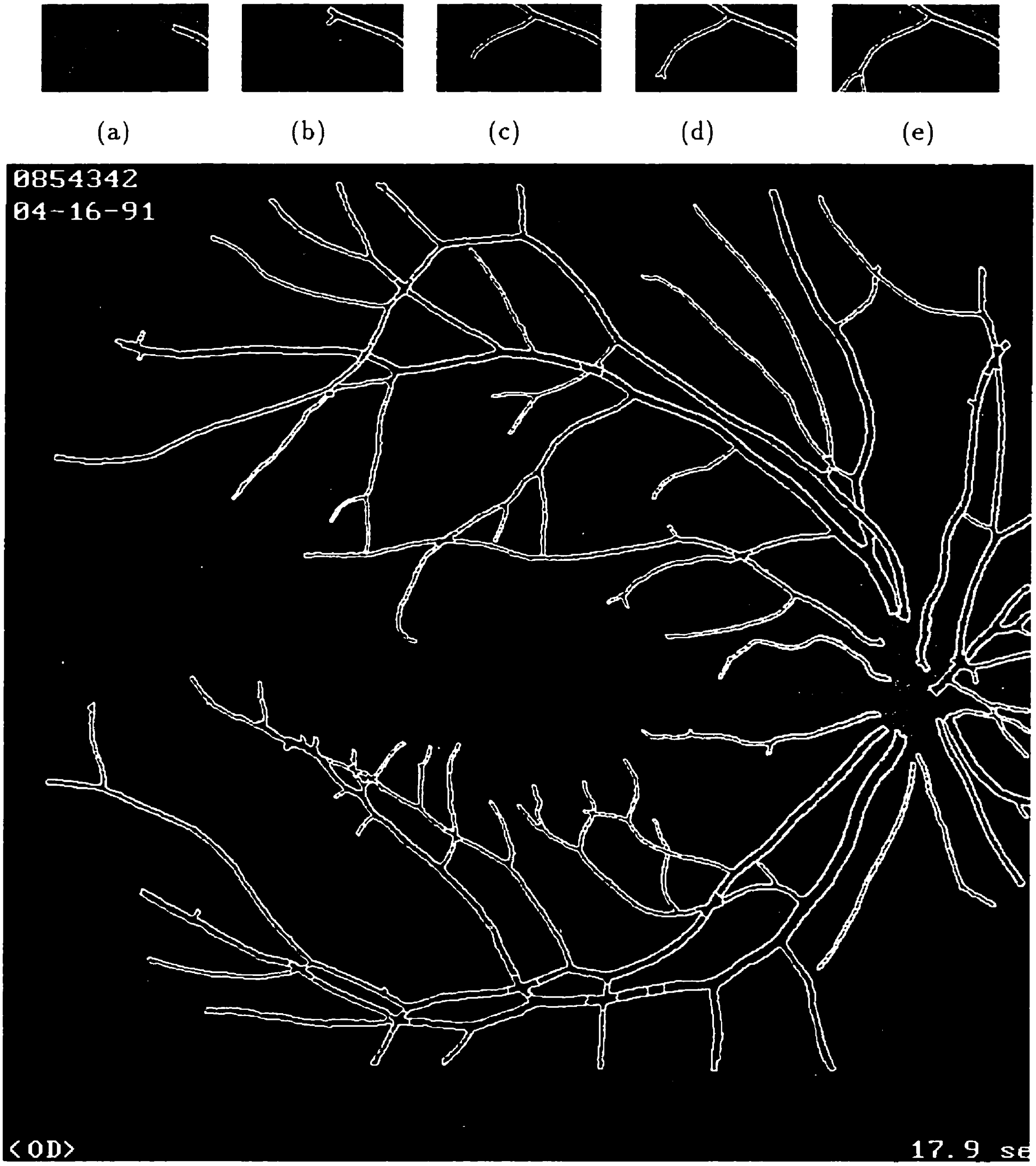
Figure 7.1: Segmentation of two cross sectional images of a human vertebra phantom.

7.2 Image Segmentation with T-Snakes

7.2.1 Segmentation of Vertebra Phantom Cross-sections

Our first experiment demonstrates the geometric and topological flexibility of T-snakes on a more realistic image. We segment several 128×120 image slices from a CT image volume of a human vertebra phantom. We use a 50×50 squared cell grid (with each cell divided into two triangles) and model parameters: $p = 51.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.005$, and $N = 10$ (time steps per deformation step). Once the T-snake collides with itself in Figures 7.1a–d, it automatically splits into three parts, two parts segmenting the inner boundary of the vertebra and one for the outer boundary. In Figures 7.1e–h, the snake shrink-wraps itself around the objects and splits to segment the different parts.

7.2.2 Segmentation of Vasculature from Retinal Images



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

Figure 7.2: Segmentation of the blood vessels in angiogram of retina. The top row is an image sequence showing a snake flowing and branching along a vessel.

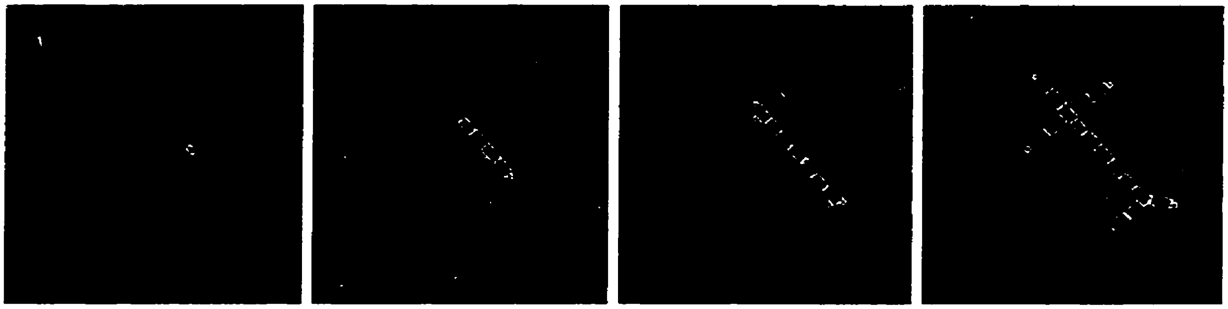
Retinal angiograms provide information about the blood supply system of the retina. They are used in diagnosing diabetic retinopathy, hypertension and various vascular disorders. Different imaging techniques, such as color or fluorescin photography, and scanning laser opthalmoscopy are now available. Fluorescent angiography is the most common technique, as it provides high resolution images. Routine analysis of retinal images involves inspection of the vessel tree (for abnormally shaped vessels, their branching pattern, vessel caliber), presence and location of microaneurysms, haemorrhages, exudates, new vessels, degree of retinal perfusion and the shape and size of the optic disc.

Analyzing the vessel morphology first requires their extraction and reconstruction in some suitable analytic form. Skeletonization algorithms can then be applied to compute vessel topology and geometry. Segmentation of the vessels is an especially difficult problem. Fluorescent angiograms of the retina show a marked decrease of image intensity from the center of the image to the periphery due to the two dimensional projection of the retinal surface and inhomogeneity of the illumination. For this reason, we currently require some user interaction to perform the segmentation using T-snakes.

We initialized a T-snake at the source of each major branch of a 1024 × 1024 pixel angiogram (Figure 7.2) in the vicinity of the optic disk and performed the segmentation one branch at a time. To enable the T-snake to flow along the narrow vessels, a pixel-resolution grid was required with model parameters: $p = 51.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.0004$, and $N = 10$. Due to the image intensity variation we used several barriers or forbidden zones (Section 5.2.1) to prevent a T-snake from leaking into non-vessel areas. To decrease or eliminate this user interaction and to force a T-snake into the very narrow vessels may require image preprocessing to adjust or normalize the average intensity and contrast. This topic is not explored in this thesis.

Segmentation using Global Shape Constraints

In this experiment we demonstrate the potential of the cell growing and splitting process for segmenting and analyzing branching structures. We seed a vessel with a single cell in a clipped portion of the retinal image and start the cell growing-splitting process (Figure 7.3a-d). When a cell splits, it passes current information about the local image



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

Figure 7.3: Superquadric cells growing and splitting to recover vessels in clipped retinal image.

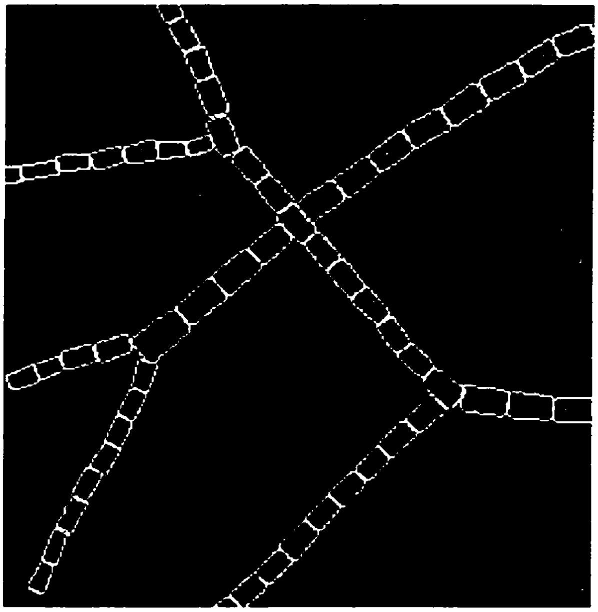


Figure 7.4: The result of the cell growing-splitting process.

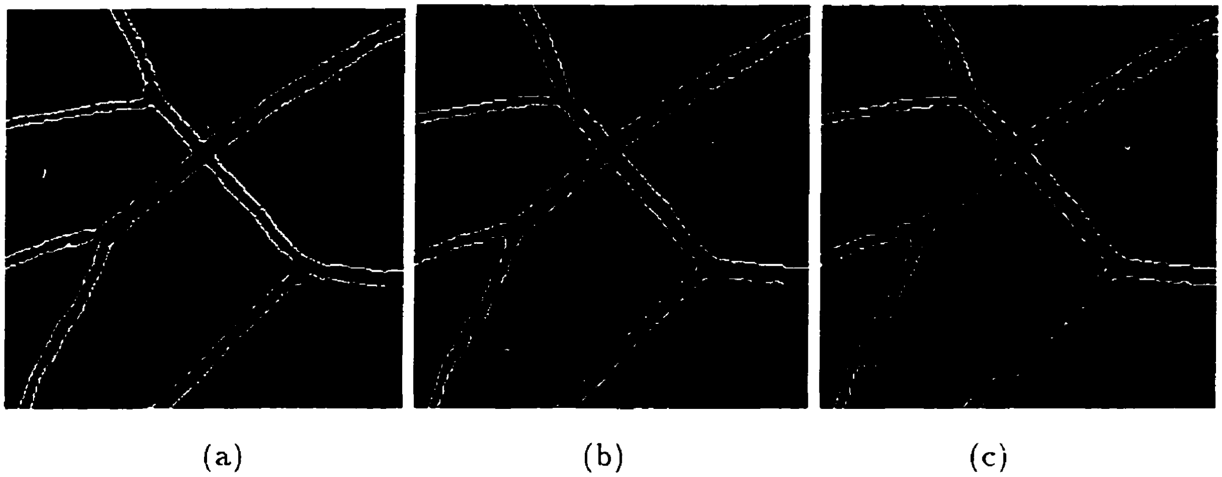

(a) (b) (c)

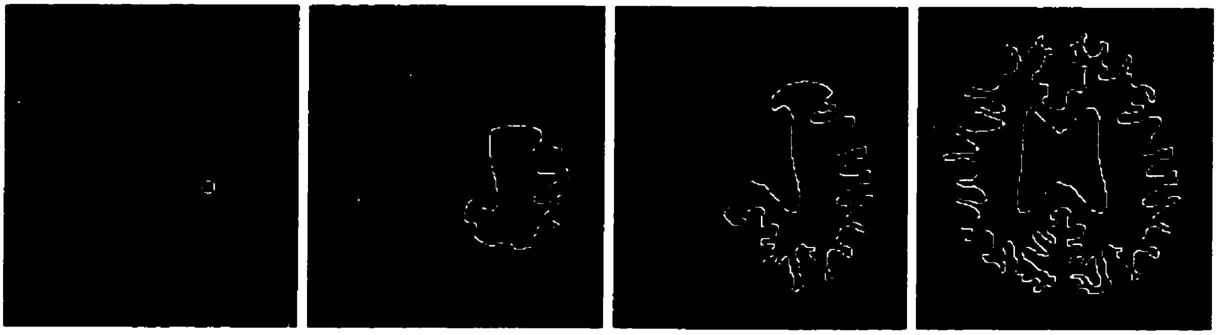
Figure 7.5: (a) Superquadric cells blended to form initial T-snake. (b) Final T-snake segmentation. (c) T-snake with underlying cells.

intensity to its child. The child cell is able to alter the image intensity threshold based on this information, effectively compensating for the intensity variation mentioned earlier. The final result of the cell fitting is shown in Figure 7.4. The cells are blended to form the initial T-snake (Figure 7.5a), which then deforms to accurately capture the vessel boundaries (Figure 7.5b). The combination of the T-snake and underlying cells (Figure 7.5c) provides sufficient information to perform a complete analysis of vessel morphology.

7.2.3 Segmentation of Brain Contours

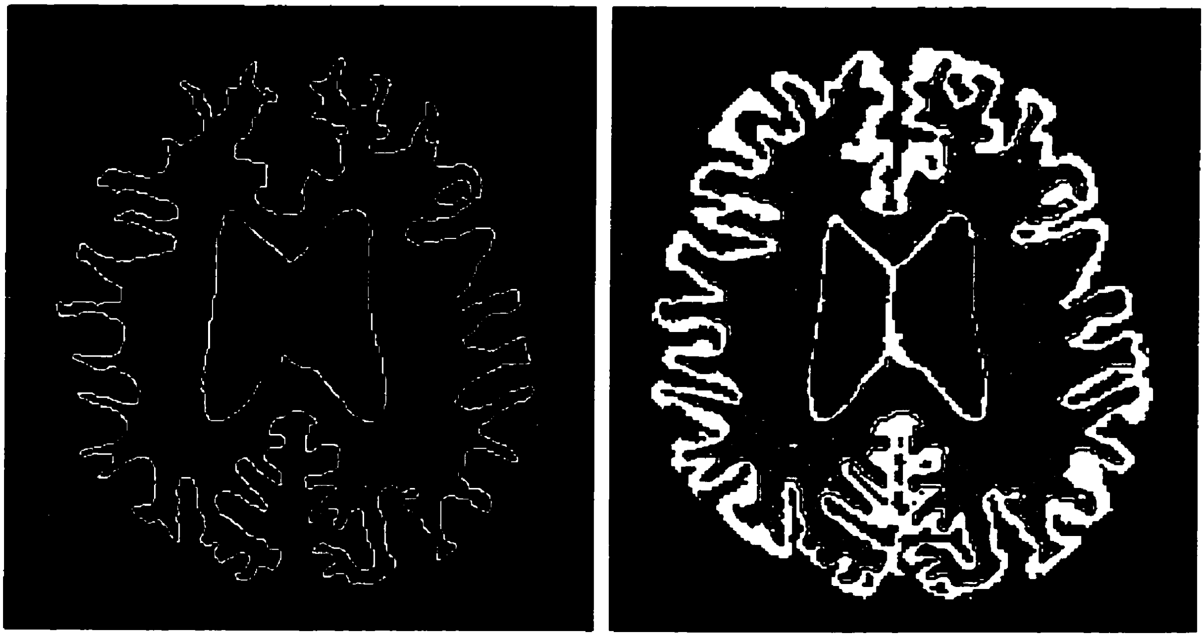
Segmenting the human cerebral cortex from MR images is an important first step in subsequent operations such as visualization, quantitative analysis, multimodal registration, and mapping and unfolding the cortex. Contours representing the highly convoluted cortex are usually determined through edge detection and linking, or contour following, or are drawn by hand. Edge detection methods suffer from the identification of either too few or too many edge points, creating problems in edge linking. Contour following methods are typically noise sensitive and cannot easily be made to satisfy known boundary conditions. Finally, drawing contours manually is labor-intensive, time-consuming and difficult to reproduce.

The use of snakes and other deformable models can overcome many of these problems. However, classical snakes often have difficulty deforming into the narrow cavities



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

Figure 7.6: T-snake used to segment gray-matter/white-matter interface in MR brain image slice.



(a) (b)

Figure 7.7: (a) Closeup of T-snake fit. (b) Fitted T-snake (black) overlaid on manually segmented image.

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

Figure 7.8: T-snake shrinking and segmenting CSF/gray matter interface in MR brain image slice.

and protrusions of the cortex. T-snakes, on the other hand, are an effective tool for this segmentation problem. In the first experiment, we attempt to extract the gray matter/white matter interface of the cortex from a 256 × 256 MR image slice. A T-snake is initialized within the white matter (Figure 7.6a), flows around the ventricles (Figure 7.6b,c) and finally conforms to the gray/white matter interface (Figure 7.6d). A pixel-resolution grid was used with model parameters: $p = 0.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.001$, and $N = 5$, and the segmentation requires about 90 seconds on an SGI Indigo workstation. Statistics of white matter pixel intensity are used to weight the T-snake inflation force (Section 5.2.2). Figure 7.7a shows a scaled-up view of the result. Figure 7.7b shows the segmentation result overlaid on a manually segmented version of the image. Evidently, the T-snake performs very well on this problem.

In the second experiment, we wrap a T-snake around an MR image slice from the same data set and allow it to shrink and conform to the outer surface (cerebrospinal fluid/gray matter interface) of the cortex (Figure 7.8a-d). A 128 × 128 cell grid was used with model parameters: $p = 0.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.001$, and $N = 5$. In this experiment we not only use statistical constraints but also the topological constraint described in Section 5.2.3. As mentioned in that section, robust skeletonization is an excellent technique for constructing structural representations of cortical topography. Figure 7.9 shows a scaled-up view of the segmentation result and Figure 7.10 shows a close-up view of the T-snake near a very narrow region of gray matter. Segmentation

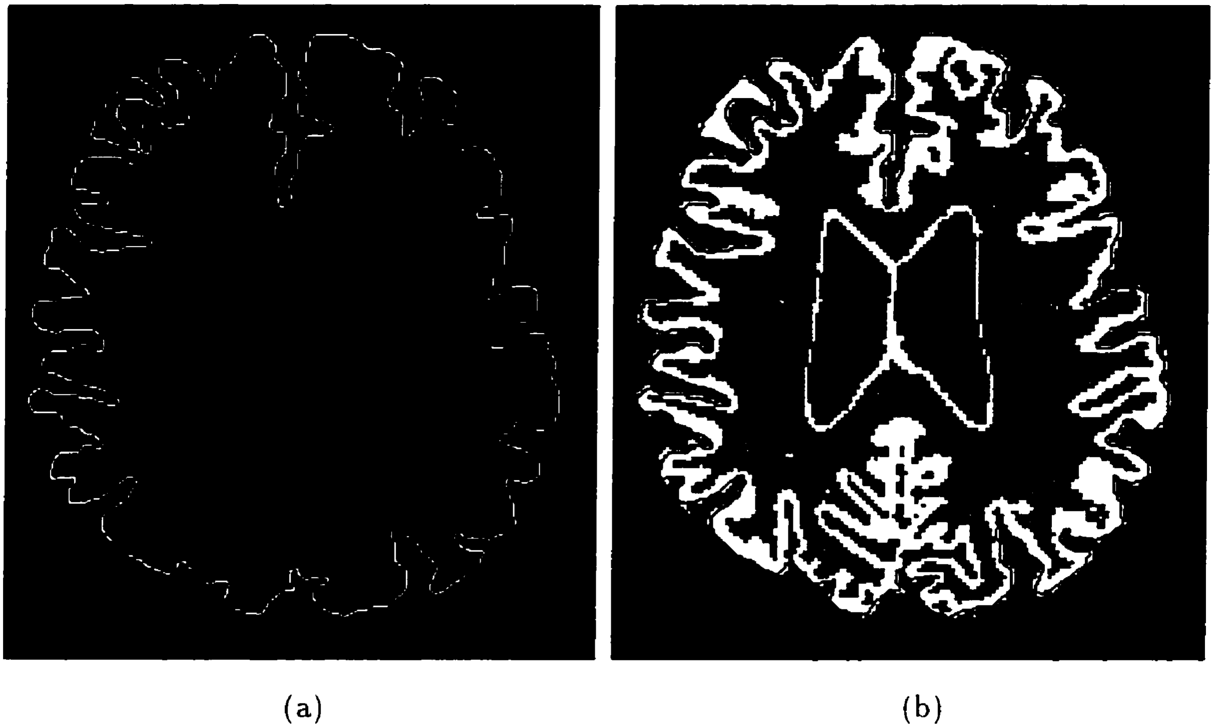


Figure 7.9: (a) Closeup of T-snake segmentation. (b) Fitted T-snake overlaid on manually segmented image.

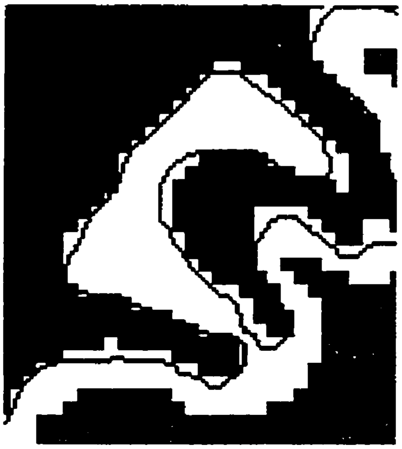


Figure 7.10: Closeup of fitted T-snake near narrow cavity. Topological constraints prevent T-snake self-intersection.

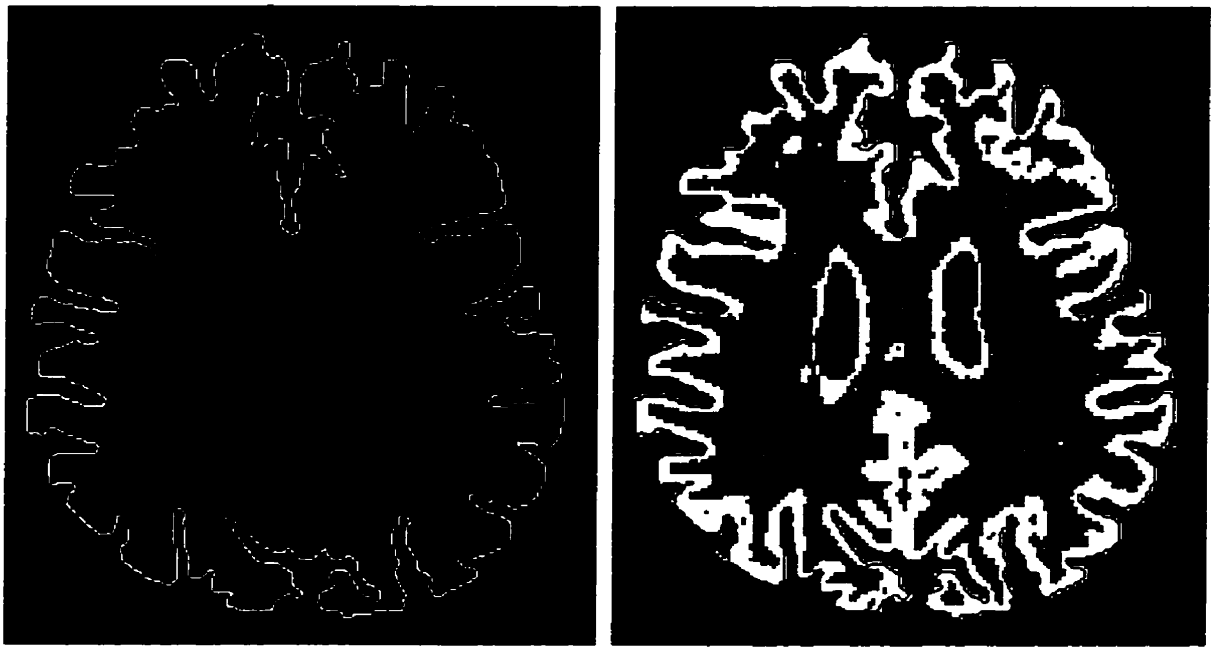

(a) (b)

Figure 7.11: (a) T-snake fitted to second MR brain image slice. (b) Fitted T-snake overlaid on manually segmented image.

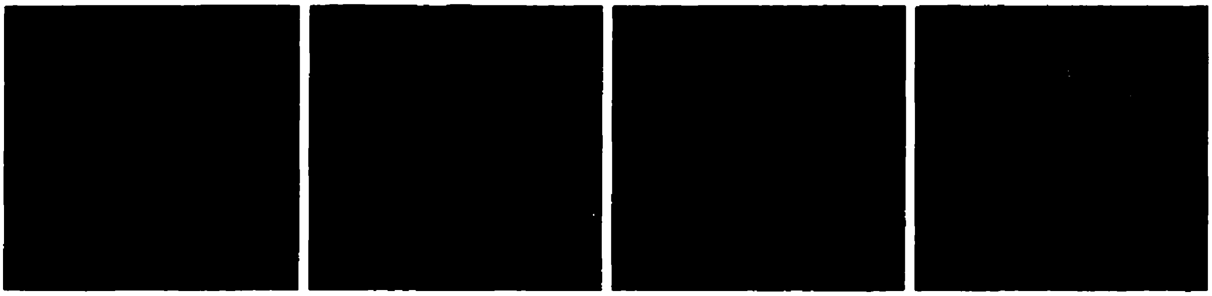
results from a second slice are also presented (Figures 7.11).

7.2.4 Segmentation of the Corpus Callosum

In this experiment we use cell growing and splitting process to segment the corpus callosum from a mid-sagittal slice of an MRI brain data set. A superquadric cell is manually seeded approximately in the center of the corpus callosum and the cell growing-splitting process is initiated (Figure 7.12a-d). Automatic seeding could be performed by computing the center of a box bounding the structure or the center of gravity of the structure. The result of the cell fitting is shown in Figure 7.13.

We "tailor" the cell growing-splitting process to fit this particular segmentation problem by

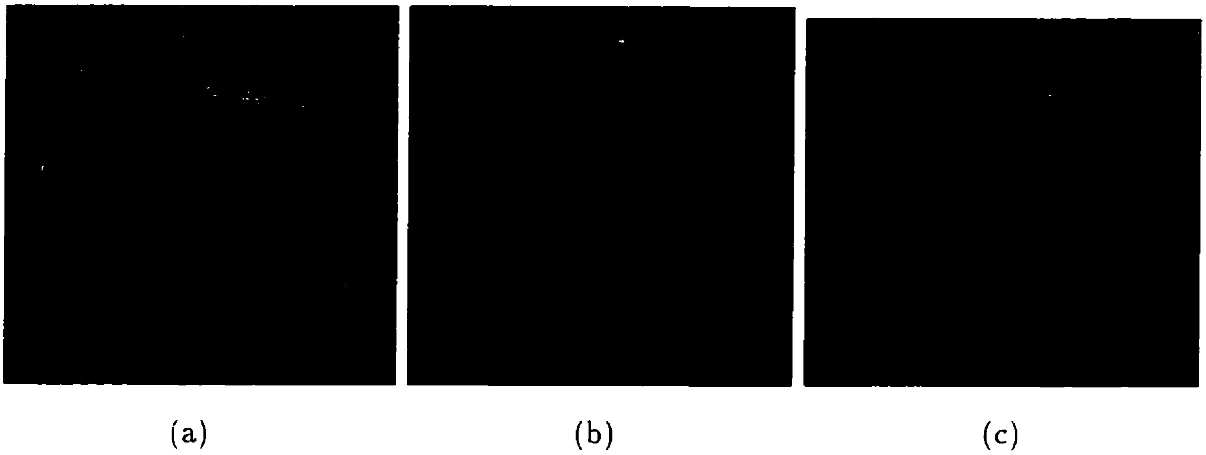
1. using a preset number of cells.
2. specifying a different superquadric "squareness" parameter for the two "end" cells and removing the point constraint with their parent cells.



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

Figure 7.12: Superquadric cells growing and splitting to recover coarse shape of corpus callosum from MR brain image slice.

Figure 7.13: Final fitted superquadric cells.



(a) (b) (c)

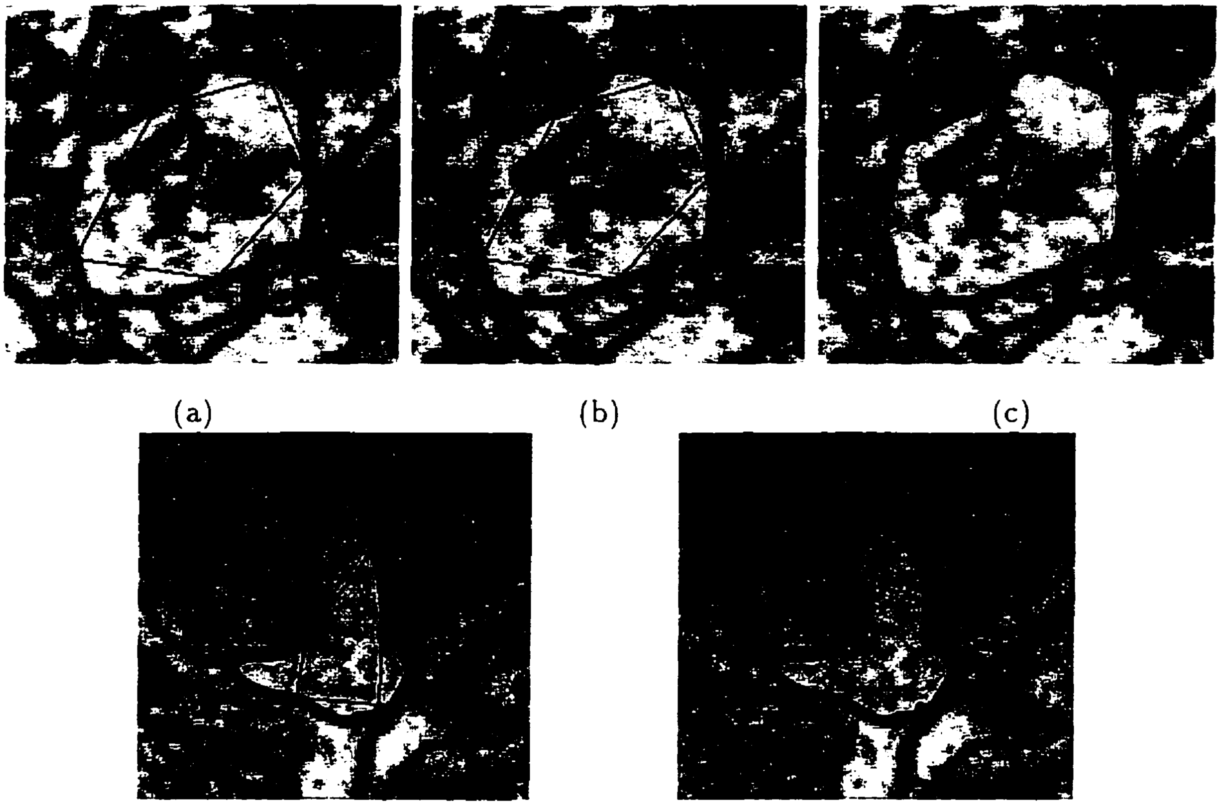
Figure 7.14: (a) Blended superquadric cells forming initial T-snake. (b) Final fitted T-snake. (c) Example of T-snake segmentation without global shape constraint.

Once the cells have extracted the rough shape of the corpus callosum, they are blended to form the initial T-snake (Figure 7.14a). The T-snake then deforms to robustly and accurately capture the shape details (Figure 7.14b). Figure 7.14(c) shows a segmentation result using a T-snake only, without the constraints imposed by superquadric cells. In this example, the T-snake has "leaked" into a neighboring structure.

7.2.5 Segmentation of Neuronal Cells

In this experiment we make use of geometric point constraints - both "soft" and "hard" constraints - to segment neuronal cells in an EM photomicrograph. We begin by defining a series of points on the cell boundary, forming a closed polygon from these points and finally converting the polygon to an initial T-snake (Figure 7.15a). The T-snake deforms and recovers the cell boundary (Figure 7.15b) and is constrained to pass through the initial user-defined points (Figure 7.15c). The segmentation requires under a second on an SGI Indigo workstation and only a few seconds to select the initial points. A second example is shown in Figure 7.15c,d.

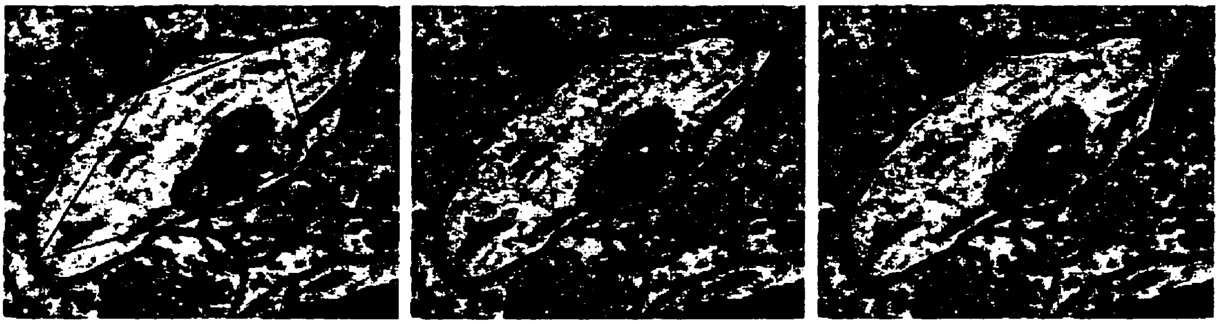
In a third experiment we make use of soft geometric point constraints in the form of anchored springs. The springs exert forces on T-snake nodes that are within a specified neighborhood of the springs. In Figures 7.16a,b the initialization and result of the T-snake segmentation without spring constraints is shown. The T-snake has attached itself



(a) (b) (c)

(c) (d)

Figure 7.15: T-snake (derived from closed polygon) used to segment neuronal cells. The T-snake is constrained to pass through the initial user-specified points.



(a) (b) (c)

Figure 7.16: (a) (b) T-snake formed from user-specified closed polygon deforms to segment neuronal cell. (c) Soft point constraints are used to pull T-snake off spurious edges.

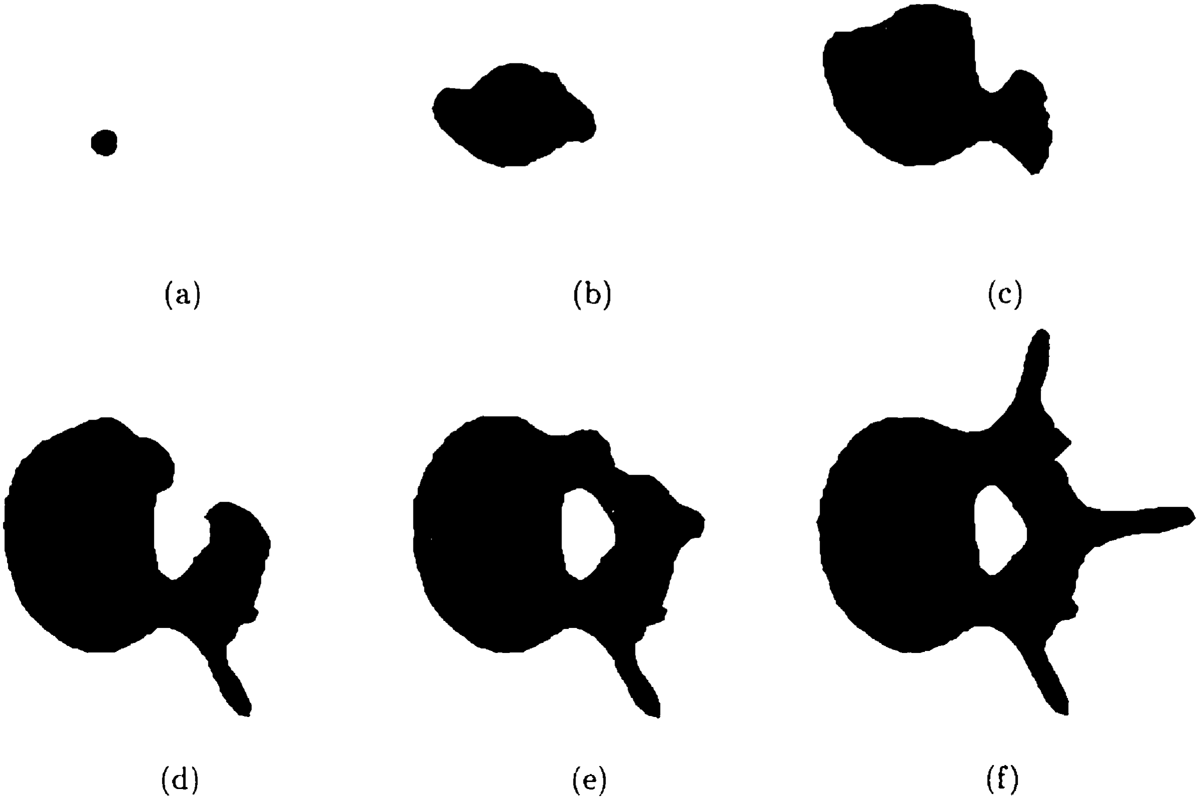


Figure 7.17: T-surface segmenting vertebra phantom from CT image volume.

to several interior parts of the cell. Figure 7.16c shows the addition of spring constraints (the springs were added dynamically) has resulted in a correct segmentation of the cell boundary. All experiments use a 160 × 122 cell grid on the 640 × 488 pixel image and model parameters: $p = 51.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.005$, and $N = 10$.

7.3 Volume Image Segmentation with T-Surfaces

7.3.1 Vertebra Phantom Segmentation and Reconstruction

In the first example we apply a T-surface to a 120 × 128 × 52 CT image volume of a human vertebra phantom to demonstrate the topological adaptability of the 3D model (Figure 7.17). We use a 32 × 30 × 13 cell grid (where each cubical cell is divided into 6

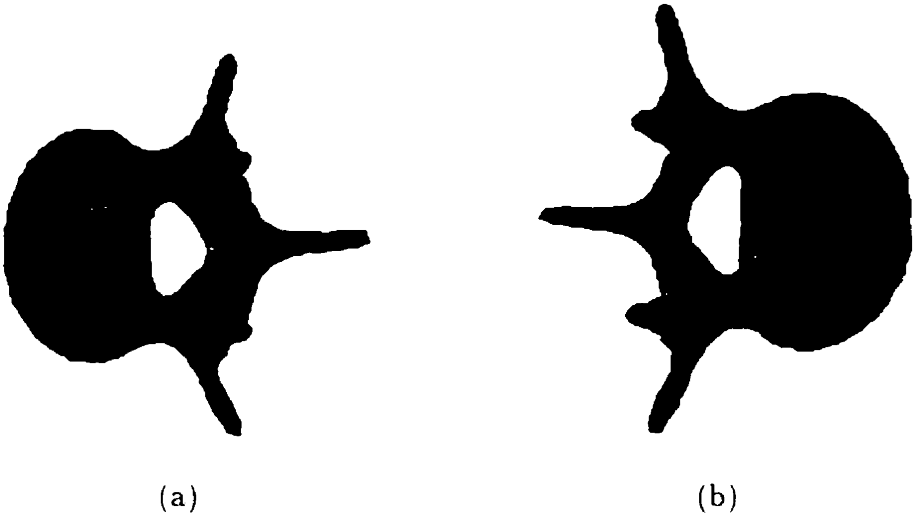


Figure 7.18: Two views of the fitted T-surface.

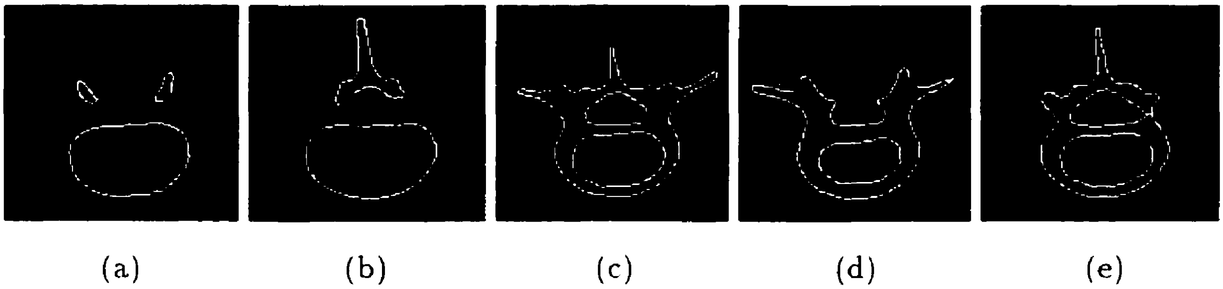


Figure 7.19: Several cross-sections of T-surface overlaid on CT vertebra image volume slices.

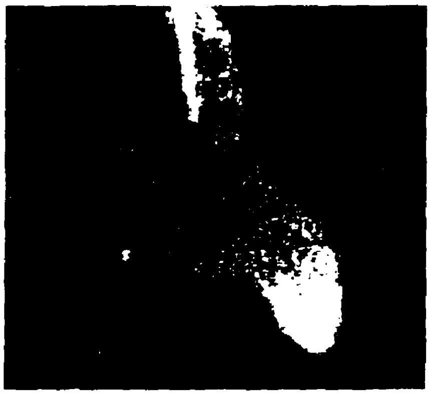


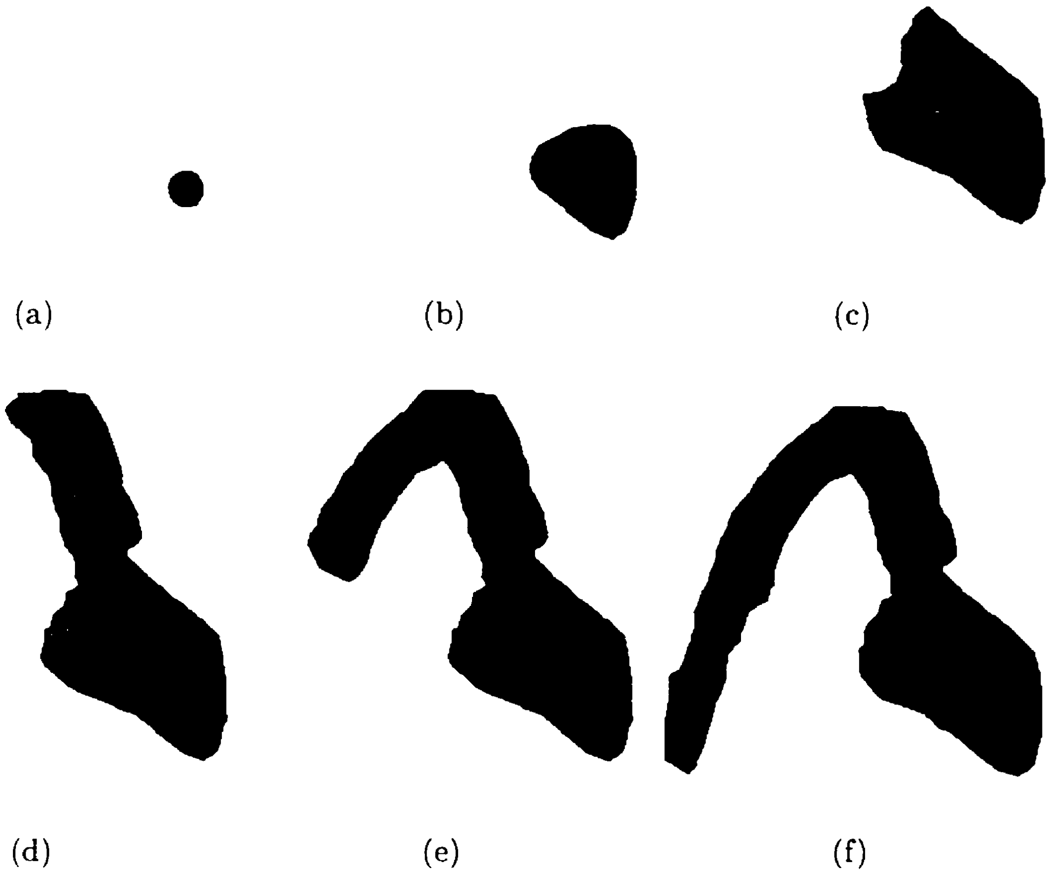
Figure 7.20: CT sagittal image slice of canine heart.

tetrahedra) with model parameters: $p = 20.0$, $q = 80.0$, $a = 20.0$, $b = 60.0$, $\Delta t = 0.001$, and $N = 10$. Figure 7.18 shows two views of the final result after conversion to a classical deformable surface. Figure 7.19 shows several cross-sections of the fitted model.

7.3.2 Segmentation of the Left Ventricle and Aorta

In this experiment we segment and reconstruct the left-ventricular (LV) chamber and aorta from 3D CT images of a canine heart. The image volume dimensions are $128 \times 128 \times 118$ and a $20 \times 20 \times 20$ resolution grid was used (model parameters: $p = 61.0$, $q = 60.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.002$, and $N = 10$). The volumetric heart data comes from the dynamic spatial reconstructor (DSR), a high speed volumetric X-ray CT scanner. Each slice of the image volume represents an approximately 0.9 mm thick transverse cross-section of the scanned anatomy and each voxel represents a $(0.9mm)^3$ cube of tissue. In order to bring out the LV as a bright object, a Roentgen contrast agent is injected into the right atrium several seconds prior to the scan of the left ventricle.

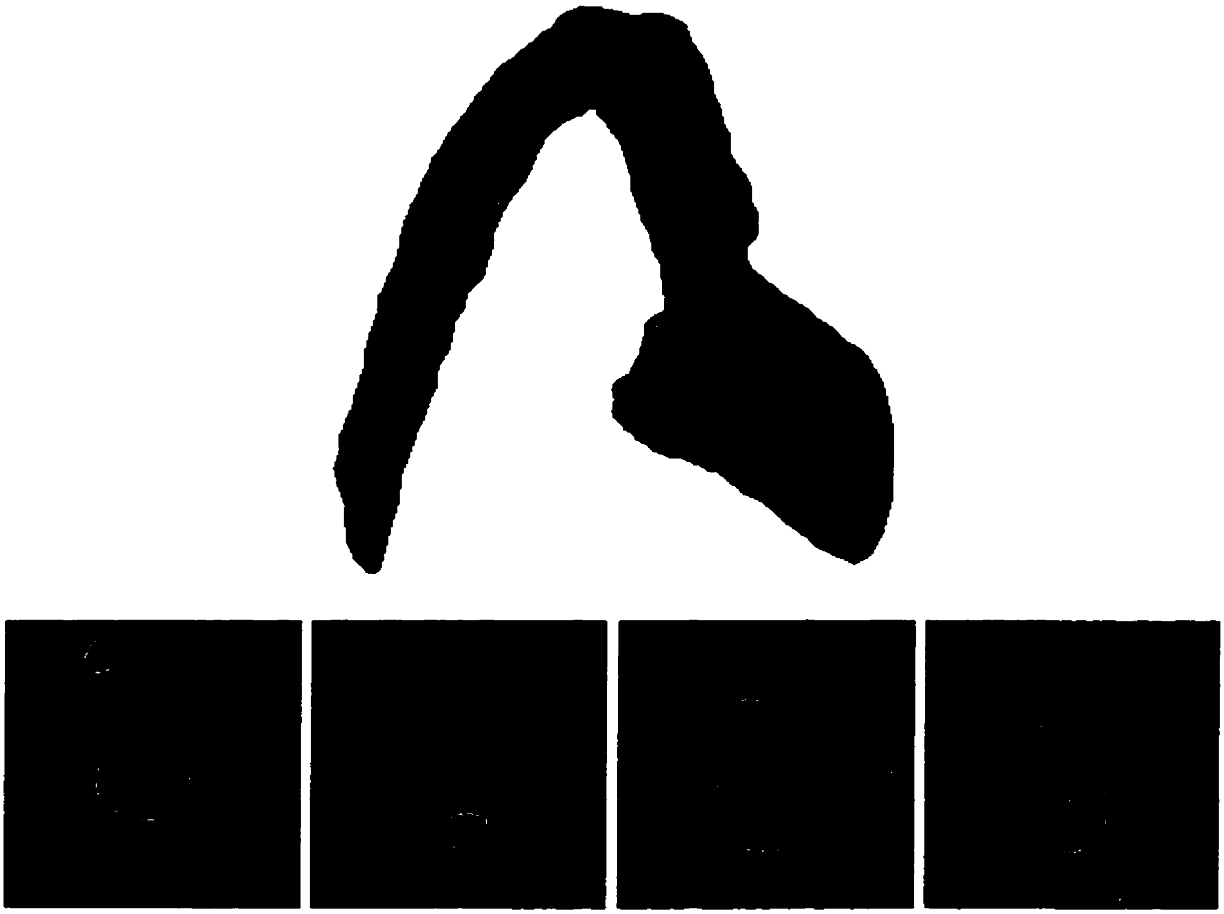
Fig. 7.20 depicts a sagittal (y-z plane) slice of a canine heart. The bright (high intensity) regions include the LV chamber, left atrial (LA) chamber, and aorta. The surrounding lower intensity region is the myocardium. The portion of the LV chamber near the valves is referred to as the base, and the lower tip is known as the apex. In this experiment, we manually seed the LV with a small T-surface and the segmentation then proceeds automatically (Figure 7.21). The inflation force is weighted with LV region image intensity statistics to reinforce the image edge forces. The final result, after



(a) (b) (c)

(d) (e) (f)

Figure 7.21: T-surface segmenting LV and aorta from CT image volume.



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

Figure 7.22: Final fitted T-surface and four cross-sections of the model.

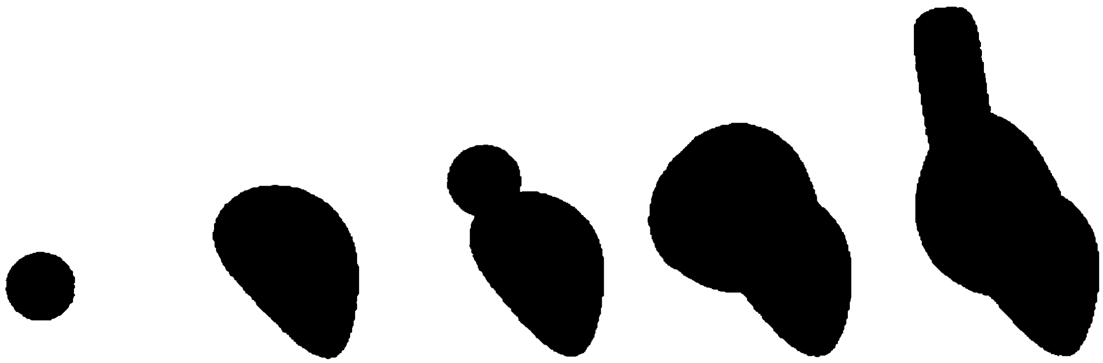
discarding the grid and converting to a classical deformable surface is shown in Figure 7.22. Figures 7.22a-d depict several cross-sections of the fitted model. While there are a few cross-sections that may require manual editing, most of the model fits very accurately (using visual inspection) and the entire process takes under 4 minutes on an SGI Indigo 2 workstation. The LV/aorta segmentation is also highly reproducible: as long as the T-surface is seeded within the bright region of the LV, it produces almost identical results.

7.3.3 LV Segmentation with T-Surfaces and Superquadrics

In this experiment, we apply the 3D cell growing and splitting process to segment the LV. We initially seed a superquadric cell near the LV apex and start the cell growing-splitting process (Figure 7.24. In this simple example we limit the number of cells to three. This number is sufficient to robustly capture the coarse shape of the LV and the initial portion of the aorta. Figure 7.24f shows the result of the blended superquadric cells. This isosurface is converted to an initial T-snake which then deforms to capture the shape details of the LV (Figure 7.24g). This experiment demonstrates the potential of T-surface global shape constraints to form the basis of a completely automatic segmentation and analysis technique.

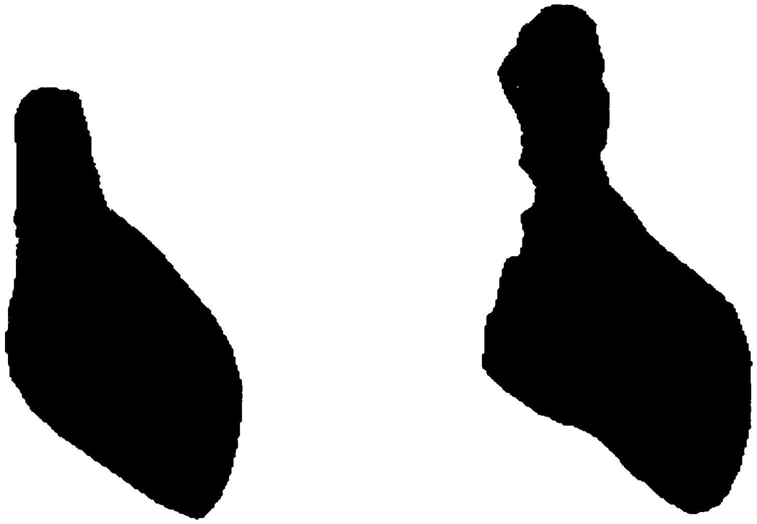
7.3.4 Segmentation of Brain Ventricles

In this experiment we use a T-surface to extract the major brain ventricles from a $256 \times 256 \times 85$ ($0.9375mm \times 0.9375mm \times 1.5mm$) MR image volume (Figure 7.25). The volume was interpolated to produce cubical voxels resulting in 136 slices rather than 85. A $128 \times 128 \times 68$ resolution grid was used, with model parameters $p = 30.0$, $q = 48.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.0008$, and $N = 10$, as well as a region statistics-weighted inflation force. Since the T-surface provides a boundary and interior representation, the volume occupied by the model can be be quickly and accurately calculated. Brain ventricle volumes are useful indicators of many brain pathologies. Figures 7.25c-g show several cross sections of the fitted model.



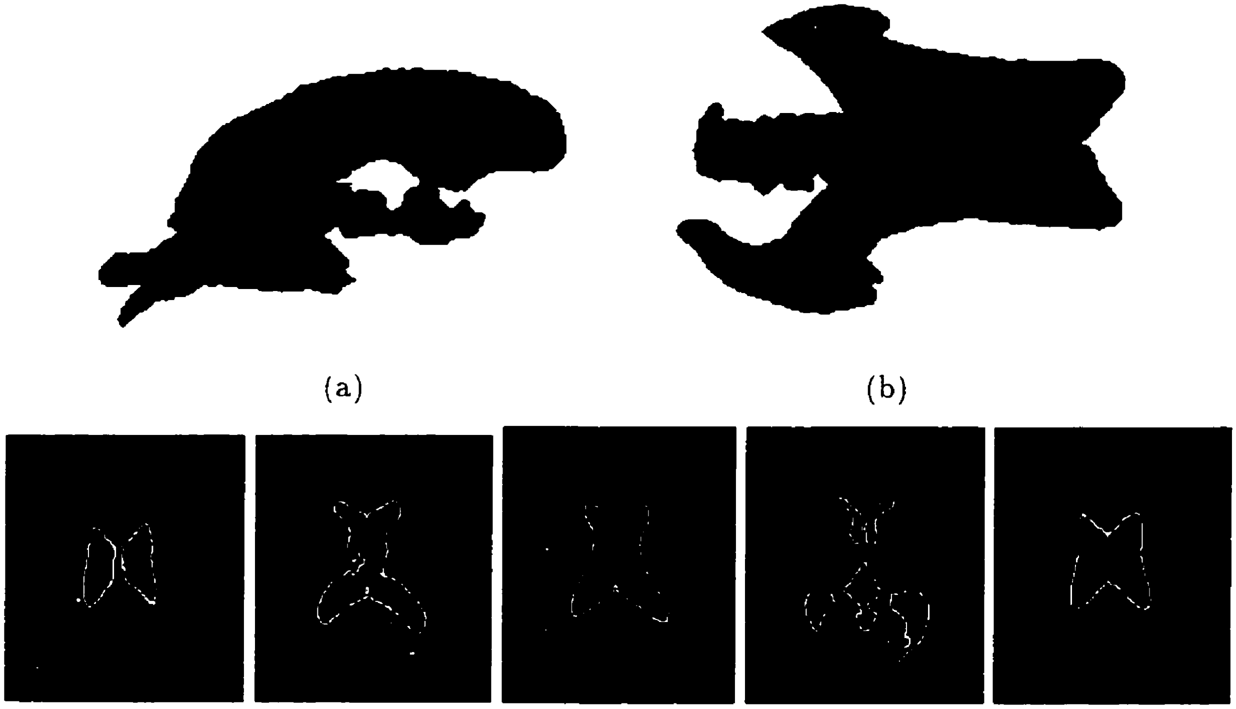
(a) (b) (c) (d) (e)

Figure 7.23: Superquadric cells growing and splitting to segment coarse shape of LV and part of aorta.



(f) (g)

Figure 7.24: (f) Blend of superquadric cells to form initial T-surface. (g) Final fitted T-surface.



(a) (b)

(c) (d) (e) (f) (g)

Figure 7.25: T-surface segmentation of ventricles from MR image volume of the brain.

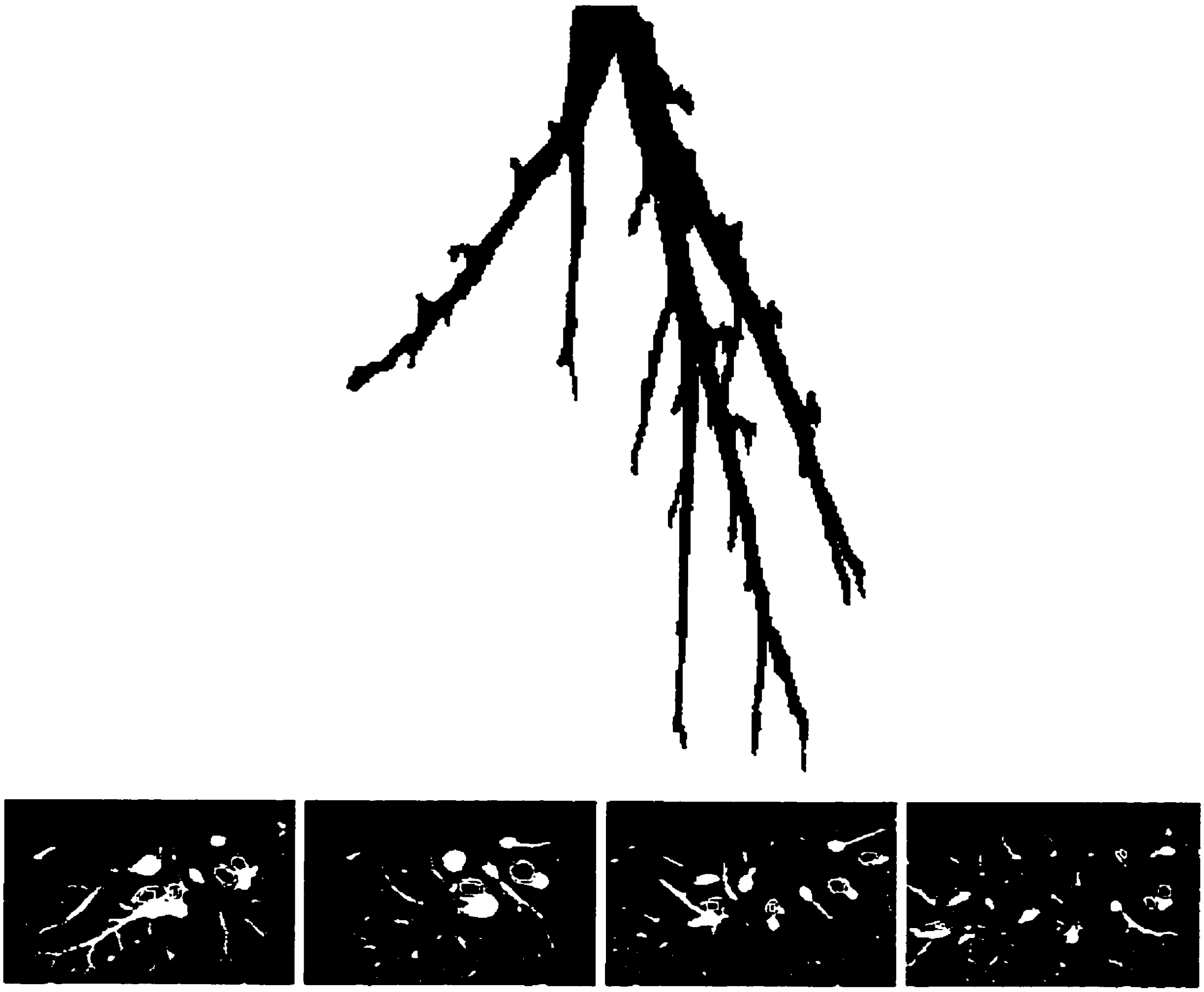
7.3.5 Segmentation of Airway Structures of the Lung

Accurate quantitative measurments of airway and vascular dimensions are essential for evaluating function in both the normal and the diseased lung. Image volumes of these pulmonary tree structures can be generated using High Resolution Computed Tomography (HRCT). To obtain quantitative data, each tree can be reduced to its central axis. From the central axis, branch length is measured as the distance between two successive branch points, branch angle is measured as the angle produced by two child branches, and cross-sectional area is measured from a plane perpendicular to the central axis point.

Robust, automatic central axis extraction is dependent on accurate, noise-free segmentations. In this experiment, serially scanned two-dimensional slices of the lower left lobe of an isolated canine lung were stacked to create a volume of data. The volume dimensions are $256 \times 256 \times 133$, each slice is $2mm$ in thickness and they are $1mm$ apart (there is overlap between the successive images). The image volume was interpolated to produce cubical voxels resulting in 266 slices instead of 133. The airway is then extracted and reconstructed using a T-surface seeded at the root of the tree structure (Figure 7.3.5). A voxel resolution grid was used with model parameters: $p = 00.0$, $q = 50.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.001$, and $N = 10$. Since the openings to branches of the airway structure can be very narrow, edge extraction techniques can produce incorrect results near these regions; false edges may be created that completely block these openings. For this reason, we set the image edge force weight p to zero and use the image threshold intensity force (Equation 4.4) only. Although the T-surface performed well on the major branches (Figure 7.3.5a-d), there is some over-segmentation on the tiny branches. An adaptive image threshold technique may ameliorate the over-segmentation problem. Furthermore, some over-segmentation may be tolerated by central axis extraction algorithms without significantly affecting the result [118].

7.3.6 Segmentation of Cerebral Vasculature

The introduction of Magnetic Resonance Angiography (MRA) and spiral CT Angiography into clinical routine protocols has made the acquisition of vascular volume data sets feasible. Typically the acquired data consist of a stack of up to 150 images with 256^2 or



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

Figure 7.26: T-surface segmentation of the airway structures of a canine lung from HRCT image volume. Several cross-sections of the fitted model are also shown.

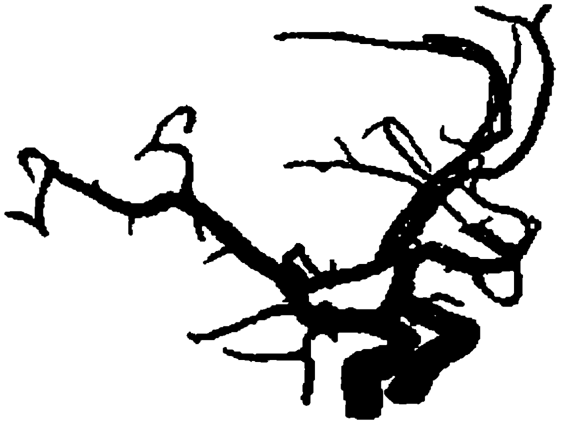


Figure 7.27: T-surface segmentation of cerebral vasculature from MRA image volume.

512^2 pixels, a spatial resolution of less than $1.0mm$ and an intensity resolution of 12 bits. Region growing techniques have been used with some success to segment the strongly connected structures of the vascular system. When combined with statistical constraints, T-surfaces behave as an active region growing model that is also able to integrate edge information and filter out noise using the model smoothness constraints.

We have used a T-surface to segment and reconstruct the vascular system of the brain from an MRA image volume. The data consist of a stack of 100 slices each with 512^2 pixels, 2 bytes per pixel (voxel size $0.4296875mm \times 0.4296875mm \times 0.7mm$). The image volume was interpolated to produce cubical voxels resulting in 162 slices. A T-surface is seeded at the root of the vessel tree and then flows into the vessels, automatically extracting the vascular system (Figure 7.3.6). A voxel resolution grid was used (model parameters: $p = 00.0$, $q = 80.0$, $a = 20.0$, $b = 60.0$, $\Delta t = 0.0005$, $N = 10$) and the entire segmentation process takes approximately 1 hour on an SGI Reality Engine.[1]

Although both X-ray and MR angiography produce high contrast images, they can have a large deviation in the gray scale range. This large range makes it difficult to segment the images with simple threshold techniques. Currently, the T-surface inflation force is weighted using *global* image statistics and, consequently, some oversegmentation

---

[1] The limited memory on the SGI Reality Engine (64M bytes) coupled with the massive MRA image volume and program memory requirements (> $100M$ bytes) result in voluminous disk activity and disk thrashing. This factor significantly affects segmentation performance. Increased memory capacity coupled with parallel machine architectures should result in dramatic performance improvements.

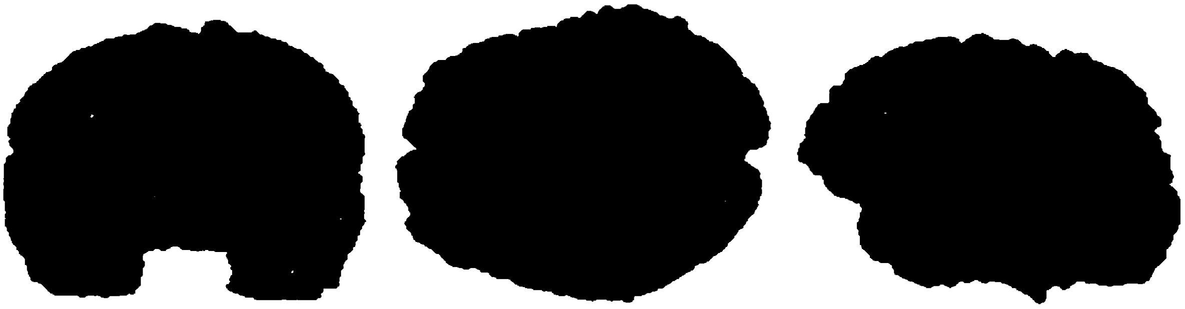
can result. As mentioned previously, the use of an adaptive threshold, based on *local* gray scale statistics, may overcome this problem [87]. This topic is not explored in this thesis and is a subject of future research.

7.3.7 Segmentation of the Cerebral Cortex

The analysis of cerebral functional data obtained from positron-emission tomography (PET) or magneto-encephalography (MEG), often requires complementary anatomical information obtained from MR images. As mentioned in Section 5.2.3, a promising technique for the interpretation of cerebral cortex topography from 3D MR images is 3D skeletonization. Due to the noise sensitivity of the skeletonization algorithms, it is desirable to produce a smooth (noise-filtered) segmentation of the cortex which is guaranteed to have the same topology as the real object. In this experiment, we use a T-surface with a region statistics-weighted inflation force as well as a global topological constraint to segment the cerebral cortex from a preprocessed MR image volume (the skull region has been manually removed from the image) (Figure 7.28). A $64 \times 64 \times 34$ resolution grid was used for the $256 \times 256 \times 136$ voxel image volume (model parameters: $p = 0.0$, $q = 48.0$, $a = 20.0$, $b = 40.0$, $\Delta t = 0.0008$, and $N = 10$). The T-surface was initialized to surround the cortex and then shrinks and conforms to the CSF/gray matter interface. The automatically subdividing T-surface is able to penetrate, with good success, into the narrow and deep cavities of the highly convoluted cortex. A more accurate result could be obtained with a finer resolution grid at the expense of increased computation time and a larger number of model triangles.

Two techniques that could potentially significantly improve the performance and accuracy of this particular segmentation scenario are the following:

- The T-surface could be allowed to initially extract the gray-matter/white-matter interface of the cortex. From this position, the topologically constrained T-surface could expand out towards the CSF/gray matter interface. This process could ensure that the T-surface completely penetrates all of the narrow and deep cavities of the cortex.



(a) (b) (c)

Figure 7.28: T-surface segmentation of the cerebral cortex from MR image volume. (a) front view, (b) top view, (c) side view.

- To improve performance, a coarse resolution T-surface could initially be used to segment the cortex in a smoothed image volume. This process could be repeated for progressively finer resolution T-surfaces and image volumes until a desired accuracy is reached.

Chapter 8

Conclusions

The increasingly important role of medical imaging in the diagnosis and treatment of disease has opened an array of challenging problems centered on the computation of accurate geometric models of anatomic structures from medical images. Deformable models offer an attractive approach to tackling such problems, because these models are able to represent the complex shapes and broad shape variability of anatomical structures. Deformable models overcome many of the limitations of traditional low-level image processing techniques, by providing compact and analytical representations of object shape, by incorporating anatomic knowledge, and by providing interactive capabilities.

The challenge is to increase the degree of automation of deformable models to an extent that the labor-intensive aspects of interactive anatomic structure segmentation are reduced to a basic minimum, while the accuracy, efficiency, and reproducibility of the segmentation is maximized. Furthermore, many segmentation scenarios now exist that are suitable for complete automation. This thesis proposed an affine cell decomposition-based deformable model framework in an effort to meet these challenges. This simple, elegant framework provides curve, surface, and solid deformable models with the ability to adapt to arbitrary object geometry and topology. It produces intrinsic model parameterizations and insensitivity to model initial conditions. By maintaining the traditional parametric physics-based formulation of deformable models, ACD-based deformable models are able to incorporate *a priori* anatomic knowledge in the form of energy and force-based constraints, and provide intuitive interactive capabilities. By defining the models in terms of the affine cell framework, "hard" geometric, topological, and global shape constraints can be efficiently realized. These combined constraints produce models that are insensitive to spurious image features and ensure that real boundaries of objects are recovered. and create a sound foundation for automatic image analysis.

The simplicial cell complex or grid is constructed using an on-demand scheme which results in minimal memory or computational overhead. The novel iterative simplicial approximation or reparameterization process provides a simple. effective means of tracking the activity ("temperature") of individual model elements; elements can be removed from the computation once their activity falls below a threshold level. significantly improving model performance. At the heart of the reparameterization process are robust. inherently parallel geometric algorithms that utilize simple. fast algebraic operations. These combined factors result in an efficient implementation that usually enhances shape recovery performance. With the advent of parallel machines architectures. we expect further significant performance gains.

We have demonstrated the power and flexibility of ACD-based deformable models by using our prototype interactive segmentation system to extract and reconstruct a wide array of objects from medical images in a highly automated manner. We have been able to process some the largest image volumes currently produced by modern imaging technologies. Furthermore. we have demonstrated the potential for fully-automatic medical image processing by combining ACD-based models with a variety of constraint mechanisms. Finally. our framework is designed in a hierarchical fashion that is built on top of a classical deformable snakes and surfaces. The modular design allows the ACD framework to be easily incorporated into existing variants and implementations of these models.

8.1 Comparison of T-Snakes/T-Surfaces to Alternative Models

In this section we compare T-snakes/T-surfaces to two models reviewed in Chapter 3 in order to underscore the viability our technique. Specifically. we show that ACD-based deformable models combine several advantageous features of both parametric and implicit models. while at the same time eliminate some of their difficulties.

8.1.1 Comparison to Dynamic Particle Systems

Like the active particle system of Szeliski *et al.* [96], T-surfaces (or T-snakes) can be viewed as a particle system that evolves according to Lagrangian mechanics. Both models use inter-particle forces to maintain the smoothness of the surface (or contour), and both use external image forces to attract particles to the image features. T-snakes use the simplicial grid to automatically add particles (model elements) and perform topological transformations, and particle connectivity is explicitly maintained throughout the evolution of the model. Szeliski's technique seeds particles on the surface of the object and particles are added until their density on the surface is within some threshold value. Good initial seed particle sites may be difficult or cumbersome to find, especially automatically. A T-snake on the other hand, can be quickly seeded anywhere within the object of interest or surrounding the object. To deal with the $O(N^2)$ complexity of inter-particle interaction, Szeliski spatially hashes particle positions to reduce the complexity to $O(N \log N)$. T-snakes maintain explicit particle connectivity via the simplicial grid: inter-particle interaction complexity is constant. Furthermore, while Szeliski's model is topologically adaptable, it is a strictly parametric approach. T-snakes can be considered both an implicit model — space is partitioned into an interior, boundary and exterior — as well as a parametric model. This property allows topological, statistical and global shape constraints to be imposed on the model. One advantage of Szeliski's particle system over T-snakes is the ease with which it is able to form open surfaces, as well as closed surfaces.

8.1.2 Comparison to Level Set Evolution Techniques

As discussed in chapter 3 several researchers have adopted the level set evolution technique developed by Osher and Sethian [74] to the problem of shape recovery [13, 59, 115, 14, 88] (see Appendix A for a detailed discussion of level set evolution). These models are formulated in terms of a partial differential equation, based on curvature dependent motion, and are not the result of minimizing an energy functional. They are viewed as an evolving contour which is the level set of some implicitly defined surface. While the Osher-Sethian front propagation technique provides a well defined mathematical framework, the strictly implicit model formulation is not as convenient as a parametric formulation for interactive guidance by expert users or high-level control mechanisms and in terms of computational burden. In particular, by using the higher dimensional surface to define and control the evolving contour, the implicit formulation loses the flexibility to impose arbitrary constraints in the form of forces or energy functions or other geometric or topological constraints; it may be difficult, if not impossible, to translate these constraints into the higher dimensional representation. Furthermore, the numerical implementation of the level set evolution techniques requires, for every iteration, a (bounded) search of the domain for the new level set. The implementation can also require massive amounts of computer memory to store the fixed domain points (used in the Eulerian formulation to define the level set) unless special data structures are used.

A T-snake is a hybrid model that maintains a parametric formulation while providing the capabilities of an implicit model without resorting to a higher dimensional representation. By maintaining a parametric formulation, T-snakes, like classical snakes, are able to easily incorporate soft constraints expressed as energy functions or force functions and provide interactive capabilities. T-snakes are also able to incorporate hard constraints through the simplicial grid and by maintaining direct control of the T-snake boundary evolution. The parametric formulation also allows for a simple, and "cheap" (in terms of computational cost and memory) implementation. The computational algorithm is composed of basic, well-defined algebraic operations and uses a few standard, efficient data structures such as lists, and hash tables. For this reason, as mentioned previously, we have been able to apply T-snakes and T-surfaces to even the most massive medical data sets.

8.2 Research Directions

Our experience with ACD-based deformable models has revealed several issues that are relevant to the continued development and success of this approach. This section summarizes these issues and indicates some potential research directions.

8.2.1 Model Simplification

A geometric model of shape may be evaluated based on the parsimony of its formulation, its representational power, and its topological flexibility. Generally, parameterized models offer the greatest parsimony, free-form (spline) model feature broad geometric coverage, and implicit models have the greatest topological flexibility. Deformable models have been developed based on each of these geometric classes. The hybrid ACD-based deformable models are an attempt to combine some of the complementary features of these geometric classes. However, in certain scenarios, such as the segmentation and modeling of complex, multipart structures like arterial trees, or topologically complex structures such as vertebrae, a large number of model triangles can be produced. In order to reduce the number of triangles, simplification algorithms are needed [107, 38]. These algorithms could attempt to take advantage of the hierarchical nature of the simplicial grid. Alternatively, regions of triangles could be replaced with more compact representations such as high-order finite element or spline patches [29].

8.2.2 Robustness

Ideally, a deformable model should be insensitive to initial conditions and noisy data. Deformable models, such as T-snakes and T-surfaces, are able to exploit multiple image attributes and high level or global information to increase the robustness of shape recovery. Strategies worthy of further research include the incorporation of shape constraints into the deformable model framework that are derived from low level image processing operations such as thinning, medial axis transforms, or mathematical morphology [84]. A classical approach to improve the robustness and performance of model fitting is the use of multiscale image preprocessing techniques [47, 105]. These techniques would blend nicely with the multiresolution ACD-based models.

8.2.3 Curve versus Surface versus Solid Models

The earliest deformable models were curves and surfaces. Anatomic structures in the human body, however, are either solid or thick-walled. To support the expanding role of medical images into tasks such as surgical planning and simulation, and the functional modeling of structures such as bones, muscles, skin, or arterial blood flow, may require volumetric or solid deformable models rather than surface models. For example, the planning of facial reconstructive surgery requires the extraction and reconstruction of the skin, muscles, and bones from 3D images using accurate solid models. It also requires the ability to simulate the movement and interactions of these structures in response to forces, the ability to move, cut and fuse pieces of the model in a realistic fashion, and the ability to stimulate the simulated muscles of the model to predict the effect of the surgery. Several researchers have begun to explore the use of volumetric or solid deformable models of the human face and head for computer graphics applications [50, 30] and for medical applications, particularly reconstructive surgery [114, 28, 78, 37] and there is much room for further research. T-surfaces are well-suited for this research direction since they automatically generate solid models through the tetrahedral grid. The solid models generated by the segmentation and reconstruction of artery trees presented in Chapter 7 for example, can serve as the basis for performing complex fluid flow analysis. Furthermore, incorporating physical properties such as the vessel elasticity into the models would allow for their use in a surgical planning environment.

8.2.4 Automatic Initialization, Shape Recovery and Analysis

Completely automatic medical image segmentation using deformable models requires the automatic "seeding" or initialization of the models, and the incorporation of *a priori* anatomical knowledge to ensure robust shape recovery. Automatic interpretation of these images requires models that not only describe the size, shape, location and orientation of the target object but that also permit expected variations in these characteristics. ACD-based deformable models combined with topological and/or global shape constraints are an attempt to meet some of these requirements. Further research is needed to create automatic model initialization techniques. In addition, further development, formalization and generalization of the global shape constraint mechanism is required. Finally, the combination of T-surfaces with topological constraints and deformable anatomical atlases is a very promising approach that is worthy of future study.

Appendix A

Level Set Curve Evolution

The Euclidean heat flow equation

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial s^2} = \kappa \cdot \mathbf{N}, \tag{A.1}$$

where $C(s,t) = (x(s,t), y(s,t))$ is a parameterized planar curve, s is arc length, t is time, κ is curvature, and $\mathbf{N}$ is the unit inward normal, is well known for its excellent geometric smoothing properties [34, 39]. The flow decreases the total curvature of the curve as well as the number of zero crossings and the value of curvature maxima/minima. This equation is also known as curve shortening flow since it evolves the curve in the gradient direction of its arc length. Kimia *et al.* [48] have studied a generalization of this equation

$$\frac{\partial C}{\partial t} = \alpha(\kappa) \cdot \mathbf{N}, \tag{A.2}$$

in the context of constructing a representation of shape in computer vision. In particular, they analyze planar curve evolution by a linear speed function $\alpha(\kappa)$:

$$\frac{\partial C}{\partial t} = (\alpha_0 + \alpha_1 \kappa) \cdot \mathbf{N}, \tag{A.3}$$

where α_0, α_1 are constants. The constant velocity term $\alpha_0\mathbf{N}$, called the *reaction* component by Kimia *et al.*, is a hyperbolic term and makes the evolution nonlinear. The term can lead to *shocks* or singularities when the characteristic curves of (A.3) collide [94]. In order to obtain physically correct solutions, the classical entropy condition for a propagating flame front is used [74]: *once a particle is burnt, it stays burnt.* The constant velocity term is related to classical mathematical morphology [81] and shape offsetting in CAD [45]. In [89], the authors show that all mathematical morphology operations (for convex structuring elements) are obtained via curve evolution using the constant velocity term only, where the coefficient of the term depends on the structuring element. The term $\alpha_1 \kappa \mathbf{N}$ is called the *diffusion* term. It provides a quasi-linear, parabolic behavior to the evolution. Based on these principles, Kimia *et al.* define a reaction-diffusion scale space of shapes parameterized by (α_0, α_1). That is, the scale space is the result of deforming shape by various combinations of reaction and diffusion. The different types of shocks or singularities that form in this space are equivalent to shape skeletons [11], and together with the time of their formation are used to derive a set high-level shape descriptors.

The numerical implementation of the curve evolution equation faces a number of problems, including the handling of topological changes, capturing discontinuities, and the build-up of error. For example, using a Lagrangian formulation of the equation can lead to numerical instabilities as small errors in the computed curve particle positions can be amplified by the curvature term. These problems can be resolved by the use of level set evolution, proposed by Osher and Sethian for modeling flame propagation [74], introduced to computer vision by Kimia et al., and first applied to segmentation using active contours in [13, 59]. This method uses techniques from the theory of hyperbolic conservation laws to successfully compute the curve evolutions. Osher and Sethian show that by embedding the curve in a two dimensional surface and then evolving the surface rather than the curve, one derives robust, stable, and reliable evolution equations. Formally, the curve C is considered as the 0th level set of an evolving surface $\phi(x, y, t)$. The evolution of the surface is defined by a Hamilton-Jacobi-like equation

$$\frac{\partial \phi}{\partial t} = \beta \cdot |\nabla \phi|, \qquad (A.4)$$

where β is an arbitrary speed function. The advantages of this formulation are that the parameterization is defined in terms of the intrinsic local geometry of the curve, the formulation is stable and ϕ remains a function as long as β is smooth, while the level sets of ϕ can arbitrarily change topology. Furthermore, the intrinsic properties $(\mathbf{N}, \kappa)$ of the contour can be easily determined and the method is extensible to higher dimensions. One disadvantage of the approach is that it requires a representation one dimension higher than that of the original data.

To apply the level set technique to shape recovery problems, Caselles *et al.* [13] and Malladi et al. [59] independently formulated the problem as the steady state solution of the evolution ($\phi(0,C) = \phi_0(C)$)

$$\frac{\partial\phi}{\partial t} = g(I)|\nabla\phi|(div(\frac{\nabla\phi}{|\nabla\phi|}) + c) = g(I)(c+\kappa)|\nabla\phi|, \qquad (A.5)$$

where $g(I) = \frac{1}{1+(\nabla G_\sigma * I)^2}$ is a negative speed function that stops the curve at object boundaries, c is a constant speed term similar to the inflation force of a snake, $G_\sigma * I$ is the convolution of the image I with the Gaussian G_σ. The gradient of the surface $\nabla\phi$ is the normal to the level set C, (i.e. $\mathbf{N}$), and the term $div(\frac{\nabla\phi}{|\nabla\phi|})$ is its curvature κ. The image based speed term has values that are close to zero in regions of high intensity gradient and values that are close to unity in regions of constant intensity. Caselles *et al.* proved the existence and uniqueness of solutions of this PDE in the viscosity sense for bounded Lipschitz continuous initial data.

Appendix B

Classification Algorithm

In Chapter 4 we claimed that the classification algorithm is sufficient to unambiguously label a grid vertex point v as burned or not-burned. To prove this claim, we must show that the algorithm uniquely determines whether v lies inside or outside of polygon Q (using Definitions 4.5 and 4.6) for each of the 16 possible cases. We note that if $p1n$ or $p2n$ lie exactly on line $L1$ or $L2$, then they will be classified as being on the inside half-space of $L1$ or $L2$, respectively. We present a simple, geometric proof for cases 1-6 and 12-15 next (the remaining cases follow by symmetry):

Proof.

- Case 1: Since $p1n, p2n \in S2$, they lie to the right of $L2$ and therefore edge $\overline{p1np2n}$ cannot cut $S3$. Point $p1n$ also lies to the left of $L1$ and $p1$ is on $L1$ therefore (all points of) edge $\overline{p1p1n}$ must lie to the left of $L1$. A similar argument proves that edge $\overline{p2p2n}$ lies to the right of $L2$ and therefore cannot cut $S3$. Since $p1n$ lies to the right of $L2$ then $\overline{p1p1n}$ must cut $S3$ into two, 2D open sets. Therefore, a ray cast from v to the left, parallel to $\overline{p1p2}$ passes through $S3$ and intersects only one edge $\overline{p1p1n}$ of polygon Q.

- Case 2: Using arguments similar to those of case 1, $\overline{p1p1n}$ and $\overline{p1np2n}$ lie entirely to the left of $L1$ and therefore cannot cut $S1$. Edge $\overline{p2p2n}$ lies to the right of $L2$ and $p2n \in S2$ therefore, $\overline{p2p2n}$ cuts $S1$ into two open sets. Therefore, a ray cast from v to the right, parallel to $\overline{p1p2}$ passes through $S1$ and intersects only one edge $\overline{p2p2n}$ of Q.

- Case 3, 4, 5, 6, 12: Edge $\overline{p2p2n}$ lies entirely to the left of $L2$ and $\overline{p1p1n}$ lies either to the left of $L1$, $L2$ or both. Therefore, no edge of Q cuts $S1$ and a ray cast from v to the right, parallel to $\overline{p1p2}$, intersects zero edges of Q.

- Case 13: Point $p1n \in S3$, therfore $\overline{p1p1n} \in S3$. Similarly, $\overline{p2p2n} \in S1$. Therefore, $\overline{p1np2n}$ must cut either $S2$ or $S0$ into two open sets. A ray cast from v along $L1$ and away from $p1$ intersects only one edge $\overline{p1np2n}$ if $\overline{p1np2n}$ cuts $S2$, or intersects zero edges if $\overline{p1np2n}$ cuts $S0$.

- Case 14: Similar to case 13 with points $p1n$ and $p2n$ reversed.

- Case 15: Edge $\overline{p1p1n}$ and $\overline{p2p2n}$ lie entirely to the left of $L1$ and cannot cut $S1$. Since $p1n \in S2$ and $p2n \in S0$ then $\overline{p1np2n}$ must cut either $S1$ or $S3$ into two open sets. A ray cast from v to the right and parallel to $\overline{p1p2}$ intersects only one edge $\overline{p1np2n}$ if $\overline{p1np2n}$ cuts $S1$, or intersects zero edges if $\overline{p1np2n}$ cuts $S3$.

Appendix C

Superquadric Cells and Cell Fitting

A variation of the classic ellipse function is the superellipse which can be defined by the implicit equation $f(x, y) = 0$ with

$$f(x, y) = \left|\frac{x}{a_0 a_1}\right|^{\epsilon} + \left|\frac{y}{a_0 a_2}\right|^{\epsilon} - 1, \tag{C.1}$$

where ϵ controls the roundness of the shape, a_0 is a global scaling factor, and a_1 and a_2 are horizontal and vertical scaling factors, respectively. For $\epsilon = 2$ the superellipse coincides with the ellipse, as ϵ increases more and more beyond 2, it bulges outward. For very large values of ϵ, it becomes boxlike. Its "sides" are nearly straight and its "corners" rather sharp. On the other hand, as ϵ decreases below 2, there is less and less bulge outward, so that at $\epsilon = 1$ the side are straight lines. For $\epsilon < 1$, it bulges inward with a look of being pinched.

The superellipse also has a parametric representation $\mathbf{e}(u) = (e_1(u), e_2(u))^{\mathsf{T}}$, where

$$\mathbf{e} = a_0 \begin{pmatrix} a_1 C_u^{\epsilon} \\ a_2 S_u^{\epsilon} \end{pmatrix}, \tag{C.2}$$

and $-\pi \leq u \leq \pi$, and where $S_w^{\epsilon} = \operatorname{sgn}(\sin w)|\sin w|^{\epsilon}$ and $C_w^{\epsilon} = \operatorname{sgn}(\cos w)|\cos w|^{\epsilon}$ respectively.

In 3D, the implicit form of a superellipsoid is

$$f(x, y) = \left(\left(\left(\frac{x}{a_0 a_1}\right)^{\frac{2}{\epsilon_1}} + \left(\frac{y}{a_0 a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_0 a_3}\right)^{\frac{2}{\epsilon_1}}\right)^{\frac{\epsilon_1}{2}} - 1, \tag{C.3}$$

and the parametric equation $\mathbf{e}(u, v) = (e_1(u, v), e_2(u, v), e_3(u, v))^\top$ is defined as

$$\mathbf{e} = a_0 \begin{pmatrix} a_1 C_u^{\epsilon_1} C_v^{\epsilon_2} \\ a_2 C_u^{\epsilon_1} S_v^{\epsilon_2} \\ a_3 S_u^{\epsilon_1} \end{pmatrix}. \tag{C.4}$$

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v < \pi$, and where $S_w^\epsilon = \mathrm{sgn}(\sin w)|\sin w|^\epsilon$ and $C_w^\epsilon = \mathrm{sgn}(\cos w)|\cos w|^\epsilon$ respectively. Here, $a_0 \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$, are aspect ratio parameters, and $\epsilon_1, \epsilon_2 \geq 0$ are "squareness" parameters.

C.1 Dynamic Deformable Superquadric Cells

Following Metaxas and Terzopoulos [102, 68], the positions of points on a superquadric cell at time t, relative to an inertial frame of reference in space are given by

$$\mathbf{x}(\mathbf{u}, t) = \mathbf{c}(t) + \mathbf{R}(t)\mathbf{p}(\mathbf{u}, t), \tag{C.5}$$

where $\mathbf{c}$ is the origin of a cell-centered reference frame, and $\mathbf{R}$ is a rotation matrix that gives the orientation of the cell-centered frame relative to the inertial frame. In the cell-centered frame, the positions of cell points $\mathbf{p}$ is the sum of a reference shape $\mathbf{s}$ and a local displacement $\mathbf{d}$:

$$\mathbf{p}(\mathbf{u}, t) = \mathbf{s}(\mathbf{u}, t) + \mathbf{d}(\mathbf{u}, t). \tag{C.6}$$

The local displacements allow the representation of fine shape detail while the reference shape $\mathbf{s}$ captures the coarse shape of an object. The reference shape is defined as

$$\mathbf{s} = \mathbf{T}(\mathbf{e}(\mathbf{u}; a_0, a_1, \ldots); b_0, b_1, \ldots). \tag{C.7}$$

where $\mathbf{e}$ is the parametric superellipsoid described previously. The geometric primitive $\mathbf{e}$ is subjected to the global deformation $\mathbf{T}$ (such as bending and tapering) which depends on the deformation parameters b_i. Although nonlinear, $\mathbf{e}$ and $\mathbf{T}$ are assumed differentiable so that the Jacobian of $\mathbf{s}$ may be computed. $\mathbf{T}$ may be a composite sequence of primitive deformation functions $\mathbf{T}(\mathbf{e}) = \mathbf{T}_1(\mathbf{T}_2(\ldots \mathbf{T}_n(\mathbf{e})))$. The global deformation parameters are concatenated into a vector $\mathbf{q}_s = (a_0, a_1, \ldots, b_0, b_1, \ldots)^\top$. The displacment $\mathbf{d}$ is expressed as a linear combination of basis functions $\mathbf{d} = \mathbf{S}\mathbf{q}_d$, where $\mathbf{S}$ is a shape matrix whose entries are the shape functions and $\mathbf{q_d} = (\ldots, \mathbf{d}_i^T, \ldots)^T$ is a vector of local deformation parameters (see [102, 68] for details).

The dynamics framework detailed in [102, 68] is used to fit a superquadric cell to data. The framework requires the computation of the Jacobian $\mathbf{J} = \frac{\partial \mathbf{s}}{\partial \mathbf{q_s}}$. A vector of generalized superquadric cell coordinates $\mathbf{q} = (\mathbf{q_c}^T, \mathbf{q}_\theta^T, \mathbf{q_s}^T, \mathbf{q_d}^T)^T$, where $\mathbf{q_c} = \mathbf{c}(t)$, $\mathbf{q}_\theta$ is the quaternion used to specify $\mathbf{R}(t)$, is integrated into the dynamics framework which computes an updated set of the coordinates such that the cell more closely represents the data. The framework combines the applied forces described in 4.2.2 acting on the cell degrees of freedom $\mathbf{q_d}$ with internal smoothness forces acting on these variables. A simple Euler integration method is used to implement dynamic cell fitting process. We also scale or warp the material coordinates $\mathbf{u}$ of a cell to create a more even distribution of cell nodes (which improves cell fitting) [26].

C.2 Point-to-point Cell Constraints

Point-to-point constraints between two superquadric cells are implemented using a technique introduced by [35] known as "displacement constraints". This approach uncouples the constraints connecting two cells from the effects of the external forces. A cell is fitted to the data dynamically as described in the previous section for a specified number of iterations. This step is then followed by a step that attempts to satisfy the geometric constraints between two cells, by iteratively adjusting the position of each cell until they are within a preset tolerance. This technique is simpler and more efficient than coupled dynamic systems.

C.3 Cell Blending

The implicit functions of the superquadric cells can be blended to form a smooth transition between their bounding surfaces. We use a blending function $b : \Re^k \rightarrow \Re$ known as the superelliptic blend [110] is defined by

$$\left[\sum_{i=0}^{k}(f_i + 1)^p\right]^{\frac{1}{p}} - 1. \tag{C.8}$$

This function generates a smooth isosurface that closely bounds the superquadric cells. As the blend parameter p is increased, the isosurface wraps the cells more tightly (i.e. as $p \to \infty$, the isosurface approaches the union of the cell bounding surfaces).

Bibliography

[1] E.L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods.* Springer–Verlag, Berlin, Heidelberg, 1990.

[2] A.A. Amini, T.E. Weymouth, and R.C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[3] N. Ayache, editor. *Proc. First International Conf. on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), Nice, France, April, 1995*, volume 905 of *Lectures Notes in Computer Science*, Berlin, Germany, 1995. Springer–Verlag.

[4] D. Ballard and C. Brown. *Computer Vision.* Prentice–Hall, Englewood Cliffs, NJ, 1982.

[5] E. Bardinet, L.D. Cohen, and N. Ayache. Superquadrics and free-form deformations: A global model to fit and track 3D medical data. In Ayache [3], pages 319–326.

[6] E. Bardinet, L.D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding*, 1996. In press. Also research report 2617, INRIA, Sophia-Antipolis, France.

[7] M.O. Berger. Snake growing. In O. Faugeras, editor, *Computer Vision – Proc. First European Conf. on Computer Vision (ECCV'90), Antibes, France, April, 1990*, Lectures Notes in Computer Science, pages 570–572. Springer–Verlag, 1990.

[8] M.O. Berger and R. Mohr. Towards autonomy in active contour models. In *Proceedings 10th International Conference on Pattern Recognition, Atlantic City, NJ*, pages 847–851, Los Alamitos, CA, 1990. IEEE Computer Society Press.

[9] Y. Bizais, C. Barillot, and R. Di Paola, editors. *Information Processing in Medical Imaging: Proc. 14th Int. Conf. (IPMI'95), Ile de Berder, France, June, 1995*, volume 3 of *Computational Imaging and Vision*, Dordrecht, The Netherlands, 1995. Kluwer Academic.

[10] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[11] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–287, 1973.

[12] I. Carlbom, D. Terzopoulos, and K. Harris. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. *IEEE Trans. on Medical Imaging*, 13(2):351–362, 1994.

[13] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66, 1993.

[14] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. Fifth International Conf. on Computer Vision (ICCV'95), Cambridge, MA, June, 1995*, pages 694–699, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[15] A. Chakraborty and J.S. Duncan. Integration of boundary finding and region-based segmentation using game theory. In Bizais et al. [9], pages 189–200.

[16] A. Chakraborty, L.H. Staib, and J.S. Duncan. Deformable boundary finding influenced by region homogeneity. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA, June, 1994*, pages 624–627, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[17] L.W. Chang, H.W. Chen, and J.R. Ho. Reconstruction of 3D medical images: A nonlinear interpolation technique for reconstruction of 3D medical images. *Computer Vision, Graphics, and Image Processing*, 53(4):382–391, July 1991.

[18] C.W. Chen, T.S. Huang, and M. Arrott. Modeling, analysis and visualization of left ventricle shape and motion by hierarchical decomposition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:342–356, April 1994.

[19] A.J. Chorin. Flame Advection and Propagation Algorithms. *Journal of Computational Physics*, 35:1–11, 1980.

[20] I. Cohen, L.D. Cohen, and N. Ayache. Using deformable surfaces to segment 3D images and infer differential structures. *CVGIP: Image Understanding*, 56(2):242–263, 1992.

[21] L.D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[22] L.D. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, November 1993.

[23] T. Cootes, A. Hill, C. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–366, July 1994.

[24] C. Davatzikos and R.N. Bryan. Using a deformable surface model to obtain a mathematical representation of the cortex. In *International Symp. on Computer Vision, Coral Gables, FL, November, 1995*, pages 212–217, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[25] C.A. Davatzikos and J.L. Prince. An active contour model for mapping the cortex. *IEEE Trans. on Medical Imaging*, 14(1):65–80, March 1995.

[26] D. DeCarlo and D. Metaxas. Blended deformable models. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA, June, 1994*, pages 566–572, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[27] H. Delingette, M. Hebert, and K. Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–144, April 1992.

[28] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. Virtual reality and craniofacial surgery simulation. In Robb [83], pages 607–618.

[29] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proc. SIGGRAPH'95, Los Angeles, CA, August, 1995, in Computer Graphics Proc., Annual Conf. Series 1995*, pages 173–182, New York, NY, 1995. ACM SIGGRAPH.

[30] I. Essa, S. Sclaroff, and A.P. Pentland. Physically-based modeling for graphics and vision. In R. Martin, editor, *Directions in Geometric Computing*. Information Geometers, U.K., 1993.

[31] G. Farin. *Curves and Surfaces for CAGD*. Academic Press, New York, NY, 1993.

[32] J.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture*. Halstead Press, Horwood, NY, 1981.

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

[34] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 23:69–96, 1986.

[35] J.D. Gascuel and M.P. Gascuel. Displacement constraints: A new method for interactive dynamic animation of articulated bodies. In *Proc. EUROGRAPHICS'92*, 1992.

[36] J.M. Gauch, H.H. Pien, and J. Shah. Hybrid boundary-based and region-based deformable models for biomedical image segmentation. In *Mathematical Methods in Medical Imaging III*, volume 2299 of *SPIE Proc.*, pages 72–83, San Diego, CA, 1994. SPIE.

[37] B. Geiger. Three dimensional simulation of delivery for cephalopelvic disproportion. In *First International Workshop on Mechatronics in Medicine and Surgery, Costa del Sol, Spain, October, 1992*, pages 146–152, October 1992.

[38] A. Gourdon. Simplification of irregular surface meshes in 3D medical images. In Ayache [3], pages 413–419.

[39] M. Grayson. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 26:285–314, 1987.

[40] R.P. Grzeszczuk and D.N. Levin. Brownian strings: Segmenting images with stochastically deformable contours. In Robb [83], pages 72–89.

[41] A. Gupta, T. O'Donnell, and A. Singh. Segmentation and tracking of cine cardiac MR and CT images using a 3-D deformable model. In *Proc. IEEE Conf. on Computers in Cardiology, September, 1994*, 1994.

[42] I.L. Herlin, C. Nguyen, and C. Graffigne. A deformable region model using stochastic processes applied to echocardiographic images. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'92), Urbana, IL, June, 1992*, pages 534–539, Los Alamitos, CA, 1992. IEEE Computer Society Press.

[43] A. Hill, A. Thornham, and C.J. Taylor. Model-based interpretation of 3D medical images. In *Proc. 4th British Machine Vision Conf. (BMVC'93), Surrey, UK, September, 1993*, pages 339–348. BMVA Press, 1993.

[44] G.E. Hinton, C.K. Williams, and M.D. Revow. Adaptive elastic models for hand-printed character recognition. In J.E. Moody, S.J. Hanson, and R.P. Lippman, editors, *Advances in Neural Information Processing Systems*, San Mateo, CA, 1992. Morgan Kauffman.

[45] J. Hoschek. Offset curves in the plane. *Computer Aided Design*, 17:77–82, 1996.

[46] J. Ivins and John Porrill. Statistical snakes: Active region models. In *Proc. 5th British Machine Vision Conf. (BMVC'94)*, pages 377–386. BMVA Press, 1994.

[47] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[48] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker. Toward a computational theory of shape: An overview. In O. Faugeras, editor, *Computer Vision – Proc. First European Conf. on Computer Vision (ECCV'90), Antibes, France, April, 1990*, Lectures Notes in Computer Science. Springer–Verlag, 1990.

[49] V.A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46:141–161, 1989.

[50] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proc. SIGGRAPH'95, Los Angeles, CA, August, 1995, in Computer Graphics Proc., Annual Conf. Series 1995*, pages 55–62, New York, NY, 1995. ACM SIGGRAPH.

[51] F. Leitner and P. Cinquin. Complex topology 3D objects segmentation. In *Model-Based Vision Development and Tools*, volume 1609 of *SPIE Proc.*, pages 16–26, Bellingham, WA, 1991. SPIE.

[52] F. Leitner and P. Cinquin. From splines and snakes to Snakes Splines. In C. Laugier, editor, *Geometric Reasoning: From Perception to Action*, volume 708 of *Lectures Notes in Computer Science*, pages 264–281. Springer–Verlag, 1993.

[53] F. Leymarie and M. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):635–646, 1993.

[54] W.C. Lin and S.Y. Chen. A new surface interpolation technique for reconstructing 3D objects from serial cross-sections. *Computer Vision, Graphics, and Image Processing*, 48:124–143, Oct. 1989.

[55] P. Lipson, A.L. Yuille, D. O'Keefe, J. Cavanaugh, J. Taaffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In O. Faugeras, editor, *Computer Vision - Proc. First European Conf. on Computer Vision (ECCV'90), Antibes, France, April, 1990*, Lectures Notes in Computer Science, pages 477–484. Springer–Verlag, 1990.

[56] S. Lobregt and M.A. Viergever. A discrete dynamic contour model. *IEEE Trans. on Medical Imaging*, 14(1):12–24, March 1995.

[57] W.E. Lorenson and H.E. Cline. Marching Cubes. A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[58] R. Malladi, R. Kimmel, D. Adalsteinsson, G. Sapiro, V. Caselles, and J.A. Sethian. A geometric approach to segmentation and analysis of 3D medical images. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, San Francisco, CA, June, 1996*, pages 244–252, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[59] R. Malladi, J. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158–175, Feb. 1995.

[60] J.F. Mangin, F. Tupin, V. Frouin, I. Bloch, R. Rougetet, J. Regis, and J. Lopez-Krahe. Deformable topological models for segmentation of 3D medical images. In Bizais et al. [9], pages 153–164.

[61] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, January 1995.

[62] T. McInerney and D. Terzopoulos. Medical image segmentation using topologically adaptable snakes. In Ayache [3], pages 92–101.

[63] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. Fifth International Conf. on Computer Vision (ICCV'95), Cambridge, MA, June, 1995*, pages 840–845, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[64] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2), 1996.

[65] T. McInerney and D. Terzopoulos. Constraints on topologically adaptable deformable models. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'97), Puerto Rico, June, 1997*, 1997. Under review.

[66] T. McInerney and D. Terzopoulos. Medical image segmentation using topologically adaptable deformable surface models. In *Second International Conf. on Computer Vision, Virtual Reality and Robotics in Medicine, Grenoble, France, March*, 1997.

[67] D. Metaxas and I. Kakadiaris. Elastically adaptive deformable models. In *Computer Vision - Proc. Fourth European Conf. on Computer Vision (ECCV'94), May, 1994*, volume II, pages 550–559, 1996.

[68] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[69] J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, and M.J. Wozny. Geometrically deformed models: A method for extracting closed geometric models from volume data. In *Computer Graphics (Proc. SIGGRAPH'91 Conf., Las Vegas, NV, July, 1991)*, volume 25(4), pages 217–226, July 1991.

[70] E.N. Mortensen and W.A. Barrett. Intelligent scissors for image composition. In *Proc. SIGGRAPH'95, Los Angeles, CA, August, 1995, in Computer Graphics Proc., Annual Conf. Series 1995*, pages 191–198, New York, NY, 1995. ACM SIGGRAPH.

[71] J.R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA, 1984.

[72] C. Nastar and N. Ayache. Fast segmentation, tracking, and analysis of deformable objects. In *Proc. Fourth International Conf. on Computer Vision (ICCV'93), Berlin, Germany, May, 1993*, pages 275–279, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[73] W. Neuenschwander, P. Fua, G. Székely, and O. Kubler. Initializing snakes. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA, June, 1994*, pages 658–663, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[74] S. Osher and J.A. Sethian. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton–Jacobi Formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[75] T. Pavlidis and YT Liow. Integrating region growing and edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(3):225–233, 1990.

[76] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):730–742, July 1991.

[77] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modelling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[78] S. Pieper, J. Rosen, and D. Zeltzer. Interactive graphics for plastic surgery: A task-level analysis and implementation. In *Proc. ACM 1992 Symposium on Interactive 3D Graphics*, pages 127–134, 1992.

[79] T. Poggio, V. Torre, and C. Koch. Computational Vision and Regularization Theory. *Nature*, 317(6035):314–319, 1985.

[80] C. S. Poon, M. Braun, R. Fahrig, A. Ginige, and A. Dorrell. Segmentation of medical images using an active contour model incorporating region-based images features. In Robb [83], pages 90–97.

[81] W.K. Pratt. *Digital Image Processing, 2nd edition.* Wiley, Washington,DC, 1991.

[82] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[83] R.A. Robb, editor. *Proc. Third Conf. on Visualization in Biomedical Computing (VBC'94), Rochester, MN, October, 1994*, volume 2359 of *SPIE Proc.*, Bellingham, WA, 1994. SPIE.

[84] N. Rougon and F. Prêteux. Deformable markers: Mathematical morphology for active contour models control. In *Image Algebra and Morphological Image Processing II*, volume 1568 of *SPIE Proc.*, pages 78–89, Bellingham, WA, 1991. SPIE.

[85] N. Rougon and F. Prêteux. Directional adaptive deformable models for segmentation with application to 2D and 3D medical images. In *Medical Imaging 93: Image Processing*, volume 1898 of *SPIE Proc.*, pages 193–207, Bellingham, WA, 1993. SPIE.

[86] R. Samadani. Adaptive snakes: control of damping and material parameters. In *Geometric Methods in Computer Vision*, volume 1570 of *SPIE Proc.*, pages 202–213, Bellingham, WA, 1991. SPIE.

[87] A.R. Sanderson, D.L. Parker, and T.C. Henderson. Simultaneous segmentation of mr and x-ray angiograms for visualization of cerebral vascular anatomy. In *Int. Conf. Volume Image Processing (VIP'93), Utrecht, Netherlands*, 1993.

[88] G. Sapiro, R. Kimmel, and V. Caselles. Object detection and measurements in medical images via geodesic deformable contours. In *Vision Geometry IV*, volume 2573 of *SPIE Proc.*, pages 366–378, Bellingham, WA, 1995. SPIE.

[89] G. Sapiro, R. Kimmel, D. Shaked, B.B. Kimia, and A.M. Bruckstein. Implementing continuous-scale morphology via curve evolution. *Pattern Recognition*, 26(9):1363–1372, 1993.

[90] J. A. Sethian. Curvature and the Evolution of Fronts. *Commun. in Mathematical Physics*, 101:487–499, 1985.

[91] A. Singh, L. von Kurowski, and M.Y. Chiu. Cardiac MR image segmentation using deformable models. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905 of *SPIE Proc.*, pages 8–28, Bellingham, WA, 1993. SPIE.

[92] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, November 1992.

[93] L.H. Staib and J.S. Duncan. Deformable Fourier models for surface finding in 3D images. In R.A. Robb, editor, *Proc. Second Conf. on Visualization in Biomedical Computing (VBC'92), Chapel Hill, NC, October, 1992*, volume 1808 of *SPIE Proc.*, pages 90–104, Bellingham, WA, 1992. SPIE.

[94] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.

[95] G. Székely, A. Kelemen, Ch. Brechbuhler, and G. Gerig. Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier surface models. *Medical Image Analysis*, 1(1), 1996.

[96] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'93), New York, NY, June, 1993*, pages 82–87, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[97] H. Tek and B. Kimia. Shock-based reaction-diffusion bubbles for image segmentation. In Ayache [3], pages 434–438.

[98] D. Terzopoulos. On matching deformable models to images. Technical Report 60, Schlumberger Palo Alto Research, 1986. Reprinted in *Topical Meeting on Machine Vision*, Technical Digest Series, Vol. 12 (Optical Society of America, Washington, DC) 1987, 160-167.

[99] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[100] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.

[101] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4(6):306–331, 1988.

[102] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.

[103] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Computer Graphics (Proc. SIGGRAPH'87 Conf., Anaheim, CA, July, 1987)*, volume 21, pages 205–214, 1987.

[104] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, 1994.

[105] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[106] A.N. Tikhonov and V.A. Arsenin. *Solutions of Ill-posed Problems*. Winston and Sons, Washington,DC, 1977.

[107] G. Turk. Re-tiling polygonal surfaces. In *Computer Graphics (Proc. SIGGRAPH'92 Conf., Chicago, IL, July, 1992)*, volume 26(2), pages 55–64. ACM SIGGRAPH, July 1992.

[108] N. Ueda and K. Mase. Tracking moving contours using energy-minimizing elastic contour models. In G. Sandini, editor, *Computer Vision – Proc. Second European Conf. on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy, May, 1992*, Lectures Notes in Computer Science, pages 453–457. Springer–Verlag, 1992.

[109] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: Irregular triangulation, discontinuities and hierarchical subdivision. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'92), Urbana, IL, June, 1992*, pages 829–832, Los Alamitos, CA, 1992. IEEE Computer Society Press.

[110] L. Velho. *Piecewise Descriptions of Implicit Surfaces and Solids.* PhD thesis. Department of Computer Science. University of Toronto. Toronto. ON. Canada. 1994.

[111] B.C. Vemuri and A. Radisavljevic. Multiresolution stochastic hybrid shape models with fractal priors. *ACM Trans. on Graphics*. 13(2):177–207. April 1994.

[112] B.C. Vemuri. A. Radisavljevic. and C. Leonard. Multiresolution 3D stochastic shape models for image segmentation. In A.C.F. Colchester and D.J. Hawkes. editors. *Information Processing in Medical Imaging: Proc. 13th Int. Conf. (IPMI'93). Flagstaff. AZ. June. 1993*. Lectures Notes in Computer Science. pages 62–76. Springer-Verlag. 1993.

[113] G. Wahba. *Spline Models for Observational Data.* SIAM. Philadelphia.PA. 1990.

[114] K. Waters. A physical model of facial tissue and muscle articulation derived from computer tomography data. In R.A. Robb. editor. *Proc. Second Conf. on Visualization in Biomedical Computing (VBC'92). Chapel Hill. NC. October. 1992*. volume 1808 of *SPIE Proc.*. pages 574–583. Bellingham. WA. 1992. SPIE.

[115] R. Whitaker. Volumetric deformable models. In Robb [83].

[116] B. Widrow. The rubber mask technique. part I. *Pattern Recognition*. 5(3):175–211. 1973.

[117] D.J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*. 55:14–26. March 1992.

[118] S.A. Wood. J.D. Hoford. E.A. Hoffman. E. Zerhouni. and W. Mitzner. Method for measurement of cross sectional area. segment length. and branching angle of airway tree structures in situ. *Computerized Medical Imaging and Graphics*. 19(1):145–152. January 1995.

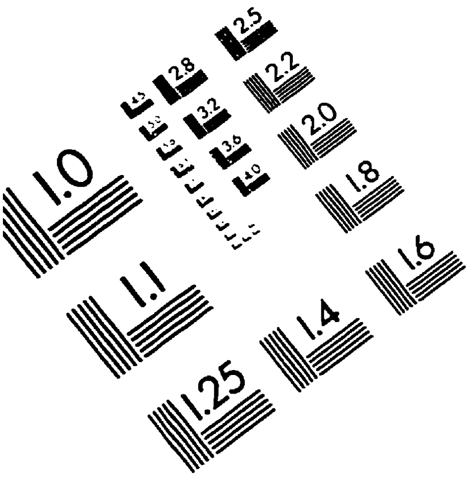
[119] M. Worring. A.W.M. Smeulders. L.H. Staib. and J.S. Duncan. Parameterized feasible boundaries in gradient vector fields. In A.C.F. Colchester and D.J. Hawkes. editors. *Information Processing in Medical Imaging: Proc. 13th Int. Conf. (IPMI'93).*

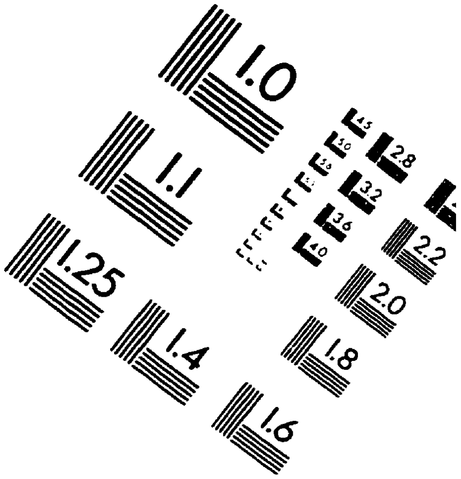
Flagstaff, AZ, June, 1993, Lectures Notes in Computer Science, pages 48–61. Springer–Verlag, 1993.

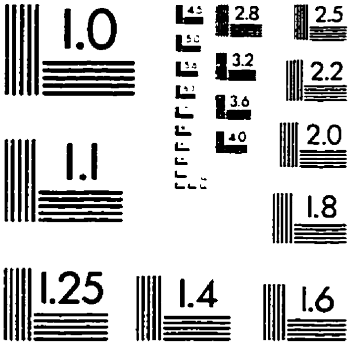
[120] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8:99–111, 1992.

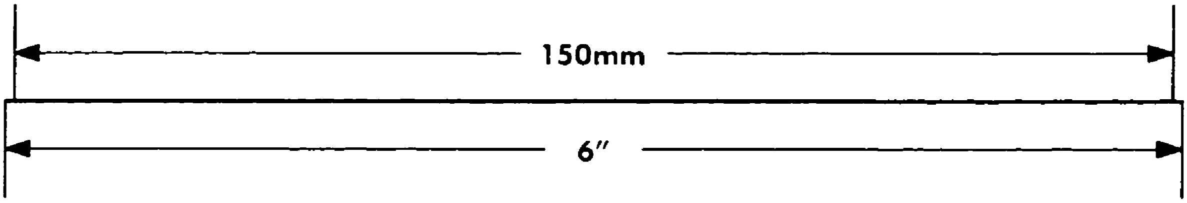
[121] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. McGraw-Hill, New York, NY, 1989.

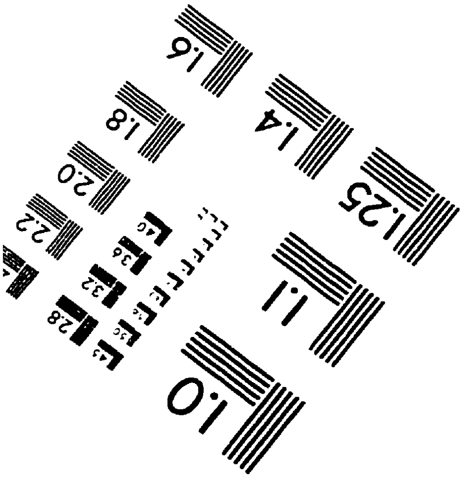
IMAGE EVALUATION
TEST TARGET (QA-3)

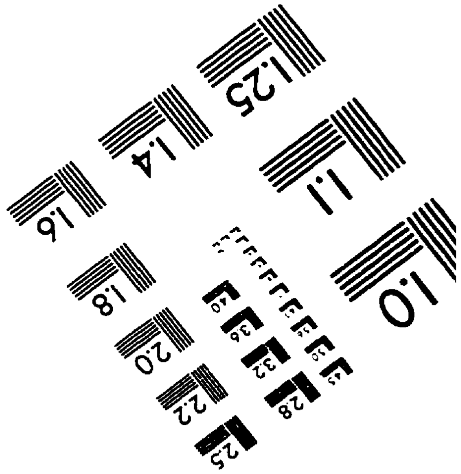












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved