3D object segmentation using deformable models

Chen, Xujian

2007

Chen, X. (2007). 3D object segmentation using deformable models. Doctoral thesis,
Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/39152>

<https://doi.org/10.32657/10356/39152>

3D Object Segmentation Using Deformable Models

Chen Xujian

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

Acknowledgements

My time as a graduate student has been the best period of my life so far and for that wonderful experience, I owe many thanks to the following people.

I had an excellent supervisor, Associate Professor Teoh Eam Khwang. I would like to express my sincere gratitude and appreciation to my supervisor for his enthusiastic encouragement, excellent guidance and steadfast support throughout the duration of this research work. It amazes me how much I have learned from him over the past three years or so. Without his guidance, this research work would not be possible.

I would like to express my appreciation to my parents, Runze Chen and Lihua Feng, my brother, Shudong Chen and my girl friend, Wenting Sun. They had given me a lot of encouragement since I started my study in Singapore in 2002. Every time when I meet difficulty, I know they are the most steadfast support in the world for me.

I would also like to thank all staff members and friends in Intelligent Machines Research Lab at the School of Electrical and Electronic Engineering, Nanyang Technological University for their kind support and help.

People at Singapore have always been warm and kind. It was very pleased to live in this warm, beautiful garden city in the past three years, particular thanks to my friends in Singapore.

Finally, I would like to acknowledge the School of Electrical and Electronic Engineering, Nanyang Technological University, for providing the award of a scholarship and the research facilities for my work in NTU.

Summary

This thesis presents research work on deformable surface model for 3D object segmentation. Over the past decades, there have been many research activities in 3D object segmentation using 3D and 2D deformable models. Full 3D methods will produce much better results than those obtained based on the 2D ones. Contextual intensity information of one voxel in one direction will be lost in each 2D image. Furthermore, a post-processing step is required to connect the sequence of 2D contours in a continuous surface. Reconstruction of the surface will be difficult if the topology of 2D contours is complicated. Therefore, it is desired to detect objects in 3D space directly to avoid the shortcomings of 2D methods, especially in application to 3D medical image analysis.

In this thesis, novel methods for 3D object segmentation are proposed. These methods are applied to the analysis of medical images. Firstly, a novel B-Surface algorithm is proposed for the segmentation of single smooth 3D objects. This algorithm works in 3D space directly, overcoming the difficulty that arises from analyzing 3D volume images slice by slice. After a coarse boundary of the target is obtained using a 3D edge extraction method, an improved 3D external force field combined with the normalized Gradient Vector Field (GVF) is computed. After the initialization of a surface model near the target, the proposed B-Surface algorithm starts to locate the boundary of the object by the deformation process. Experimental results show that B-Surface can segment a smooth 3D object with high accuracy. This method also facilitates the calculation of the geometric information of the target. It has the ability to achieve a high compression ratio by presenting the whole surface with only a relatively small number of control points. Compared with other deformable

models, the B-Surface algorithm has attractive properties such as spatial uniqueness, boundedness and continuity, and local shape controllability. The computational cost is reduced because internal forces are not required in the B-Surface algorithm. This is because a B-spline surface has a strong implicit constraint which ensures that a smooth surface is obtained. However, it will cause the difficulty of segmenting and representing the object with subtle structures such as amygdale/hippocampus or superior temporal gyrus.

Next, a Growing Deformable Surface Patches (GDSP) model is proposed for the segmentation of complicated 3D objects. A growing mechanism is introduced to achieve topologically adaptable surface extraction. Only a surface patch is initialized in the first stage. It is deformed to reach and stop at the boundary of the object and anchor there. A new surface patch is then initialized based on the existing anchored patch for deformation in the next iteration. This process is repeated until a closed surface of the subject is obtained. As a general deformable surface model, it has been applied to the segmentation of complicated objects such as the segmentation of brain ventricle and human vertebra in 3D MR images without building a template in advance. Experimental results demonstrate that this method can segment complicated objects successfully in real applications. Compared with the existing topologically adaptable deformable surfaces, the computational cost is reduced significantly by the introduction of the growing mechanism. This method is explicit and therefore, it is convenient to provide geometrical information such as area, volume, or local curvature of the object. Furthermore, surface curvature adaptation is achieved in the proposed deformable model by associating the surface curvature with the size of the surface patches.

Finally, a robust GDSP model is proposed. Normally, deformable models are sensitive to the initialization. The deformation procedure may not work properly when the initialized surface patch is parallel to the external force vector in 3D space. In the proposed method, a novel internal force is introduced to overcome the sensitivity of the position of initialization. This internal force could rotate the deforming surface

patch and force it to move perpendicularly to the external force vector. Therefore, under the influence of internal forces and external forces, a surface patch can extract the boundary accurately even if the initialization is bad. Detailed analysis on the performance of robust GDSP is given in the thesis. Furthermore, a hierarchical mechanism is introduced in the robust GDSP model in order to speed up the algorithm and reduce local minima.

An efficient algorithm for surface area calculation of objects of interest is developed as well in this thesis. The surface area of the entire model is obtained by summing the areas of all the surface patches. With the help of the novel internal force, this method can locate the boundary of the target successfully even when the first surface patch is initialized vertically to the target which is the worst case of initialization. It also improves the accuracy in segmenting objects with sharp corners. The experimental results show that computational costs are reduced because of the introduction of a hierarchical mechanism. It can calculate the surface area of the target efficiently. The experimental results show that the proposed method achieves high accuracy in surface area calculation.

In summary, research work on 3D object segmentation using deformable surface models is presented in this thesis. A B-Surface model is proposed for simple 3D object segmentation. GDSP model is introduced to generate a topologically adaptable deformable model. Finally, a robust GDSP model is presented to overcome the sensitivity to initialization of deformable models. All the proposed algorithms have been applied to the segmentation of 3D medical images. Promising experimental results are obtained. The experimental results of B-Surface model show that the proposed method can locate and represent the boundary of smooth 3D objects with high accuracy and data compression ratio. The experimental results of GDSP show that it overcomes the limitations of traditional explicit deformable models and achieves topologically adaptable surface extraction. The experimental results of robust GDSP show that the proposed method can obtain the correct boundary of the target even if the initialized surface patch is in the worst situation.

Contents	v
----------	---

Table of Contents

Acknowledgements	i
Summary	ii
List of Figures	xvi
List of Tables	xvii
List of Symbols	xviii
1 Introduction	1
1.1 Motivation	3
1.2 Objectives	4
1.3 Original Contributions	6
1.4 Organization of the Thesis	8
2 A Literature Review	10
2.1 Introduction	10
2.2 Methods on Image Segmentation	11

Contents	vi
2.3 2D and 3D Deformable Models	14
2.3.1 Implicit Deformable Models	14
2.3.2 Explicit Deformable Models	17
2.4 On B-spline Curve and Surface	25
2.4.1 B-spline curve	25
2.4.2 Uniform Cubic B-spline Curve	28
2.4.3 Uniform Bi-Cubic B-spline Surface	31
2.4.4 Non-Uniform Rational B-spline (NURBS)	36
2.5 B-spline Based Deformable Models	38
2.6 Statistical Deformable Models	42
2.7 Topologically Adaptable Deformable Models	49
2.8 Concluding Remarks	55
3 A Novel B-Surface Algorithm for Segmentation of Smooth 3D Ob-	
jects	57
3.1 Introduction	57
3.2 External Force Field for the B-Surface Algorithm	58
3.2.1 Extraction of the Coarse Boundary	59
3.2.2 External Force Field for the B-Surface Algorithm	62
3.3 Presentation of the B-Surface Model	64
3.4 Deformation of the B-Surface Model	66
3.5 Experimental Results and Discussion	69

Contents	vii
3.5.1 Experiments on Segmentation of Simulated Object	69
3.5.2 Experiments on Segmentation of Brain Tumor	75
3.5.3 Experiments on Segmentation of Brain Ventricle	80
3.6 Concluding Remarks	87
4 A Novel Deformable Model for Segmentation of Complicated 3D Objects	89
4.1 Introduction	89
4.2 3D Topologically Adaptable Growing Deformable Surface Patches (GDSP) Model	91
4.2.1 Representation of the Model	91
4.2.2 Topologically Adaptable Growing Mechanism	94
4.2.3 Implementation of the Growth of Surface Patches	97
4.3 Deformation of Surface Patches for Object Segmentation	101
4.3.1 Internal Forces	102
4.3.2 External Force	105
4.4 Experimental Results and Discussion	106
4.4.1 Experiments on Segmentation of Torus	106
4.4.2 Experiments on Segmentation of Human Ventricle	109
4.4.3 Experiments on Segmentation of Human Vertebra	114
4.5 Concluding Remarks	118
5 Development of a Robust Algorithm for 3D Object Segmentation	119

Contents	viii
5.1 Introduction	119
5.2 Robust Growing Deformable Surface Patches (GDSP) Model	121
5.2.1 Inclusion of a New Internal Force	121
5.2.2 Inclusion of a Hierarchical Framework	129
5.2.3 Computation of the Surface Area	132
5.3 Experimental Results and Discussion	135
5.3.1 Effect of the New Internal Force	135
5.3.2 Effect of the Hierarchical Framework	147
5.3.3 Computation of the Surface Area	153
5.4 Concluding Remarks	156
6 Conclusion and Recommendations	157
6.1 Conclusion	157
6.2 Recommendations for Future Work	160
6.2.1 B-Surface Algorithm Based on Closed Surface Model	160
6.2.2 Growing B-Surface Algorithm without Overlapping	160
6.2.3 Statistical GDSP Model	161
6.2.4 On-line Learning GDSP Model	161
Author's Publications	163
Bibliography	165

List of Figures

2.1 Two edge snakes on a pear and potato. (a) The use has pulled one of the snakes away from the edge of the pear; (b)-(d) After the user lets go, the snake snaps back to the edge of the pear. (Adopted from [1]). 19

2.2 Initial 3D reconstruction of a finger. (a) Finger stereo pair for first time instant. (b) User-initialized cylinder. (c)-(f) Initial reconstructed shape from three viewpoints. Every other grid line has been drawn on the surface. (Adopted from [2]). 21

2.3 The points where a regular grid defined was mapped through DFSA using: (a) and (b) zero iterations, (c) and (d) three iterations, and (e) and (f) nine iterations of the fixed-point algorithm. The regularity of the grid, which is equivalent to the homotheticity of the map, improves with the number of iterations. (Adopted from [3]). 22

2.4 (a) A digital volumetric brain. (b) Initial approximation of the brain by an octahedron. (c)-(g) Intermediate subdivision steps. (h) Brain shown by the final mesh in shaded form. (Adopted from [4]). 24

2.5 Segmentation results of the lateral horn of a cerebral ventricle at the m-rep level of scale (i.e., before boundary displacement) from MRI using a single figure model. (a) Lateral view. (b) Coronal view 1. (c) Coronal view 2. (d) Coronal view 3. (Adopted from [5]). 25

List of Figures	x
2.6 Uniform cubic B-spline curve.	28
2.7 Basic functions of uniform cubic B-spline curve.	30
2.8 Uniform bi-cubic B-spline surface.	32
2.9 Optimization of snakes on 2.00 mm fixed-stenosis phantom: (a) ini- tialized, (b) after five iterations of DP algorithm, (c) after ten itera- tions, (d) after 15 iterations, (e) after 20 iterations, and (f) optimum reached at 23 iterations, lines connect initial control point locations and optimized locations. (Adopted from [6]).	39
2.10 Figure illustrating the PERM strategy. (Adopted from [7]).	40
2.11 (a) The target contour. (b) Spline with nine control points (GA). (c) Spline with nine control points(CC). (d) Spline from GA which has similar averaged errors to the spline in (c). (e) Spline from (c) superimposed on the original image. (f) A graph comparing squared errors when using different numbers of control points with the two algorithms. (Adopted from [7]).	41
2.12 Lane detection results using B-Snake. (Adopted from [8]).	41
2.13 The reconstructed surface for the epicardial boundary of the left- ventricle. (Adopted from [9]).	42
2.14 Section of printed circuit board with resistor model superimposed, showing its initial position and its location after 30, 60, 90 and 120 iterations. (Adopted from [10]).	45

List of Figures	xi
2.15 Example of automated brain image segmentation and interpretation: (a) manual tracings; (b) initial average position of the shape model; (c) optimal shape model position after linear transform step; (d) op- timal shape model position after piecewise linear transform step; (e) outlier detection — marked by dark dots; (f) final point adjustment. Images contrast-enhanced to improve visibility. (Adopted from [11]). .	46
2.16 Multi-Resolution search from displaced position. (Adopted from [12]).	48
2.17 (a) Implementation of axial transformations: triangulation between the two surfaces and deletion of the two old vertices; (b) implementa- tion of annular transformations: splitting of the problematic set UVO in two parts, then separate merging. (Adopted from [13]).	50
2.18 Reconstruction of a surface model of a cup from silhouettes: (a) cup bounding volume represented as an octree, (b) triangulated surface of reconstructed model. (Adopted from [14]).	51
2.19 Morphing between a sphere and a torus (a) pinched sphere (b) stran- gled torus. (Adopted from [15]).	52
2.20 T-surface segmenting vertebra phantom from CT image volume. The dark shaded regions are frozen and have been removed from the com- putation. (Adopted from [16]).	53
2.21 Surface evolution (no Gouraud shading is shown): (a) iteration 0 on image I_0 ; (b) iteration 200 on image I_0 ; (c) iteration 400 on image I_0 ; (d) iteration 1100 on image I_0 . (Adopted from [13]).	54
2.22 Shape recovery from volumetric image data a phantom vertebral. (Adopted from [17]).	55
3.1 Result of 3D edge extraction method.	61

List of Figures	xii
3.2 To compute the external forces on surface points and calculate the influence of the external forces on control points.	67
3.3 A simulated sphere for segmentation.	70
3.4 External force field obtained using improved 3D GVF. (a) 3D external force vectors are depicted in horizontal plane; (b) 3D external force vectors are depicted in vertical plane.	71
3.5 Final experimental result of B-Surface algorithm. (a) Initial B-Surface model; (b) 6 surface patches from the all 52 are presented; (c) Final result of the sphere.	73
3.6 Original images and the coarse edges: (a1)-(a3) 3 slices of the original brain image; (b1)-(b3) the coarse edges of the tumor.	76
3.7 3D external force field of the brain tumor. (a) X-Y plane with Z value at 45; (b) Y-Z plane with X value at 45.	77
3.8 Initial B-Surface model for brain tumor segmentation.	77
3.9 Experimental result on segmentation of brain tumor.	78
3.10 Evaluation of the tumor segmentation result: (a1)-(a4) 4 slices of the original brain image; (b1)-(b4) the overlapping image of the boundary of the tumor with the original image.	79
3.11 Original MR images of human brain.	81
3.12 Edge maps of the MR image.	81
3.13 3D external force field of the brain ventricle.	82
3.14 The locally enlarged image of the 3D external force field of the brain ventricle.	83
3.15 Different view of the final result of the B-Surface in brain ventricle segmentation: (a) left view; (b) top view; (c) 3D view.	85

List of Figures	xiii
3.16 Evaluation of the result of B-Surface in brain ventricle segmentation. (a1) is the 75th slice of the original MR image; (a2) and (a3) are the 86th and 87th slices of the original MR image respectively; (b1) to (b3) are the overlapping images of the segmentation result with the original MR image.	86
4.1 Each square surface patch, S_i , consists of 16 knots: P_{11} to P_{44} . P_{11} , P_{14} , P_{41} and P_{44} are the four vertices of the square patch.	91
4.2 Triangulate a square patch by merging two adjacent vertices.	92
4.3 Triangulate a square patch by merging two vertices when the average curvature of the surface changes significantly	93
4.4 Triangulate a square patch by merging two vertices when surface patches come to a pole of the object to generate a closed surface . . .	93
4.5 Flowchart of the GDSP model.	95
4.6 Surface patch growth: type 1.	97
4.7 Surface patch growth: type 2.	98
4.8 Surface patch growth: type 3.	98
4.9 Surface patch growth: type 4.	99
4.10 Surface patch growth: type 5.	100
4.11 Surface patch growth: type 6.	100
4.12 Surface patch growth: type 7.	101
4.13 (a) θ is near 180° if the surface of the object is simple; (b) θ decreases when the surface of the object is complicated.	103
4.14 Simulated torus for segmentation.	107

4.15 Experimental result of torus segmentation without noise: (a) result with 100 surface patches; (b) result with 150 surface patches; (c) result with 200 Surface Patches; (d) result with 300 surface patches; (e) result with 350 Surface Patches; (f) result with 473 surface patches.108

4.16 Change of the number of the ASPs as the iteration number increases during the deformation procedure in torus segmentation. 110

4.17 The growing steps of the proposed deformable model in human brain ventricle segmentation: (a) result with 100 surface patches; (b) result with 200 surface patches; (c) result with 300 surface patches; (d) result with 400 surface patches; (e) result with 500 surface patches; (f) result with 600 surface patches. 111

4.18 Different views of the final result of human brain ventricle segmentation: (a) top view; (b) right view; (c) 3D view. 112

4.19 Change of the number of the ASPs as the iteration number increases during the deformation procedure in brain ventricle segmentation. . . 113

4.20 The growing steps of the proposed deformable model in human vertebra segmentation: (a) result with 200 surface patches; (b) result with 400 surface patches; (c) result with 600 surface patches; (d) result with 800 surface patches; (e) result with 900 surface patches; (f) result with 1000 surface patches. 115

4.21 Different view of the final result of human vertebra segmentation: (a) left view; (b) top view; (c) 3D view. 116

4.22 Change of the number of the ASPs as the iteration number increases during the deformation procedure in human vertebra segmentation. . 117

5.1 The traditional external and internal forces fail to let the knots converge to the actual boundary of the object. 122

- 5.2 The additional internal force rotates the surface patch if the initialized surface patch is parallel to the external force vector. 123
- 5.3 The unit vector $\mathbf{n}$ which is orthogonal to those vectors and whose sense is such that $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{n}$ form a right-handed triple of vectors, is called the unit normal vector to S at P 124
- 5.4 The unit vector $\mathbf{n}_{22}$, $\mathbf{n}_{23}$, $\mathbf{n}_{32}$ and $\mathbf{n}_{33}$ are normals to the surface patch at point P_{22} , P_{23} , P_{32} and P_{33} 126
- 5.5 α is the difference between the direction of the normal and the external force. 126
- 5.6 I_0 is the original image and the base of the pyramid. I_1 is the image of level 1 in the pyramid. I_2 is the image of level 2 in the pyramid. . . 130
- 5.7 Surface evolution without the new introduced internal force f_n in the segmentation of torus: (a) initial surface patch model; (b) surface patch at iteration 5; (c) surface patch at iteration 10; (d) surface patch at the last iteration. 137
- 5.8 Surface evolution without the new introduced internal force f_n in ring segmentation, but α_c and α_e are reduced to remove the strong opposite force against the external force: (a) initial surface patch model; (b) surface patch at iteration 4; (c) surface patch at iteration 7; (d) surface patch at the last iteration. 139
- 5.9 Surface evolution with the new introduced internal force f_n in ring segmentation: (a) initial surface patch model; (b) surface patch at iteration 3; (c) surface patch at iteration 6; (d) surface patch at iteration 8. 140
- 5.10 (a1) and (a2) are parts of the surface extraction result with f_n ; (b1) and (b2) are parts of the surface extraction result without f_n 143

List of Figures	xvi
5.11 External force field of sharp corner.	144
5.12 (a) is the ventricle detection result with f_n ; (b) is the ventricle detection result without f_n	146
5.13 A coarse boundary of the brain ventricle is obtained based on the low level surface model and external force field $E_{f_{ext}2}$	148
5.14 A coarse boundary of the brain ventricle is obtained based on the median level surface model and external force field $E_{f_{ext}1}$	149
5.15 A refined boundary of the brain ventricle is obtained based on the low level surface model and external force field $E_{f_{ext}0}$	149
5.16 A comparison of the two methods of the segmentation of brain ventricle (with and without multi-scale frame work).	150
5.17 A coarse boundary of the human vertebra is obtained based on the low level surface model and external force field $E_{f_{ext}2}$	151
5.18 A coarse boundary of the human vertebra is obtained based on the median level surface model and external force field $E_{f_{ext}1}$	152
5.19 A refined boundary of the human vertebra is obtained based on the low level surface model and external force field $E_{f_{ext}0}$	152
5.20 Two methods of human vertebra segmentation are compared on the graph (with and without multi-scale frame work).	153
5.21 The segmentation results of the simulated objects.	155

List of Tables

3.1	Comparison of B-Surface algorithm with traditional methods in simulated object segmentation.	74
3.2	Comparison of B-Surface algorithm with traditional methods in brain tumor segmentation.	80
3.3	Comparison of B-Surface algorithm with the traditional methods in brain ventricle segmentation.	87
4.1	Experimental results on segmentation of torus.	109
4.2	Experimental results on segmentation of human ventricle.	112
4.3	Experimental results on segmentation of human vertebra.	116
5.1	Experimental results of internal force in the segmentation of human vertebra.	142
5.2	Experimental results of internal force in the segmentation of brain ventricle.	145
5.3	The comparison of the theoretical area value and the area value obtained by the proposed algorithm.	156

List of Symbols

A	Surface area
B_R	Bi-cubic B-spline surface control point matrix
E	Energy
L_i	Lagrange interpolation surface control point matrix
M	Surface model
M_R	Coefficient matrix of bi-cubic B-spline
M_L	Coefficient matrix of Lagrange interpolation surface
n	Normal to surface
P	Surface point
P	Surface point matrix
Q	Control point
S	Surface patch
U	Parameter vector for surface
W	Parameter vector for surface

Chapter 1

Introduction

Over the past decades, there have been many research activities in object segmentation. By analyzing the input 2D or 3D images, a mathematical description of the object is to be derived. Boundary extraction plays a crucial role as an initial step in many computer vision applications such as tracking, motion analysis, surgery simulation, and computing the position and size of the object for pattern recognition. It can help to augment our understanding and form conclusions about various properties of objects of interest.

However, the task of object segmentation is still difficult because of the noise presented in the input image, variation of shape, complexity of the object of interest, and low contrast between the target and the background/nearby objects. For example, in the application of analyzing the image of printed circuit boards, the noise presented could be high enough to blur the actual edge of the resistors and a rigid template for detecting resistors is not suitable for their variations in shapes. How to extract the boundary of objects automatically and accurately in similar situation are still open problems.

There are many categories of methods to achieve object detection in computer vision. One category of methods is model-based object detection. First, a model of the object of interest is defined. A best fit of the model is then searched within the

input image. The second category is image invariance method. In this case, objects are detected based on a matching of a set of invariant image pattern relationships. The third category is example-based learning algorithm. This category of methods is based on the experience obtained by learning the salient features of a class from sets of labelled positive and negative examples.

By combining the geometry, approximation and statistics theories, deformable models offer a powerful approach to image analysis. They are widely used in image segmentation, object detection, matching, tracking and motion analysis, non-rigid modelling, etc.

Deformable models, which were first introduced by Kass [1], have been widely used in object detection and image segmentation. The shape of deformable models is controlled by internal continuity forces and external image forces using energy minimization. Internal forces which come from the model itself act as a smoothness constraint. External forces, which come from the input image, guide the active contour toward image features that are of interest. The features are lines, edges, and/or object boundaries.

For the past decade, there have been many research efforts for achieving 3D objects segmentation based on deformable models. Variable representations have been used to fulfill different 3D modelling needs, from deformable lines to deformable surfaces and volumes. The basic mechanism of 3D deformable models is similar to the one in 2D case. 3D models are deformed under the combined influence of the external energy which comes from the image, and the internal energy which comes from the model itself. Sometimes, other constraints are introduced to improve the performance of deformable models. The constraints are “weight” force, pressure force, etc.

1.1 Motivation

In some 3D object segmentation algorithms, 2D contours of the object are detected first by analyzing the input image slice by slice. The resulting sequence of 2D contours are then connected to form a 3D model. Segmenting 3D image slice by slice is a laborious process. Contextual intensity information of one voxel in one direction will be lost in each 2D image. Furthermore, a post-processing step is required to connect the sequence of 2D contours in a continuous surface. The reconstruction of the surface will be difficult if the topology of 2D contours is complicated. Also, the generated surface can contain inconsistencies or show rings or bands. Therefore, it is a trend to segment objects in 3D space directly to avoid the shortcomings of 2D methods, especially in applications to 3D medical image analysis.

Deformable models have mainly two formulations, i.e., explicit (parametric) form and implicit form. Each of these formulations has its advantages and disadvantages. Deformable models in implicit form are also known as level set techniques. Compared with explicit deformable models, the advantage of implicit deformable models is the topological adaptability. However, the topology of implicit deformable models is not explicit.

In the explicit form, we make a distinction between deformable surfaces having continuous and discrete representations. Deformable models in parametric form are not only compact, but also robust to both image noise and boundary gaps as it constrains the extracted boundaries to be smooth. However, using discrete deformable surfaces, we will meet difficulty in computing accurate differential quantities such as surface normals or curvatures of the surface we obtained, which are useful in high-level image analysis.

For the segmentation of complicated objects, a priori knowledge from training data could be incorporated in deformable models. It is because in medical image analysis a category of organs often have similar shapes that can be utilized to generate statistical shape models for segmentation. However, the process of constructing

statistical shape models is difficult for 3D models. More training data sets are necessary because of the high dimension (in terms of thousands) of the feature vector compared with 2D case. However, these training data sets are segmented images obtained mainly by labelling manually from experts. Therefore, it is difficult to obtain enough training data set in many real applications. Even if Point Distribution Models (PDMs) are constructed for each kind of objects that is to be segmented, it will also meet some other difficulties. In medical image analysis, if the target has a significant variation in shapes because of some disease, it is very hard to find the correct boundary for 3D statistical deformable models.

Another problem that arises in shape recovery and segmentation is the difficulty in topology transformation for parametric deformable models. Parametric deformable models are bound to their intrinsic topology. Deformable models give prior information on the shape of the objects to segment. If the topology of the initial model is different from objects of interest, the model might not work correctly.

1.2 Objectives

In this research work, one of our goals is to develop a new 3D deformable model based on bi-cubic B-spline surface for boundary extraction from 3D images. It will find the boundaries of structures accurately even under the situation that the boundaries of structures are indistinct and disconnected or even occluded. Besides accuracy and robustness, the speed of deformation process is another criterion of deformable models. The speed of computation for segmentation and representation should be as fast as possible for surgical operation or diagnosis. The way to enhance the speed of the deformable model is to find a simpler method that reduces the computation tasks in each deformation step. The more information we obtain from the detected object, the more features we could extract from the object for high-level image analysis. The differential quantities such as surface normals or curvatures of the surface are useful in image analysis. For example, the brain surface curvature is

introduced in the analysis of variability between males and females [18]. In order to obtain more information regarding the detected object for high-level image analysis, a continuous deformable model is chosen to build our 3D deformable model.

In order to achieve a 3D deformable model for complicated object segmentation, a 3D topologically adaptable deformable model is designed. Deformable surface model in the parametric form is chosen because of its advantages. Therefore, it will be convenient to provide geometrical information such as area, volume, or local curvature of the object. At the same time, it is designed to avoid the difficulty to build PDMs for each object of interest respectively. A new topology transformation strategy is proposed to avoid the high computational costs of the current topologically adaptable deformable model.

The deformation procedure may not work properly when the initialized surface patch is parallel to the direction of the external force vector in 3D space. Knots are forced to move along the direction of the external force. The knots of the surface patch will come close to each other as the influence of the external force. At the same time, the traditional internal forces just work opposite the external force. They cannot let the knots converge to the actual boundary of the object quickly. In order to remove the sensitivity of the position of the surface model initialization, an improved method is proposed. Furthermore, we also plan to speed up algorithm, reduce local minima.

1.3 Original Contributions

This thesis presents the research on 3D object segmentation and its application in medical image analysis. Its contributions can be listed as follows:

- (1) A 3D continuous deformable surface model, B-Surface algorithm, is developed for object segmentation and feature extraction in volume images. For smooth object segmentation, B-Surface has attractive properties such as spatial uniqueness, boundedness and continuity, and local shape controllability. This method also facilitates the calculation of the geometric information of the target. Compared with the traditional methods, B-Surface does not use a fixed B-spline surface model, which will constrain the method to the segmentation of a particular object corresponding to the fixed model. Furthermore, the computational cost is reduced because we need not compute the internal energy term in each deformation step. An improved 3D Gradient Vector Flow (GVF) is derived to calculate the external force field for B-Surface algorithm. The magnitude of the external force does not decrease as the distance increases. When a space point is far away from the boundaries of the object, the magnitude of the external force at that point will be strong enough to guide the surface model to the correct boundaries. The experimental results show that the proposed B-Surface algorithm can segment a smooth object with high accuracy and high data compression ratio without the help of internal force.
- (2) A novel 3D Growing Deformable Surface Patches (GDSP) model is proposed for complicated object segmentation. In the proposed method, a novel growing mechanism is introduced for topologically adaptable surface extraction. At the first stage, only one surface patch is initialized and deformed. Next, new surface patches will be generated from the deformed one and then be deformed. Finally, a closed surface model is obtained based the initial surface patch. In the proposed methods, the process of detecting and solving topological trans-

formations is simplified. We only detect the connecting probability between new patch and those surface patches with bare edges (active patches). Thus, the process of detecting and solving topological transformations is simplified significantly compared with the current topologically adaptable deformable models. The experimental results show that the proposed topologically adaptable method can segment complicated objects with high accuracy.

- (3) The deformation procedure may not work properly when the initialized surface patch is parallel to the direction of the external force vector in 3D space. In order to develop a robust algorithm, a novel internal force is introduced in our method. A positive normal of the current deforming surface patch is calculated. The difference between this normal and the direction of the external force vector will then be a factor of the internal force. Because in the ideal condition, the positive normal of the current deforming surface patch will be the same with the direction of the external force vector. This novel internal force could rotate the deforming surface patch and let it move perpendicularly to the external force vector and find the correct boundary when the surface patch is badly initialized. Thus, the proposed method is not sensitive to the surface patch initialization. The experimental results show that it can also improve the surface extraction accuracy in segmenting object with sharp corner.
- (4) A multistage segmentation mechanism is introduced into the 3D GDSP model. A hierarchical framework is introduced into the proposed GDSP model to speed up algorithm. For the two sub-sampled images, we first calculate the external force field utilizing 3D GVF because of its wide search region. A traditional method is utilized to calculate the external force field for the original image. At the beginning stage, in order to reduce the unnecessary computational cost, the surface patches are set with big size for coarse boundary extraction. Then, each patch is examined and the patches are divided into smaller size to obtain the detail of the target. The experimental results show

that the computational costs are reduced with the help of the hierarchical framework.

- (5) An efficient algorithm for surface area calculation of objects of interest was developed as well in this thesis. In the explicit robust GDSP model, the smallest element is a surface patch S_i which consists of 16 knots. The final result M consists of a number of surface patches S which are connected to each other in sequence without overlapping or gap. For each surface patch S_i , Lagrange interpolation is utilized to calculate the surface area. The surface area of the entire model is obtained by summarizing areas of all the surface patches. The experimental results show that the proposed method can calculate the surface area of the target efficiently with high accuracy.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

- **Chapter 2** A literature review of the existing 2D and 3D object segmentation methods from thresholding to region-based segmentation is presented. Deformable models are introduced in detail because of their wide application in this field. Their categories, properties, applications are described in this Chapter. B-spline based deformable model and an introduction of B-spline curve and bi-cubic B-spline surface are presented in this Chapter as well. Statistical deformable models, topologically adaptable deformable models are introduced as well.
- **Chapter 3** The details of the proposed B-Surface algorithm for specific smooth 3D object segmentation and feature extraction are described. The external force field, surface model construction and deformation strategy are presented in detail. Experimental results of the proposed B-Surface algorithm are also presented.

- **Chapter 4** A novel GDSP model, designed for the segmentation of complicated 3D objects, is presented. First, the idea of the growing mechanism to achieve topological adaption is presented in detail. It is followed by the implementation of surface model deformation with growing mechanism. Experimental results and discussion are presented as well.
- **Chapter 5** A robust GDSP model for segmentation of 3D object is presented. In order to remove the sensitivity to initialization, a novel internal force is developed in the proposed method. The idea of hierarchal framework and the details of multi-scale GDSP model are given as well. Finally, an efficient surface area calculation of objects of interest is given. Experimental results obtained using the proposed method are presented as well.
- **Chapter 6** This Chapter provides the conclusion of the thesis and recommendations for future research work in this area.

Chapter 2

A Literature Review

2.1 Introduction

Image segmentation is to subdivide an input image into a number of regions of interest. It plays a crucial role in computer vision. The accuracy in segmentation determines the eventual success or failure of subsequent computerized analysis procedures. There have been a lot of research effort to achieve image segmentation. However, segmentation of non-trivial images is a difficult problem in computer vision and image processing. In this Chapter, a literature review of the existing methods for 2D and 3D object segmentation is presented. Some of these techniques use only the abrupt changes in intensity values or the gray level histogram such as points, lines or edges detection in an image. Some use spatial details while others use fuzzy theoretic approaches. Deformable models are introduced in detail because of their wide application in this field. Their categories, properties, applications are also described in this Chapter. Implicit and explicit deformable models, which are the two main categories of deformable models, are introduced as well. More details on explicit deformable models are given. Firstly, the 2D active contour, which is the first deformable model, is presented by introducing its mechanism, energy functions and applications. More 3D explicit deformable models are then introduced. B-

spline curve and surface and B-spline based deformable models are reviewed as well because of their links to the proposed B-Surface algorithm. Statistical deformable models are widely used in computer vision recently. Their basic idea, training process, properties and applications are given in detail. Furthermore, a review on 3D topologically adaptable deformable models is presented.

In Section 2.2, various methods on image segmentation are reviewed. In Section 2.3, implicit and explicit deformable models are introduced in details. In Section 2.4, 2D B-spline curve and 3D B-spline surface are reviewed. The basic idea of the uniform cubic B-spline curve, uniform bi-cubic B-spline surface and Nonuniform Rational B-spline (NURBS) are presented. In Section 2.5, various B-spline based deformable models are reviewed. In Section 2.6, the widely used statistical deformable models are introduced in details. Deformable models to achieve topologically adaptable object segmentation are reviewed in Section 2.7. Concluding remarks are given in Section 2.8.

2.2 Methods on Image Segmentation

Image segmentation is based on one of the two basic properties of intensity values of the image, i.e., discontinuity and similarity. In the first category, image segmentation is achieved by extracting abrupt changes in intensity values, such as points, lines, edges in an image. Usually connecting edge segments is need to build region boundaries. A common way to detect discontinuities is to execute a kernel through the image. For point detection, an isolated point can be detected because its gray level is significantly different from its background. For line detection, we can run line detection kernels through the image and the threshold the absolute value of the result. The points that are left are the strongest responses correspond closest to the lines defines by kernels. Edge detection is a common approach for detecting meaningful discontinuities in gray level. An edge is a set of connected pixels that lie on the boundary between two regions. Thus, we can use first-order or second-order

derivatives of a digital image to detect the edge.

In the second category, regions with similar intensity values are labelled. Typical methods are thresholding [19–28], region-based segmentation [29–38], iterative pixel classification [39], such as Markov Random Field (MRF) based methods [40–47], Neural network based methods [48–55].

Thresholding enjoys a central position in image segmentation because of its intuitive properties and simplicity of implementation. For example, in an image composed of light objects on a dark background, object and background pixels have gray levels grouped into two dominant modes. One obvious way to segment the object from the background is to set a threshold to separate these two modes. Sometimes imaging factors such as uneven illumination can transform a perfectly segmentable histogram into a histogram that cannot be partitioned by a single global threshold. It is an approach for handling such a situation to divide the original image into subimages and then utilize a different threshold to segment each subimage.

Region-based segmentation is a procedure that groups pixels or subregions into larger regions based on predefined criteria. The basic approach is to start with a set of “seed” points and based on these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed. Selecting a set of seed points can be based on the nature of the problem. When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations show clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

There are many image segmentation methods which use the spatial interaction models like Markov Random Field to model digital images. It is a natural way to incorporating spatial correlations into a segmentation process to use Markov random fields as a priori models. The MRF is a stochastic process that specifies the local characteristics of an image and is combined with the given data to reconstruct the

true image. The MRF of prior contextual information is a powerful method for modelling spatial continuity and other scene features, and even simple modeling of this type can provide useful information for the segmentation process. The MRF itself is a conditional probability model, where the probability of a pixel depends on its neighborhood. It is equivalent to a Gibbs joint probability distribution determined by an energy function. The energy function is a more convenient and natural mechanism for modeling contextual information than the local conditional probabilities of the MRF. The MRF on the other hand is the appropriate method to sample the probability distribution. Derin *et al.* [56] extended the one-dimensional Bayes smoothing algorithm of Askar and Derin [57] to two dimensions to get the optimum Bayes estimate for the scene value at every pixel. In order to reduce the computational complexity of the algorithm, the scene is modeled as a special class of MRF models, called Markov mesh random fields which are characterized by causal transition distributions. The processing is carried out over relatively narrow strips and estimates are obtained at the middle section of the strips. These pieces together with overlapping strips yield a suboptimal estimate of the scene. Without parallel implementation these algorithms become computationally prohibitive.

Neural networks based approaches are attempted to achieve goals of having the output in real time and robustness of the system with respect to random noise and failure of processors. Neural networks are massively connected networks of elementary processors. Architecture and dynamics of some networks are claimed to resemble information processing in biological neurons. The massive connection architecture usually makes the system robust while the parallel processing enables the system to produce output in real time. Blanz and Gish [58] used a three-layer feed forward network for image segmentation, where the number of neurons in the input layer depends on the number of input features for each pixel and the number of neurons in the output layer is equal to the number of classes. Each neuron in a layer is fully connected with the neurons in the layer above; there are no connections among neurons within a layer, and the input and output layers are not connected

directly.

2.3 2D and 3D Deformable Models

By combining geometry, approximation, statistics theory, deformable models offer a powerful approach to image analysis. They are widely used in image segmentation, object detection, matching and tracking and motion analysis and non-rigid modelling, etc. In this Section, a review of categories of deformable models is presented.

Deformable models, which were first introduced by Kass *et al.* [1], have been widely used in object detection and image segmentation [59–64]. The shape of deformable models is controlled by internal continuity forces and external image forces using energy minimization. Internal forces which come from the model itself act as a smoothness constraint. External forces which come from the input image guide the active contour toward image features that are of interest. Usually the features are lines, edges, and/or object boundaries [65–69].

Deformable models have mainly two formulations, i.e., explicit (parametric) form and implicit form. Deformable models in parametric form are not only compact, but also robust to both image noise and boundary gaps as they constrain the extracted boundaries to be smooth. However, parametric deformable models have problem in topology transformation in real application. Deformable models in implicit form could handle topological changes naturally, but the topology is not explicit.

2.3.1 Implicit Deformable Models

Deformable models in implicit form, also known as level set techniques, were introduced by Osher and Sethian [70]. Compared with explicit deformable models, the advantage of implicit deformable models is the topological adaptability. The basic idea is to start with a closed model and move the model perpendicular to itself.

When the implicit model meets splitting merging, it takes the original interface and embed it in higher dimensional scalar function (uses scalar function $f : \mathbb{R}^4 \rightarrow \mathbb{R}$ to represent objects in $\mathbb{R}^3$), defined over the entire image. The interface is represented implicitly as the zero-th level set. Level function ϕ is defined as the distance function from the zero-th level set. The function ϕ is then evolved using a partial differential equation (PDE).

An approach to shape modelling was presented by Malladi *et al.* [71]. The interface is a closed non-intersecting hypersurface flowing along its gradient field with a constant speed or a speed dependent on curvatures. Consisting of a moving front, the model can be deformed into desired shape by applied halting criteria synthesized from the input image. It has the ability to split freely to represent more than one object of interest. The model capitalizes on a related initial value partial differential equation with which several advantages are added, i.e., ability to evolve the model in the presence of sharp corners, cusps and changes in topology; ability to model shapes with significant protrusions and holes in a seamless fashion; ability to extent to 3D in a straightforward way.

The central idea of level set methods is to track a propagating interface by embedding it as the zero level set of a higher-dimensional function. However, this embedding comes at a substantial price; one is now tracking all the level sets, not just the one of interest. In order to reduce the computational labor, Narrow Band Level Set Method was introduced by Adalsteinsson and Sethian [72]. Rather than track all the level sets, Narrow Band Level Set Method focuses only on those grid points which are located in a narrow band around the zero level set. The value of the level set function is updated only at these grid points, thus reducing the computational labor to the dimension of the interface times a constant corresponding to the width of the narrow band. Thus, the narrow band level set method has the same computational complexity as string and cell methods, while maintaining of the singular advantages of topological changes, correct weak solutions as corners and cusps, high accuracy, and straightforward extension to three dimensions.

A method based on level sets of volumes to reconstruct the shapes of 3D objects from range data was presented by Whitaker [73]. The strategy is to formulate 3D reconstruction as a statistical problem, i.e., to find that surface which is mostly likely, given the data and some prior knowledge about the application domain. The resulting optimization problem is solved by an incremental process of deformation. We represent a deformable surface as the level set of a discretely sampled scalar function of three dimensions, i.e., a volume. Such level-set models have been shown to mimic conventional deformable surface models by encoding surface movements as changes in the greyscale values of the volume. The result is a voxel-based modeling technology that offers several advantages over conventional parametric models, including flexible topology, no need for reparameterization, concise descriptions of differential structure, and a natural scale space for hierarchical representations. This paper builds on previous work in both 3D reconstruction and level-set modeling. It presents a fundamental result in surface estimation from range data, i.e., an analytical characterization of the surface that maximizes the posterior probability. It also presents a novel computational technique for level-set modeling, called the sparse-field algorithm, which combines the advantages of a level-set approach with the computational efficiency and accuracy of a parametric representation. The sparse-field algorithm is more efficient than other approaches, and because it assigns the level set to a specific set of grid points, it positions the level-set model more accurately than the grid itself.

A method for the segmentation of multiple objects from three-dimensional medical images using interobject constraints was presented by Jing Yang and Duncan, J.S [74]. This method utilized that neighboring structures have consistent locations and shapes that provide configurations and context that aid in segmentation. A maximum a posteriori (MAP) estimation framework using the constraining information provided by neighboring objects was defined to segment several objects simultaneously. They also introduced a representation for the joint density function of the neighbor objects, and define joint probability distributions over the variations

of the neighboring shape and position relationships of a set of training images. In order to estimate the MAP shapes of the objects, the model in terms of level set functions was formulated, and the associated Euler-Lagrange equations was computed. The contours evolve both according to the neighbor prior information and the image gray level information. Results and validation from experiments on synthetic data and medical imagery in two-dimensional and 3-D were demonstrated. There also have been many other works on implicit deformable models [75–83].

However, these implicit deformable models have certain drawbacks. Level set methods have wide capture range and ability of grabbing the topology of the object, but the evolving model will cross over of the object boundary if the initial model is asymmetrical with respect to the object of interest. Furthermore, it cannot work properly where it meets embedding of the object.

2.3.2 Explicit Deformable Models

In parametric deformable models, an explicit parametric representation of the object of interest is used. The first deformable model which was introduced by Kass *et al.* [1] in 2D is such a parametric deformable contour that segments image by energy minimization.

Representing the position of a snake parametrically by $\mathbf{v}(s) = (x(s), y(s))$, the energy functional is

$$E_{snake} = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s)) ds \quad (2.1)$$

where E_{int} denotes the internal energy of the contour, E_{image} gives rise to the image forces, and E_{con} is the external constraint energy.

The internal energy is composed of a first-order term controlled by $\alpha(s)$ and a second-order term controlled by $\beta(s)$:

$$E_{int} = \frac{1}{2}(\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2). \quad (2.2)$$

The image energy is derived from the image data. Considering a two dimensional image, the snake may be attracted to lines, edges or terminations.

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \quad (2.3)$$

where w is an appropriate weighting function.

The line functional is defined simply by the image itself,

$$E_{line} = I(x, y) \quad (2.4)$$

so that if w_{line} is large positive the spline is attracted to light lines (or areas) and if large negative then it is attracted to dark lines (or areas).

The edge functional is defined by

$$E_{edge} = -|\nabla I(x, y)|^2 \quad (2.5)$$

Then the snake is attracted to contours with large image gradients. In Figure 2.1, an example is shown. In the upper-left, a user has placed two snakes on the edges of the pear and potato. Then, one of the snakes is pulled away from the edge of the pear. The remaining pictures show that the snake snaps back to the boundary of the pear after he lets go.

Therefore, the contour is attracted to large image gradients, i.e. parts of the image with strong edges.

Finally, the termination functional allows terminations (i.e. free ends of lines) or corners to attract the snake.

Figure 2.1: Two edge snakes on a pear and potato. (a) The user has pulled one of the snakes away from the edge of the pear; (b)-(d) After the user lets go, the snake snaps back to the edge of the pear. (Adopted from [1]).

$$E_{term} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}} \quad (2.6)$$

where $C(x,y)$ is a slightly smoothed version of the input image.

$$C(x,y) = G_\sigma(x,y) * I(x,y) \quad (2.7)$$

These energy minimizing curves are modelled as having stiffness and elasticity and are attracted toward features such as lines and edges. Constraints can be applied to ensure that they remain smooth and to limit the degree to which they can be bent. The idea of fitting by using image evidence to apply forces to the model and minimizing an energy function is effective.

A user-interface for snakes is developed to experiment with different energy functions. A user selects starting points and exerts forces on snakes interactively in

the energy minimization procedure. An initial snake model will be pushed near the feature of interest. Once close enough, the energy minimization will pull the snake in the rest of the way.

A 3D deformable model was presented by Terzopoulos *et al.* [2]. A physically based modeling framework is proposed for shape and motion reconstruction of free-form flexible objects from their images. In this framework, objects are modeled as elastically deformable bodies subject to continuum mechanical laws. Constraints are expressed as forces applied to these bodies. A set of forces computed from the image and from model properties guides the surface towards the desired solution. The applied forces deform the elastic models and propel them through potentially complicated motions such that they satisfy the available constraints over time. They developed algorithms for inferring from natural images the structures and motions of flexible objects moving nonrigidly in three dimensions. The algorithms compute detailed 3D object models directly from image intensity data without making use of intermediate optic-flow fields or 2.5D surface representations. Figure 2.2 shows the experiment of 3D reconstruction of a finger.

A deformable surface model (DFSA) was introduced by Davatzikos and Bryan [3]. A thin shell with thickness w , denoted by $\mathcal{C}$ is used as the surface model. The central layer $\alpha(u, v)$ of $\mathcal{C}$ is a closed surface, which is comprised of two open surfaces and are connected with each other along their common boundary. The deformable surface deforms under the influence of internal and external forces based on this surface model. Figure 2.3 shows frames from an DFSA search for human brain.

A method for approximating digital shapes by rational Gaussian (RaG) surfaces was presented by Jackowski *et al.* [4]. Points in a shape are parametrized by approximating the shape with a triangular mesh, determining parameter coordinates at mesh vertices, and finding parameter coordinates at shape points from interpolation of parameter coordinates at mesh vertices. Knowing the locations and parameter coordinates of the shape points, the control points of a RaG surface are determined to approximate the shape with a required accuracy. The experimental results are

Figure 2.2: Initial 3D reconstruction of a finger. (a) Finger stereo pair for first time instant. (b) User-initialized cylinder. (c)-(f) Initial reconstructed shape from three viewpoints. Every other grid line has been drawn on the surface. (Adopted from [2]).

Figure 2.3: The points where a regular grid defined was mapped through DFSA using: (a) and (b) zero iterations, (c) and (d) three iterations, and (e) and (f) nine iterations of the fixed-point algorithm. The regularity of the grid, which is equivalent to the homotheticity of the map, improves with the number of iterations. (Adopted from [3]).

shown in Figure 2.4. The process starts from a small set of control points and gradually increases the control points until the error between the surface and the digital shape reduces to a required tolerance. Both triangulation and surface approximation proceed from coarse to fine. Therefore, the method is particularly suitable for multiresolution creation and transmission of digital shapes over the Internet. Application of the proposed method in editing of 3D shapes is demonstrated.

A 3D deformable model called deformable M-reps was introduced by Pizer [5] for 3D medical image segmentation. M-reps are a multiscale medical means for modeling and rendering 3D solid geometry. The representation is based on figural models, which define objects at coarse scale by a hierarchy of figures—each figure generally a slab representing a solid region and its boundary simultaneously. A single figure is a sheet of medial atoms, which is interpolated from the model formed by a net. This geometric representation has advantages in measuring both the geometric typicality and the geometry to image match in providing the efficiency advantages of segmentation at multiple scales, and in characterizing the object as an easily deformable solid. The use of single figure models to segment objects of relatively simple structure was focused. The method was tested for the extraction of lateral ventricle from MR images by a single figure, which is shown in Figure 2.5.

Figure 2.4: (a) A digital volumetric brain. (b) Initial approximation of the brain by an octahedron. (c)-(g) Intermediate subdivision steps. (h) Brain shown by the final mesh in shaded form. (Adopted from [4]).

Figure 2.5: Segmentation results of the lateral horn of a cerebral ventricle at the m-rep level of scale (i.e., before boundary displacement) from MRI using a single figure model. (a) Lateral view. (b) Coronal view 1. (c) Coronal view 2. (d) Coronal view 3. (Adopted from [5]).

2.4 On B-spline Curve and Surface

Before the description of B-spline based deformable models in the next Section, the formulation and major properties of B-spline curve and surface are introduced here. The B-splines are piecewise polynomial functions that provide local approximations to contours or surfaces using a small number of parameters (control points). Standing as one of the most efficient curve/surface representation, they have been extensively used in the representations of curves and surfaces [84] in computer-aided designs [85], computer graphics [86] and computer vision [8, 87–95]. They possess very attractive properties such as compactness and continuity, local shape controllability, and invariance to affine transformations. B-splines can be constructed of any order (with the complexity of the possible conformation increasing with spline order), but third order (cubic) is most often encountered. Cubic B-splines exhibit a favorable tradeoff between their shape complexity for describing natural curves and computational burden required to solve them.

2.4.1 B-spline curve

A B-spline is a special case of a natural spline. Before the details of B-spline are presented, the concepts of spline and natural spline are given first.

When a smooth curve passing through a specified sequence of points is generated, use of the shape of a curve produced by a long narrow elastic band has long been used. An elastic band used for such a purpose is called a spline. It is a piecewise polynomial function that can have a locally very simple form, yet at the same time be globally flexible and smooth. Splines are very useful for modeling arbitrary functions, and are used extensively in computer graphics.

A polynomial spline function of order M and degree $m = M - 1$, having an increasing sequence of real numbers:

$$x_{-1} < x_0 < x_1 < \cdots < x_{n-1} < x_n < x_{n+1}$$

as knots, is a function $S(x)$ which satisfies the following two conditions:

- (a) $S(x)$ is a polynomial of degree m or less within each interval (x_{i-1}, x_i) ($i = 0, 1, 2, \dots, n+1$; $x_{-1} = -\infty, x_{n+1} = +\infty$).
- (b) $S(x)$ and its 1-st, 2-nd, ..., $(m-1)$ -st derivatives are continuous in (x_{-1}, x_{n+1}) .

Over the entire domain of x , we have:

$$S(x) = P_{m,0}(x) + \sum_{i=0}^n c_i (x - x_i)_+^m. \quad (2.8)$$

where

$$(x - x_i)_+^m = \begin{cases} (x - x_i)^m & (x > x_i) \\ 0 & (x \leq x_i) \end{cases}$$

A spline which is at most of degree $k - 1$ in the two end intervals (x_{-1}, x_0) and (x_n, x_{n+1}) and $2k - 1$ in the other intervals is called a natural spline. It can be expressed in the following form:

$$S(x) = P_{k-1,0}(x) + \sum_{i=0}^n c_i (x - x_i)_+^{2k-1}. \quad (2.9)$$

A natural spline that can be expressed as follows in each interval is called a B-spline

of degree 2. This is given as follows:

For $x_j \leq x \leq x_{j+1}$,

$$S(x) = \frac{1}{(x_{j+1} - x_j)(x_{j+2} - x_j)} (x - x_j)^2 \quad (2.10)$$

For $x_{j+1} \leq x \leq x_{j+2}$,

$$S(x) = \frac{1}{(x_{j+1} - x_j)(x_{j+2} - x_j)} \left[(x - x_j)^2 - \frac{(x_{j+2} - x_j)(x_{j+3} - x_j)}{(x_{j+2} - x_{j+1})(x_{j+3} - x_{j+1})} (x - x_{j+1}) \right] \quad (2.11)$$

For $x_{j+2} \leq x \leq x_{j+3}$,

$$S(x) = \frac{1}{(x_{j+3} - x_{j+1})(x_{j+3} - x_{j+2})} (x - x_{j+3})^2 \quad (2.12)$$

A normalized B-spline of degree 3 can be derived by a similar method.

Unlike a natural cubic spline, a B-spline has local control using a small number of parameters, i.e., control points. This means that modifying one control point only affects the part of the curve near that control point. This is very useful when using the B-splines for designing shapes.

The attractive properties of B-splines make them suitable for curve representation. These properties are:

- (1) Compact representation. The number of variables to be estimated is reduced to the number of control points.
- (2) Smoothness and continuity. Smoothness is guaranteed by hard implicit constraints in the representation.
- (3) Shape invariance under affine transformation. B-spline representations of two

Figure 2.6: Uniform cubic B-spline curve.

sets of control points that are related through an affine transformation, are related by the same affine transformation.

- (4) Continuous formula expression. Every point of B-spline can be presented using continuous formula.
- (5) Local controllability. Local changes in B-spline are confined to the B-spline parameters locally. Modification of local parameters will not affect other portions.

2.4.2 Uniform Cubic B-spline Curve

Classified by knot vector, a uniform B-spline curve is based on a knot vector that has uniform interval. Knot vectors of nonuniform B-spline may have knot spans of

various length. This allows parameter variation between segments.

Uniform cubic B-spline is a uniform B-spline of degree 3. The term “B” is short for “Basic”, meaning that given a sufficient number of them, they can be used to represent any C^2 spline over a uniform knot sequence.

If $(n + 1)$ ordered position vectors $Q_0, Q_1, \dots, Q_{n-1}, Q_n$ are given (shown in Figure 2.6), consider the $(n - 2)$ linear combinations:

$$P_i(t) = X_0(t)Q_{i-1} + X_1(t)Q_i + X_2(t)Q_{i+1} + X_3(t)Q_{i+2}, \quad (i = 1, 2, \dots, n - 2) \quad (2.13)$$

each formed from four successive points. $X_0(t), X_1(t), X_2$, and $X_3(t)$ are polynomials in the parameter $t(0 \leq t \leq 1)$. $C_i(t)$ is a curve segment expressed in terms of varying parameter.

The condition for two neighboring curve segments $C_i(t)$ and $C_{i+1}(t)$ to be continuous at the point corresponding to $t = 1$ for the first segment and $t = 0$ for the second:

$$C_i(1) = C_{i+1}(0). \quad (2.14)$$

The condition for the 1-st derivative vector be continuous at the joint point:

$$\dot{C}_i(1) = \dot{C}_{i+1}(0). \quad (2.15)$$

The condition for the 2-st derivative vector be continuous at the joint point:

$$\ddot{C}_i(1) = \ddot{C}_{i+1}(0). \quad (2.16)$$

Thus, we have

Figure 2.7: Basic functions of uniform cubic B-spline curve.

$$X_0(t) = \frac{1}{6}(1-t)^3 \quad (2.17)$$

$$X_1(t) = \frac{1}{2}t^3 - t^2 + \frac{2}{3} \quad (2.18)$$

$$X_2(t) = -\frac{1}{2}t^3 + \frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{6} \quad (2.19)$$

$$X_3(t) = \frac{1}{6}t^3 \quad (2.20)$$

Graphs of $X_{i,t}(t)$ in the range $0 \leq t \leq 1$ are shown in Figure 2.7.

The curve segment of cubic B-spline can be expressed as:

$$\begin{aligned}
 P_i(t) &= [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \\
 &= [t^3 \quad t^2 \quad t \quad 1] \mathbf{M}_R \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \quad (2.21)
 \end{aligned}$$

where

$$\mathbf{M}_R = \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix}$$

From the formula of uniform cubic B-spline curve, it is easy to verify that these segments have the slope and curvature continuity at the joint point of any two neighboring curve segments.

2.4.3 Uniform Bi-Cubic B-spline Surface

As shown in Figure 2.8, a uniform bi-cubic B-spline surface patch is defined by 16 control points. In order to form surfaces as a scaled sum of basis functions, the X , Y and Z of the surface must be functions of two independent parameters u and w . The basis functions will be bi-cubic, nonzero only on a local parametric region

Figure 2.8: Uniform bi-cubic B-spline surface.

nonnegative, and should sum to 1.

The formula for a bi-cubic B-spline surface patch is given as follow:

$$\begin{aligned}
 P_{i,j}(u, w) &= [X_{0,4}(u) \quad X_{1,4}(u) \quad X_{2,4}(u) \quad X_{3,4}(u)] \\
 &\quad \times \begin{bmatrix} Q_{i-1,j-1} & Q_{i-1,j} & Q_{i-1,j+1} & Q_{i-1,j+2} \\ Q_{i,j-1} & Q_{i,j} & Q_{i,j+1} & Q_{i,j+2} \\ Q_{i+1,j-1} & Q_{i+1,j} & Q_{i+1,j+1} & Q_{i+1,j+2} \\ Q_{i+2,j-1} & Q_{i+2,j} & Q_{i+2,j+1} & Q_{i+2,j+2} \end{bmatrix} \begin{bmatrix} X_{0,4}(w) \\ X_{1,4}(w) \\ X_{2,4}(w) \\ X_{3,4}(w) \end{bmatrix} \\
 &= \mathbf{u} \mathbf{M}_R \mathbf{B}_{R,i,j} \mathbf{M}_R^T \mathbf{w}^T \quad (2.22)
 \end{aligned}$$

where

$$\mathbf{B}_{\mathbf{R}i,j} = \begin{bmatrix} Q_{i-1,j-1} & Q_{i-1,j} & Q_{i-1,j+1} & Q_{i-1,j+2} \\ Q_{i,j-1} & Q_{i,j} & Q_{i,j+1} & Q_{i,j+2} \\ Q_{i+1,j-1} & Q_{i+1,j} & Q_{i+1,j+1} & Q_{i+1,j+2} \\ Q_{i+2,j-1} & Q_{i+2,j} & Q_{i+2,j+1} & Q_{i+2,j+2} \end{bmatrix}$$

$$\mathbf{u} = [u^3 \quad u^2 \quad u \quad 1], \quad \mathbf{w} = [w^3 \quad w^2 \quad w \quad 1].$$

From Equation (2.22), we have

$$P_{i,j}(0, w) = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \mathbf{B}_{\mathbf{R}i,j} \begin{bmatrix} X_{0,4}(w) \\ X_{1,4}(w) \\ X_{2,4}(w) \\ X_{3,4}(w) \end{bmatrix}$$

$$= \left(\frac{1}{6}Q_{i-1,j-1} + \frac{4}{6}Q_{i,j-1} + \frac{1}{6}Q_{i+1,j-1} \right) X_0(w) \\ + \left(\frac{1}{6}Q_{i-1,j} + \frac{4}{6}Q_{i,j} + \frac{1}{6}Q_{i+1,j} \right) X_1(w) \\ + \left(\frac{1}{6}Q_{i-1,j+1} + \frac{4}{6}Q_{i,j+1} + \frac{1}{6}Q_{i+1,j+1} \right) X_2(w) \\ + \left(\frac{1}{6}Q_{i-1,j+2} + \frac{4}{6}Q_{i,j+2} + \frac{1}{6}Q_{i+1,j+2} \right) X_3(w)$$

$$= \begin{bmatrix} 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix} \mathbf{B}_{\mathbf{R}i-1,j} \begin{bmatrix} X_{0,4}(w) \\ X_{1,4}(w) \\ X_{2,4}(w) \\ X_{3,4}(w) \end{bmatrix} \quad (2.23)$$

$$= P_{i-1,j}(1, w)$$

Similarly, we have

$$P_{i,j}(1, w) = P_{i+1,j}(0, w) \quad (2.24)$$

$$P_{i,j}(u, 0) = P_{i,j-1}(u, 1) \quad (2.25)$$

$$P_{i,j}(u, 1) = P_{i,j+1}(u, 0) \quad (2.26)$$

Therefore, two connected bi-cubic B-spline surface are positioned (C^0) continuously along their common boundary.

$$\begin{aligned} \frac{\partial P_{i,j}(u, w)}{\partial u} \Big|_{u=0} &= \frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=0} \\ &= [3u^2 \quad 2u \quad 1 \quad 0] \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=0} \\ &= [3u^2 \quad 2u \quad 1 \quad 0] \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i-1,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=1} \\ &= \frac{\partial P_{i-1,j}(u, w)}{\partial u} \Big|_{u=1} \end{aligned} \quad (2.27)$$

Similarly, we have

$$\frac{\partial P_{i,j}(u, w)}{\partial u} \Big|_{u=1} = \frac{\partial P_{i+1,j}(u, w)}{\partial u} \Big|_{u=0} \quad (2.28)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial u} \Big|_{w=0} = \frac{\partial P_{i,j-1}(u, w)}{\partial u} \Big|_{w=1} \quad (2.29)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial u} \Big|_{w=1} = \frac{\partial P_{i,j+1}(u, w)}{\partial u} \Big|_{w=0} \quad (2.30)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial w} \Big|_{u=0} = \frac{\partial P_{i-1,j}(u, w)}{\partial w} \Big|_{u=1} \quad (2.31)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial w} \Big|_{u=1} = \frac{\partial P_{i+1,j}(u, w)}{\partial w} \Big|_{u=0} \quad (2.32)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial w} \Big|_{w=0} = \frac{\partial P_{i,j-1}(u, w)}{\partial w} \Big|_{w=1} \quad (2.33)$$

$$\frac{\partial P_{i,j}(u, w)}{\partial w} \Big|_{w=1} = \frac{\partial P_{i,j+1}(u, w)}{\partial w} \Big|_{w=0} \quad (2.34)$$

Therefore, two connected bi-cubic B-spline surface are sloped (C^1) continuously along their common boundary.

$$\begin{aligned} \frac{\partial^2 P_{i,j}(u, w)}{\partial u^2} \Big|_{u=0} &= \frac{\partial^2 \mathbf{u}}{\partial u^2} \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=0} \\ &= [6u \quad 2 \quad 0 \quad 0] \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=0} \\ &= [6u \quad 2 \quad 0 \quad 0] \mathbf{M}_R \mathbf{B}_{\mathbf{R}_{i-1,j}} \mathbf{M}_R^T \mathbf{w}^T \Big|_{u=1} \\ &= \frac{\partial^2 P_{i-1,j}(u, w)}{\partial u^2} \Big|_{u=1} \end{aligned} \quad (2.35)$$

Similarly, we have

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial u^2} \Big|_{u=1} = \frac{\partial^2 P_{i+1,j}(u, w)}{\partial u^2} \Big|_{u=0} \quad (2.36)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial u^2} \Big|_{w=0} = \frac{\partial^2 P_{i,j-1}(u, w)}{\partial u^2} \Big|_{w=1} \quad (2.37)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial u^2} \Big|_{w=1} = \frac{\partial^2 P_{i,j+1}(u, w)}{\partial u^2} \Big|_{w=0} \quad (2.38)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial w^2} \Big|_{u=0} = \frac{\partial^2 P_{i-1,j}(u, w)}{\partial w^2} \Big|_{u=1} \quad (2.39)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial w^2} \Big|_{u=1} = \frac{\partial^2 P_{i+1,j}(u, w)}{\partial w^2} \Big|_{u=0} \quad (2.40)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial w^2} \Big|_{w=0} = \frac{\partial^2 P_{i,j-1}(u, w)}{\partial w^2} \Big|_{w=1} \quad (2.41)$$

$$\frac{\partial^2 P_{i,j}(u, w)}{\partial w^2} \Big|_{w=1} = \frac{\partial^2 P_{i,j+1}(u, w)}{\partial w^2} \Big|_{w=0} \quad (2.42)$$

Therefore, two connected bi-cubic B-spline surfaces are curvature (C^2) continuous along their common boundary.

2.4.4 Non-Uniform Rational B-spline (NURBS)

The convex hull, local support, affine invariance, shape preserving form properties of Non-Uniform Rational B-spline (NURBS) are extremely attractive especially in engineering design applications [96–101]. It can also be applied in deformable models for image analysis [18, 102–105].

With non-uniform B-spline, multiple internal knot values and unequal spacing between the knot values can be chosen. Non-uniform B-splines provide increased flexibility in the control of a curve of surface shape. Different shapes could be obtained for blending functions in different intervals, which can be used to adjust

spline shapes.

A non-uniform rational B-spline is the ratio of two non-uniform B-spline functions. For non-uniform rational B-spline curve, it can be described with the position vector:

$$P(t) = \frac{\sum_{k=0}^n w_k Q_k X_{i,k}(t)}{\sum_{k=0}^n w_k X_{i,k}(t)} \quad (2.43)$$

where $w_i, i = 0, \dots, n$ denotes the weight associated with B-spline control points.

If a parametric NURBS curve is extended into a bi-direction parametric surface, it can be described in the following form:

$$P(u, w) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} Q_{i,j} X_i(u) X_j(w)}{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} X_i(u) X_j(w)} \quad (2.44)$$

where $w_{i,j}, i = 0, \dots, n, j = 0, \dots, m$ denotes the weight associated with B-spline control point matrix $Q_{i,j}$.

Compared to non-rational splines, rational splines have two important advantages. First, they provide an exact representation for quadric curves. This allows modelling all curve shapes with one representation without a library of curve functions to handle different design shapes. Another advantage is their invariance with respect to a perspective viewing transformation. Therefore, a correct view of the curve can be obtained when a perspective viewing transformation to the control points is applied. Therefore, NURBS have become a very popular geometry definition method.

However, NURBS is not that widely used in deformable models for image segmentation and analysis, compared with their position in computer graphics. One of the important reasons is that normally the input data in real application are noisy and blurry. The increased computational cost from upgrading non-rational B-splines to NURBS could not improve the accuracy of the result significantly.

2.5 B-spline Based Deformable Models

Based on the idea of Active Contours, many solutions have been suggested to yield more stable and faster results. Among these methods, an alternative approach was introduced by using a parametric B-spline representation of the curve and surface [106–114].

A deformable B-spline curve model was introduced by Klein *et al.* [6] for determining vessel boundaries, and enhancing their centerline features. A bank of even and odd S-Gabor filter pairs of different orientations are convolved with vascular images in order to create an external snake energy field. Each filter pair will give maximum response to the segment of vessel having the same orientation as the filters. The resulting responses across filters of different orientations are combined to create an external energy field for snake optimization. Vessels are represented by B-Spline snakes, and are optimized on filter outputs with dynamic programming. The points of minimal constriction and the percent-diameter stenosis are determined from a computed vessel centerline. The system has been statistically validated using fixed stenosis and flexible-tube phantoms. It has also been validated on 20 coronary lesions with two independent operators, and has been tested for interoperator and intraoperator variability and reproducibility. The system has been found to be specially robust in complex images involving vessel branchings and incomplete contrast filling.

The system proposed identifies vessel boundaries using energy minimizing snakes by integrating local image intensity information over a large spatial extent. These snakes use piecewise cubic B-splines as their basis functions, which efficiently and accurately describe natural contours such as those encountered in coronary angiography. Their piecewise continuous description of complex borders makes them well suited to optimization with the dynamic programming scheme described. To provide the external energy to drive the snakes onto the vessel boundaries, we have chosen to combine the outputs of a bank of S-Gabor filters. These filters are tuned

Figure 2.9: Optimization of snakes on 2.00 mm fixed-stenosis phantom: (a) initialized, (b) after five iterations of DP algorithm, (c) after ten iterations, (d) after 15 iterations, (e) after 20 iterations, and (f) optimum reached at 23 iterations, lines connect initial control point locations and optimized locations. (Adopted from [6]).

to characteristic vessel dimensions, accurately enhance both lines and edges simultaneously, and are selective of feature orientation, capturing the tortuosity of blood vessels. The unique combination of described methods produces a powerful tool for accurate and robust identification of complex boundaries of vessels in coronary angiograms; a tool which is relatively insensitive to vessel intersections and branches which confound similar systems. Figure 2.9 shows an example.

Cham and Cipolla [7] developed an automatic and reliable scheme for spline-fitting. The proposed method incorporates B-spline active contours, the minimum description length (MDL) principle, and a novel control point insertion strategy based on maximizing the Potential for Energy-Reduction Maximization (PERM). PERM strategy is illustrated in Figure 2.10. In this method, (1) isolating and discussing the main issues to spline-fitting, describing; (2) introducing a novel control-point

Figure 2.10: Figure illustrating the PERM strategy. (Adopted from [7]).

insertion strategy which attempts to maximize error-reduction through the iterative insertion of control-points. This addresses the correct issue as opposed to the case in previous heuristic schemes; (3) combining different techniques into a highly functional and automated spline-fitting algorithm. The experimental result is shown in Figure 2.11.

Wang *et al.* [8] presented a B-spline based deformable model, named B-Snake, for lane detection. In the model, a open B-Snake lane model was constructed based on the perspective relationship of lane boundaries on the image plane. A robust algorithm called CHEVP (Canny/Hough Estimation of Vanishing Points) is presented for initializing a lane model. A minimum mean square energy (MMSE) method has been formulated to update the control points of B-Snake to approaching the real lane boundaries. The results of lane detection are shown in Figure 2.12.

Huang *et al.* [9] presented a deformable B-spline surface models for anatomical object volumes, which is relative to our work. This deformable B-spline tube facilitates the calculations of the volume enclosed based on a closed-form volume calculation formula. In their methods, the energy function consists of external and internal energies. The external energy is the integral of a predefined potential function on the model surface. The internal energy is defined by the second order derivatives of the model. By minimizing the second order derivatives of the surface, the internal energy smooths the surface. In the iterative process of their deformable B-spline

Figure 2.11: (a) The target contour. (b) Spline with nine control points (GA). (c) Spline with nine control points(CC). (d) Spline from GA which has similar averaged errors to the spline in (c). (e) Spline from (c) superimposed on the original image. (f) A graph comparing squared errors when using different numbers of control points with the two algorithms. (Adopted from [7]).

Figure 2.12: Lane detection results using B-Snake. (Adopted from [8]).

Figure 2.13: The reconstructed surface for the epicardial boundary of the left-ventricle. (Adopted from [9]).

surface models, only dx and dy of the control points are updated, dz is set to 0 all the time. Therefore, their method is limited to tubular anatomical object segmentation. The reconstructed surface for the epicardial boundary of the left-ventricle is shown in Figure 2.13.

2.6 Statistical Deformable Models

Cootes *et al.* [10, 115] described Active Shape Models (ASM) which include a *prior* knowledge of object shape by incorporating *prior* probability distributions on the shape variables to be estimated. 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. The points can be connected to form a deformable contour. By examining the statistics of the positions of the labeled points a “Point Distribution Model” (PDM) is derived. The

model gives the average positions of the points, and has a number of parameters which control the main modes of variation found in the training set. Given such a model, image interpretation involves choosing values for each of the parameters to find the best fit of the model to the image. ASM has been used for several segmentation tasks in medical images [115–125].

First of all, the training set is labelled by choosing points around the boundary of the object of interest. In order to be able to compare equivalent points from different shapes, they must be aligned with respect to a set of axes in the second step. Then the statistics of a set of aligned shapes is captured. After aligned, each example in the training set (object in 2D image) can be represented by a single point in a $2n$ dimensional space. Thus, a set of N example shapes gives a cloud of N points in this $2n$ dimensional space. Thus, the $2n \times 2n$ covariance matrix $\mathbf{S}$ is:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^N d\mathbf{x}_i d\mathbf{x}_i^T. \quad (2.45)$$

where $d\mathbf{x}_i$ is deviation of each shape $\mathbf{x}_i$ from the mean $\bar{\mathbf{x}}$.

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (2.46)$$

The principal axes of the ellipsoid are described by $\mathbf{p}_k$, the unit eigenvectors of $\mathbf{S}$:

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k. \quad (2.47)$$

where λ_k is the k th eigenvalue of $\mathbf{S}$, $\lambda_k > \lambda_{k+1}$.

A shape can be approximated using the mean shape and a weighted sum of the deviations:

$$\mathbf{x}_k = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}, \quad (2.48)$$

where $\mathbf{P}$ is the matrix of the eigenvectors, $\mathbf{b}$ is a vector of weights.

Generating new shapes is allowed by varying the parameters $\mathbf{b}$ within suitable limits, thus, the new shapes are similar to those in the training set.

Since ASMs “learn” specific patterns of variability from a representative training set of the structures to be modeled by incorporating the *prior* knowledge, it can rapidly locate the modeled structures in noisy, cluttered images—even if they are partially occluded. Figure 2.14 demonstrates using ASMs to locate resistors in printed circuit board image.

Duta and Sonka [11] present a method to improve ASM by constructing a knowledge-based PDM. Three additional features of MR brain images were included in the model: Gray-level appearance, border strength and average position. The searching procedure developed for this method is based on the model fitting strategy. The fitting procedure differs from the ASM of Coots. First, the search is entirely model driven. At each step of the fitting process, several model location hypotheses are considered and evaluated. During the hypothesis generation, the actual image data play no role. Second, an outlier detection and replacement procedure has been developed to detect misplaced points and infer their new positions. The outlier detection improves robustness and accuracy of the shape model fitting process. The searching procedure consists of the following steps: (1) model fitting using linear transformations; (2) model fitting using piecewise linear transformations; (3) outlier removal; (4) final point adjustment, and (5) final outlier removal. This procedure is shown in Figure 2.15.

Coots *et al.* present the Active Appearance Model (AAM) [12, 126, 127] search algorithm which can locate points on a new image, making use of constraints of the shape models. AAM has been used widely for several segmentation tasks [128–137]. One disadvantage is that it only uses shape constraints (together with some information about the image structure near the landmarks), and does not take advantage of all the available information - the texture across the target object.

Figure 2.14: Section of printed circuit board with resistor model superimposed, showing its initial position and its location after 30, 60, 90 and 120 iterations. (Adopted from [10]).

Figure 2.15: Example of automated brain image segmentation and interpretation: (a) manual tracings; (b) initial average position of the shape model; (c) optimal shape model position after linear transform step; (d) optimal shape model position after piecewise linear transform step; (e) outlier detection — marked by dark dots; (f) final point adjustment. Images contrast-enhanced to improve visibility. (Adopted from [11]).

Interpretation is treated as an optimization problem in which the difference between a new image and one synthesized is minimized by the appearance model. The appearance model has parameters controlling the shape and texture (in the model frame). To build this model, it requires a training set of annotated images where corresponding points have been marked on each example. Procrustes analysis is applied to align the sets of landmark points and build a statistical shape model. Then each training image was warped so the points match those of the mean shape, obtaining a shape-free patch. Eigen-analysis is applied to build a texture model for each landmark point. Finally, the correlations between shape and texture are learned to generate a combined appearance model. Matching an appearance model to a target image involves finding an affine transformation, global intensity parameters and appearance coefficients that minimize the root-mean-square difference between the appearance model instance and the target image. The iterative method described by Coots suggested using a gradient descent method that relates model coefficients with the difference between a synthesized model image and the target image. Figure 2.16 shows frames from an AAM search for each face, each starting with the mean model displaced from the true face centre.

Figure 2.16: Multi-Resolution search from displaced position. (Adopted from [12]).

2.7 Topologically Adaptable Deformable Models

A major problem that arises in shape recovery and segmentation is the possible complex topology of objects in 3D data. For example, in real applications of medical image analysis, the object of interest is often with complex topology, such as vertebra, brain ventricle, brain skull. However, traditional explicit deformable models are incapable of topological transformation without additional machinery. Parametric deformable models are bound to their intrinsic topology. Deformable models give prior information on the shape of the objects to recover. If the prior model shape is too different from the data, the model might not be able to deform correctly.

In order to handle this problem, additional machinery has been introduced into explicit deformable models to achieve the ability of topology change. Merging and splitting of the model are shown in Figure 2.17. The term topology change may include both change in the mesh structure or change of the surface genus which transforms the surface such as it is not homeomorphic to its previous configuration.

An approach which is designed for surface modelling and reconstruction by Szeliski *et al.* [14] overcomes the limitations of traditional deformable models, i.e., the restrictive assumptions about object topology. The first component of the approach is a dynamic, self-organizing, oriented particle system which discovers topological and geometric surface structure implicit in visual data. Because the particle system does not give an explicitly triangulated surface directly, the second component is a triangulation scheme which connects the particles into a continuous global surface model that is consistent with the inferred structure. An experimental result is shown in Figure 2.18.

Topological evolution of surfaces are introduced by DeCarlo and Gallier [15]. A framework for generating smooth-looking transformations between pairs of surfaces that differ in topology is presented. By specifying a sparse control mesh on each surface and by associating each face in one control mesh with a corresponding face in the other, the transformation is controlled by the user. The algorithm builds a

Figure 2.17: (a) Implementation of axial transformations: triangulation between the two surfaces and deletion of the two old vertices; (b) implementation of annular transformations: splitting of the problematic set UVO in two parts, then separate merging. (Adopted from [13]).

Figure 2.18: Reconstruction of a surface model of a cup from silhouettes: (a) cup bounding volume represented as an octree, (b) triangulated surface of reconstructed model. (Adopted from [14]).

transformation from this information in two steps. The first step constructs a series of shapes and meshes that describes how topological changes should occur at critical points during the transformation. In the second step, smooth transformations is established by combining intermediate shapes in this series. Control of morph is achieved through user's control on meshes. Figure 2.19 shows two different sequences of a sphere-torus morph. Of interest to topology fans, is the fact that these two transitions are qualitatively different. In both cases, the poles of the sphere are brought together. Yet in (a), the lines pass through the torus hole, while in (b) they go around the torus hole.

Extended from 2D topologically adaptable snakes [138–140], a dynamic model, topologically adaptable surfaces (T-surfaces), introduced by McInerney and Terzopoulos [16, 141], can automatically change its topology with regard to its variable geometry. In order to allow the models to extract and reconstruct complex object structure, an Affine Cell Image Decomposition (ACID) framework is embedded. The ACID framework creates a mechanism for multiresolution deformable surface

Figure 2.19: Morphing between a sphere and a torus (a) pinched sphere (b) strangled torus. (Adopted from [15]).

to “flow” or “grow” into objects with complex geometries and topologies. T-surfaces use a superposed affine cell grid to reparameterize the models during their evolution. As the T-surface move under the influence of external and internal forces, it is reparameterized with a new set of nodes and triangles by computing the intersection points of the model with the superposed grid. This reparametrization performs topological transformations in an implicit way. Figure 2.20 shows an experimental result which uses T-surface to segment vertebra phantom from CT image volume.

An explicit deformable model, i.e., deformable meshes with automated topology changes, was presented by Lachaud and Montanvert [13, 142]. In this model, a framework for topology changes is proposed to extract complex object. Within this framework, the model dynamically adapts its topology to the geometry of its vertices according to simple distance constraints. Therefore, no a priori assumption is made on the topology of objects. Detection and resolution of topological breaks are optimized by regularizing edge lengths over the whole mesh. The topology of the model is adapted to the geometry of its vertices at each step without user interaction, topological modifications are made locally. The deformation procedure of an experiment is shown in Figure 2.21. In order to speed up the process, an algorithm of pyramid construction with any reduction factor transforms the image

Figure 2.20: T-surface segmenting vertebra phantom from CT image volume. The dark shaded regions are frozen and have been removed from the computation. (Adopted from [16]).

Figure 2.21: Surface evolution (no Gouraud shading is shown): (a) iteration 0 on image I_0 ; (b) iteration 200 on image I_0 ; (c) iteration 400 on image I_0 ; (d) iteration 1100 on image I_0 . (Adopted from [13]).

into a set of images with progressively higher resolutions. This organization into a hierarchy, combined with a model which can adapt its sampling to the resolution of the workspace, enables a fast estimation of the shapes included in the image. After that, the model searches for finer and finer details while relying successively on the different levels of the pyramid.

An Intelligent Balloon, named subdivision-based deformable model, was proposed for surface reconstruction of arbitrary topology by Duan and Qin [17]. This Intelligent Balloon is a parameterized subdivision surface whose geometry and its deformable behaviors are governed by the principle of energy minimization. Starting from a seed model that is initiated by users within regions of interest, the model is deformed under the control of a locally defined objective function associated with each vertex. Through the numerical integration of function optimization, it can subdivide the model geometry, detect self-collision of the model, properly modify its topology, evolve the model towards the region boundary and reduce fitting error

Figure 2.22: Shape recovery from volumetric image data a phantom vertebral. (Adopted from [17]).

and improve fitting quality via global subdivision. The task of recovering a shape of arbitrary was achieved by a distance based collision detection and topology changes. If the distance of two non-neighbor active vertices is smaller than the threshold, a collision will be identified and a merge-operation is triggered. To merge the two parts of the model, it was necessary to identify and collect all the one-neighborhood points for each of these two vertices. Then these two sets of points are sequenced separately and are put into correspondence. Then each point is connected with its corresponding point in the opposite point set. The result of an experiment is shown in Figure 2.22.

2.8 Concluding Remarks

In this Chapter, a review of image segmentation methods has been presented. Various methods for image segmentation, from edge detection and gray level thresholding

to neural network based approaches, are introduced. Because of the wide applications of deformable models to object segmentation and shape recovery, various 2D and 3D deformable models such as active contours, level set methods, statistical deformable model and topologically adaptable deformable models have been presented.

B-spline based deformable models possess some attractive properties such as spatial uniqueness, boundedness and continuity, local shape control ability and invariance to affine transformations. In Chapter 3, a new 3D bi-cubic B-spline surface based deformable model for object segmentation and feature extraction will be presented. In the existing explicit deformable models with topologically adaptive ability, several additional machineries were introduced into traditional 3D deformable models in order to achieve automated topology transformation. In Chapter 4, a novel method named GDSP model will be presented to achieve complex object segmentation and shape recovery with topologically adaptive ability. In Chapter 5, a robust algorithm for 3D object segmentation is presented. A novel internal force to improve the performance of the proposed method is given as well. The idea of hierarchal framework and the details of multi-scale GDSP model are given. An efficient algorithm for surface area calculation of objects of interest was developed as well in Chapter 5.

Chapter 3

A Novel B-Surface Algorithm for Segmentation of Smooth 3D Objects

3.1 Introduction

For the past decades, there has been a significant research effort for achieving 3D objects modelling based on deformable models [143, 144]. Various representations have been used to fulfill different 3D modelling needs, from deformable lines to deformable surfaces and volumes. The basic mechanism of 3D deformable models is similar to the one in the 2D case. The 3D models are deformed under the combined influence of the external energy which comes from the image, and the internal energy which come from the model itself. Sometimes, other forces are introduced to increase the performance of deformable models, for example, “weight” force, pressure force [145].

In this Chapter, a new deformable model algorithm, B-Surface, for smooth 3D object segmentation is presented. A true 3D deformable surface model algorithm,

which works in 3D space directly, is proposed. This method is robust against the shortcomings typical of sampled data which is common in real applications. It can find the boundaries of structures accurately even under the situation that the boundaries of structures are indistinct. The more information we obtain from the object of interest, the more features we could extract from the object for high-level image analysis. The differential quantities such as surface normals or curvatures of the surface are useful in image analysis. For example, the brain surface curvature is introduced in the analysis of variability between males and females [18]. In order to obtain more information regarding the object of interest for high-level image analysis, we generate a continuous deformable model.

In the proposed algorithm, a 3D edge extraction method is developed to obtain the coarse boundary of the target. An improved 3D external force field combined with the normalized GVF is then computed. The surface model is deformed under the influence of this external force field. After the initialization of a surface model near the target, B-Surface starts deformation steps to locate the boundary of the object.

In Section 3.2, the details on how to calculate the external force field is presented. In Section 3.3, the presentation of the B-Surface model is described. The details on deformation of the B-Surface model is presented in Section 3.4. Experimental results are given in Section 3.5.

3.2 External Force Field for the B-Surface Algorithm

B-Surface is a continuous surface model which is deformed under the influence of force field to locate edge of the object of interest in 3D volume image. The function of internal forces in the traditional deformable models is mainly to keep the model continuous and smooth. In the proposed B-Surface algorithm, internal forces are not necessary. It is because B-spline surface has strong implicit constraint itself to

ensure surface smoothness. Before we start the deformation step of B-Surface, 3D external force field is generated from the volume image. Using energy minimization, 3D external forces will guide B-Surface model to move towards image features that are of interest.

3.2.1 Extraction of the Coarse Boundary

Before the computation of the external force field from the given 3D volume image, the coarse boundary of the object will be extracted.

A 3D edge extraction method using 3D convolution is generated to extract the boundary of the object. Using Equation (3.1), the coarse boundary volume image, f , could be extracted from the input volume image I . The volume image is convoluted with a 3D convolution kernel k to extract 3D object boundary in 3D space directly. There is no need to separate the 3D volume image into 2D slices and to extract object edge respectively in each 2D slice.

$$f = I \otimes k \quad (3.1)$$

where, k is the 3D convolution kernel:

$$k_{(:, :, 1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$k_{(:, :, 2)} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$k_{(:, :, 3)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

As mentioned above, segmenting target by processing 2D images slice by slice has some inherent shortcomings. For example, when the slice is near to the top of the ventricle in 3D MR volume image, the edge of the ventricle in the $x-y$ plane becomes blurred because the edge intensity changes smoothly in x and y direction. It is very difficult to decide where is the exact boundary of the ventricle even for human eyes, if only analyzing the intensity information from one slice. In fact, in this situation, the contour of the ventricle edge in each slice is not single-pixel-contour anymore. If fail to determine the proper width of the edge contour, it is difficult to obtain a position-continuous 3D model. In the proposed algorithm, the intensity differences of each voxel between the nearby voxels in x, y and z directions are considered at the same time. Therefore, it can extract the boundary successfully when the intensity differences of one voxel between the nearby voxels in either one direction is not zero.

An experiment of ventricle boundary extraction is shown in Figure 3.1. Three slices of the 3D volume image (256x256x138), from 87th to 89th slice, are presented in the left column in Figure 3.1. In the 88th slice of the brain image, it reaches the top part of the ventricle, where the boundary becomes indistinct. In this slice, the width of the boundary contour is not 1-voxel-wide any more. However, it is very difficult to judge the width of boundary contour without comparing the nearby two slices. Therefore, it cannot obtain the correct width of the 2D contour in the 88th slice using 2D image analysis method. The proposed 3D edge extraction method works in 3D space directly. The intensity information of the nearby slices is utilized to extract the boundary of the target. First, the histogram 3D image is analyzed. Based on the statistic information of the gray level at each 3D point, a threshold is chosen to separate the 3D image briefly to get the coarse edge of the object. Then the convolution kernel is applied to obtain the binary image. The result of the 3D

Figure 3.1: Result of 3D edge extraction method.

edge extraction method is shown in the right column in Figure 3.1. Representing the result in 2D space, we can see that the proper width of the edge contour could be determined by the proposed 3D edge extraction method. Therefore, a continuous coarse boundary is obtain using the proposed method.

3.2.2 External Force Field for the B-Surface Algorithm

The function of external force field in the B-Surface algorithm is to guide the proposed surface model to move towards the boundaries of the target object. Here we choose gradient vector flow (GVF) field as the proposed external force field which has a large capture range and is able to move surface model into boundary concavities.

GVF is first introduced by C. Xu [146]. As mentioned in his paper, GVF can be generalized to higher dimensions for application.

For 3D GVF, let $f(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a 3D edge map defined in $\mathbb{R}^3$. The GVF field in $\mathbb{R}^3$ is defined as the vector field $\mathbf{v}(x, y, z) = (u(x, y, z), v(x, y, z), w(x, y, z)) : \mathbb{R}^n \rightarrow \mathbb{R}$ which minimizes the energy functional

$$\varepsilon = \int \int \int \mu |\nabla \mathbf{v}|^2 + |\nabla f_{3D}|^2 |\mathbf{v} - \nabla f_{3D}|^2 dx dy dz \quad (3.2)$$

where

$$|\nabla \mathbf{v}|^2 = u_x^2 + u_y^2 + u_z^2 + v_x^2 + v_y^2 + v_z^2 + w_x^2 + w_y^2 + w_z^2,$$

and the gradient operator ∇ is applied to each component of $\mathbf{v}$ separately.

Finally, we could obtain vector field $\mathbf{v}(x, y, z) = (u(x, y, z), v(x, y, z), w(x, y, z))$ using the following iterative solution:

$$\begin{aligned} u_{i,j,k}^{n+1} &= (1 - b_{i,j,k} \Delta t) u_{i,j,k}^n + r \nabla^2 u + c_{i,j,k}^1 \Delta t \\ v_{i,j,k}^{n+1} &= (1 - b_{i,j,k} \Delta t) v_{i,j,k}^n + r \nabla^2 v + c_{i,j,k}^2 \Delta t \\ w_{i,j,k}^{n+1} &= (1 - b_{i,j,k} \Delta t) w_{i,j,k}^n + r \nabla^2 w + c_{i,j,k}^3 \Delta t \end{aligned} \quad (3.3)$$

where

$$r = \frac{\mu \Delta t}{\Delta x \Delta y \Delta z}$$

$$b(x, y, z) = f_x(x, y, z)^2 + f_y(x, y, z)^2 + f_z(x, y, z)^2$$

$$c^1(x, y, z) = b(x, y, z) f_x(x, y, z)$$

$$c^2(x, y, z) = b(x, y, z)f_y(x, y, z)$$

$$c^3(x, y, z) = b(x, y, z)f_z(x, y, z)$$

However, using this method to compute GVF, the magnitude of the external force at a space point will decrease significantly as the distance from this point to the boundary increases. In another word, if a space point is a little far away from the boundaries of the object, the magnitude of the external force at that point will be too small to deform the surface model. Thus, we combine GVF with normalized GVF using the following equation:

$$\mathbf{v}' = \alpha_1 \mathbf{v} + \alpha_2 \frac{\mathbf{v}}{\|\mathbf{v}\|} \quad (3.4)$$

Thus,

$$\begin{aligned} u_{i,j,k}^{n+1} &= (1 - b_{i,j,k}\Delta t)u_{i,j,k}^n + r\nabla^2 u_{i,j,k}^n + c_{i,j,k}^1\Delta t \\ v_{i,j,k}^{n+1} &= (1 - b_{i,j,k}\Delta t)v_{i,j,k}^n + r\nabla^2 v_{i,j,k}^n + c_{i,j,k}^2\Delta t \\ w_{i,j,k}^{n+1} &= (1 - b_{i,j,k}\Delta t)w_{i,j,k}^n + r\nabla^2 w_{i,j,k}^n + c_{i,j,k}^3\Delta t \end{aligned} \quad (3.5)$$

where

$$u' = \alpha_1 u + \alpha_2 \frac{u}{\|\mathbf{v}\|} \quad (3.6)$$

$$v' = \alpha_1 v + \alpha_2 \frac{v}{\|\mathbf{v}\|} \quad (3.7)$$

$$w' = \alpha_1 w + \alpha_2 \frac{w}{\|\mathbf{v}\|} \quad (3.8)$$

With the help of the normalized GVF, the magnitude of the external force does not decrease significantly. In this way, it is convenient and more accurate for the proposed algorithm to locate the boundary.

3.3 Presentation of the B-Surface Model

The more information we obtain from the object of interest, the more features we could extract from the object for high-level image analysis. The differential quantities such as surface normals or curvatures of the surface are useful in image analysis. Therefore, we choose B-spline surface as the surface model of the B-Surface algorithm. We can segment and represent 3D objects based on this continuous 3D surface model. It facilitates the calculation of differential quantities such as surface normals or curvatures of the surface.

A bi-cubic surface patch can be defined using the uniform B-spline functions. It is shown in Figure 2.8 in Chapter 2.

The formula for a bi-cubic B-spline surface patch is given in Chapter 2, Equation (2.22).

Here, $P_{i,j}(u, w)$ is one surface point corresponding to parameter u and w . $\mathbf{B}_{\mathbf{R}_{ij}}$ is the control points matrix. $\mathbf{u}$ and $\mathbf{w}$ are parameter vectors. As shown in Figure 2.8, when parameter u and w increase from 0 to 1, each surface point on the B-spline surface patch (red surface) is generated from the control points matrix $\mathbf{B}_{\mathbf{R}_{ij}}$ using Equation (2.22).

Based on Equation (2.22), we extend it to Equation (3.10) for calculating the i, j th B-spline surface patch.

$$\begin{bmatrix} P_{i,j}(u_1, w_1) & P_{i,j}(u_1, w_2) & P_{i,j}(u_1, w_3) & \dots & P_{i,j}(u_1, w_n) \\ P_{i,j}(u_2, w_1) & P_{i,j}(u_2, w_2) & P_{i,j}(u_2, w_3) & \dots & P_{i,j}(u_2, w_n) \\ P_{i,j}(u_3, w_1) & P_{i,j}(u_3, w_2) & P_{i,j}(u_3, w_3) & \dots & P_{i,j}(u_3, w_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{i,j}(u_m, w_1) & P_{i,j}(u_m, w_2) & P_{i,j}(u_m, w_3) & \dots & P_{i,j}(u_m, w_n) \end{bmatrix}$$

$$= \begin{bmatrix} u_1^3 & u_1^2 & u_1^1 & 1 \\ u_2^3 & u_2^2 & u_2^1 & 1 \\ u_3^3 & u_3^2 & u_3^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ u_m^3 & u_m^2 & u_m^1 & 1 \end{bmatrix} \mathbf{M}_R \mathbf{B}_{R_i} \mathbf{M}_R^T \begin{bmatrix} w_1^3 & w_1^2 & w_1^1 & 1 \\ w_2^3 & w_2^2 & w_2^1 & 1 \\ w_3^3 & w_3^2 & w_3^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ w_n^3 & w_n^2 & w_n^1 & 1 \end{bmatrix}^T \quad (3.9)$$

We obtain

$$\mathbf{P}_{i,j} = \mathbf{U} \mathbf{M}_R \mathbf{B}_{R_i} \mathbf{M}_R^T \mathbf{W}^T \quad (3.10)$$

where

$$\mathbf{P}_{i,j} = \begin{bmatrix} P_{i,j}(u_1, w_1) & P_{i,j}(u_1, w_2) & P_{i,j}(u_1, w_3) & \dots & P_{i,j}(u_1, w_n) \\ P_{i,j}(u_2, w_1) & P_{i,j}(u_2, w_2) & P_{i,j}(u_2, w_3) & \dots & P_{i,j}(u_2, w_n) \\ P_{i,j}(u_3, w_1) & P_{i,j}(u_3, w_2) & P_{i,j}(u_3, w_3) & \dots & P_{i,j}(u_3, w_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{i,j}(u_m, w_1) & P_{i,j}(u_m, w_2) & P_{i,j}(u_m, w_3) & \dots & P_{i,j}(u_m, w_n) \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} u_1^3 & u_1^2 & u_1^1 & 1 \\ u_2^3 & u_2^2 & u_2^1 & 1 \\ u_3^3 & u_3^2 & u_3^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ u_m^3 & u_m^2 & u_m^1 & 1 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} w_1^3 & w_1^2 & w_1^1 & 1 \\ w_2^3 & w_2^2 & w_2^1 & 1 \\ w_3^3 & w_3^2 & w_3^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ w_n^3 & w_n^2 & w_n^1 & 1 \end{bmatrix}$$

Here, m, n is the number of surface points in u, w direction when the parameter u, w increase from 0 to 1 respectively. We set $n = m$ for computational convenience. Therefore, $\mathbf{P}_{i,j}$ is a $n \times n$ matrix. It is a patch of surface consists of $n \times n$ surface points.

In the proposed B-Surface algorithm, the shape of the surface model is controlled by the position of the control points. After the control points matrix $\mathbf{B}_{R_{ij}}$ is updated in each deformation step, B-Surface model is deformed towards the boundaries of the object.

3.4 Deformation of the B-Surface Model

In B-Surface algorithm, the shape of the surface model is deformed by changing the position of the control points. However, the external forces do not have direct influence on the control points of the B-Surface model. The position of the control points cannot be updated by the influence of external force field directly. However, we can compute the external force on every surface point. Next, the relationship between surface points and control points is utilized to obtain the indirect influence of external forces on the control points. This is denoted in Figure 3.2. In an iterative deformation process, this 3D deformable model is deformed by updating the position of control points to extract the boundaries of the target object according to the external force field generated from the 3D volume image.

From Equation (3.10), a B-spline surface patch could be obtained from the 16 control points. Next step, Equation (3.10) is reversed to obtain the 16 control points from the surface patch. Given $n \times n$ surface points, compute the 16 surface control points $\mathbf{B}_{R_i}$. In order to ensure the rank of $\mathbf{U}$ and $\mathbf{W}$ equal to 4, n must be greater than 4. In another word, we cannot solve Equation (3.10) for computing 16 control points only with $n \times n$ equations if $n < 4$.

From Equation (3.10), we have

$$\mathbf{U}^T \mathbf{P}_{i,j} \mathbf{W} = \mathbf{U}^T \mathbf{U} \mathbf{M}_R \mathbf{B}_{R_i} \mathbf{M}_R^T \mathbf{W}^T \mathbf{W} \quad (3.11)$$

Because $n > 4$, the rank of $\mathbf{U}$ and $\mathbf{W}$ equal to 4. We could compute $\mathbf{U}^T \mathbf{U}^{-1}$ and

Figure 3.2: To compute the external forces on surface points and calculate the influence of the external forces on control points.

$\mathbf{W}^T \mathbf{W}^{-1}$. The rank of $\mathbf{M}_R$ is 4 and $\mathbf{M}_R^{-1}$ exists. Therefore

$$\mathbf{B}_{R_i} = \mathbf{M}_R^{-1} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{P}_{i,j} \mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1} (\mathbf{M}_R^T)^{-1} \quad (3.12)$$

Equation (3.12) shows how the positions of the surface control points are changed to obtain a certain surface. Then, this equation is used to find the influence of 3D forces, which initially acts on surface points, on surface control points.

The surface points generated by the control points are in a 3D external forces field. In this field, the surface points will be pushed to the object edge by the external force. When a surface point reach the object edge, the total external force on this point is zero because external forces in x, y, z directions reach a balance here.

$FP_{i,j}(u, w)$ is the external force which has influence on surface point $P_{i,j}(u, w)$. Using Equation (3.12), we can obtain the influence of the external force on the sixteen control points, $\mathbf{B}_R$, in the i, j th surface patch. The external forces on the surface patch are

$$\mathbf{FP}_{i,j} = \begin{bmatrix} FP_i(u_1, w_1) & FP_i(u_1, w_2) & FP_i(u_1, w_3) & \dots & FP_i(u_1, w_n) \\ FP_i(u_2, w_1) & FP_i(u_2, w_2) & FP_i(u_2, w_3) & \dots & FP_i(u_2, w_n) \\ FP_i(u_3, w_1) & FP_i(u_3, w_2) & FP_i(u_3, w_3) & \dots & FP_i(u_3, w_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ FP_i(u_n, w_1) & FP_i(u_n, w_2) & FP_i(u_n, w_3) & \dots & FP_i(u_n, w_n) \end{bmatrix} \quad (3.13)$$

The surface model of B-Surface is deformed step by step in an iterative process.

After each iteration, the control point matrix $\mathbf{B}_{\mathbf{R}i}$ is updated by

$$\mathbf{B}_{\mathbf{R}i}' = \mathbf{B}_{\mathbf{R}i} + \Delta\mathbf{B}_{\mathbf{R}i} \quad (3.14)$$

$\Delta\mathbf{B}_{\mathbf{R}i}$ is the change of the control point matrix. The change of the position of each control point $Q_{i,j}$ is $\Delta Q_{i,j}$.

From Equation (3.12) and (3.13), we have

$$\begin{aligned} \Delta\mathbf{B}_{\mathbf{R}i} &= \alpha \mathbf{M}_{\mathbf{R}}^{-1} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{FP}_{i,j} \mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1} (\mathbf{M}_{\mathbf{R}}^T)^{-1} \\ &= \alpha \mathbf{A} \cdot \mathbf{FP}_{i,j} \cdot \mathbf{B} \end{aligned} \quad (3.15)$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{M}_{\mathbf{R}}^{-1} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \\ \mathbf{B} &= \mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1} (\mathbf{M}_{\mathbf{R}}^T)^{-1} \end{aligned} \quad (3.16)$$

α is the coefficient of the deformation step. If the value of α increases, the B-Surface deformation interval increases too. We set $\alpha = 0.5$.

The deformation process of B-Surface is summarized as follows:

- (1) Obtain surface points $\mathbf{P}_{i,j}$ from initialized control points using Equation (3.10).

- (2) Calculate external forces $\mathbf{FP}_{i,j}$ on curve points $\mathbf{P}_{i,j}$.
- (3) Calculate $\Delta\mathbf{B}_{\mathbf{R}_i}$ using Equation (3.15).
- (4) Replace $\mathbf{B}_{\mathbf{R}_i}$ by $\mathbf{B}_{\mathbf{R}_i} + \Delta\mathbf{B}_{\mathbf{R}_i}$.
- (5) If $\Delta\mathbf{B}_{\mathbf{R}_i} < \text{threshold}$, go to step 6; otherwise, go to step 1.
- (6) Stop.

The inverses of matrices are necessary to compute $\Delta\mathbf{B}_{\mathbf{R}_i}$ which may cause high computation load for this algorithm if they are computed at each iteration. However, before the deformation process is started, n , which decides how many voxels does a surface patch consist of is first set. $\mathbf{M}_{\mathbf{R}}^{-1}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$ and $\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}(\mathbf{M}_{\mathbf{R}}^T)^{-1}$ are fixed, denoted as $\mathbf{A}$ and $\mathbf{B}$ respectively. Therefore, $\mathbf{M}_{\mathbf{R}}^{-1}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$ and $\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}(\mathbf{M}_{\mathbf{R}}^T)^{-1}$ are only computed once, rather than at each iteration in the deformation process. Compared with other algorithms in which the internal force should be computed and updated at each iteration in their deformation process, the computation load of B-Surface algorithm is reduced because $\mathbf{A}$ and $\mathbf{B}$ are computed only once before the deformation process starts. Furthermore, we need not compute the internal force.

3.5 Experimental Results and Discussion

In this Section, the experimental implementation of the deformation step for B-Surface algorithm is introduced. B-Surface experimental results and analysis are presented.

3.5.1 Experiments on Segmentation of Simulated Object

To evaluate the proposed method, a simulated object was tested first. A sphere was chosen as the object of interest, which was defined in a $128 \times 128 \times 128$ matrix.

Figure 3.3: A simulated sphere for segmentation.

The center of the sphere was set at (64,64,64), the radius was set as 40 as shown in Figure 3.3.

Using Equation (3.1), the coarse edge of the target object was extracted. After 3D edge detection algorithm, the external force field of the object surface was calculated using 3D GVF as described in Section 3.2.2. Two slices of the 3D external force field of the sphere is shown in Figure 3.4. The arrows denote the external force vector at each point in the 3D force field. The direction of the arrow denotes the direction of the force, the magnitude of the arrow denotes the magnitude of the force.

The B-Surface model was initialized in the 3D external force field. First, 52 control point matrices were initialized. From Equation (3.10), we obtained an initial B-Surface model which consists of 52 surface patches, shown in Figure 3.5 (a). Then, under the influence of the external forces, each initial surface patch was deformed in an iterative process to reach and stop at the surface of the object. After each

Figure 3.4: External force field obtained using improved 3D GVF. (a) 3D external force vectors are depicted in horizontal plane; (b) 3D external force vectors are depicted in vertical plane.

deformation step, the surface control points $\mathbf{B}_{Rij}$ was updated, and a new surface patch was generated based on the resulting control points. We calculated the external forces on the new surface, and deformed the surface control points based on the external forces. The deformation process will be continued until the B-Surface patch stops at the edge of the object of interest.

Using Equation (3.10), we could represent the object based on the experimental result. In order to present how does each B-Surface patch work clearly, we displayed only six of all the deformed B-Surface patches in Figure 3.5 (b). The final result is shown in Figure 3.5 (c). The colored surfaces are the initial surfaces which reach and stop at the boundary of the object of interest. We can see that the B-Surface method can find the boundary of the object of interest successfully. In the final result, we can see some portion of the surface model looks dark. However, the dark portion of the surface is not the concave. Actually, it is the overlapping portion of the different B-spline surface patches. In this simulated object segmentation experiment, 52 B-spline surface patches are initialized. And for each B-spline surface patches, a large number of the surface points are used to represent the object. Each B-spline surface

patch consists 400 surface points. In the overlapping portion of the different B-spline surface patches, the dense of surface point is higher than 1 point/voxel. Therefore, it looks dark.

Using the proposed B-Surface algorithm for object segmentation and surface extraction, the normal of the surface could be obtained easily. Consider only the i, j th surface patch, from Equation (3.10), we have

$$P_u(u_0, w_0) = \frac{\partial P(u_0, w_0)}{\partial u} = \frac{\partial \mathbf{u}_0}{\partial u} \mathbf{M}_R \mathbf{B}_R \mathbf{M}_R^T \mathbf{w}_0^T \quad (3.17)$$

$$P_w(u_0, w_0) = \frac{\partial P(u_0, w_0)}{\partial w} = \mathbf{u}_0 \mathbf{M}_R \mathbf{B}_R \mathbf{M}_R^T \left(\frac{\partial \mathbf{w}_0}{\partial w} \right)^T \quad (3.18)$$

where

$$\frac{\partial \mathbf{u}_0}{\partial u} = [3u_0^2 \quad 2u_0 \quad 1 \quad 0] \quad (3.19)$$

$$\frac{\partial \mathbf{w}_0}{\partial w} = [3w_0^2 \quad 2w_0 \quad 1 \quad 0] \quad (3.20)$$

Using the following Equation, we could obtain $N(u_0, w_0)$, the normal at the surface point $P(u_0, w_0)$:

$$N(u_0, w_0) = \frac{P_u(u_0, w_0) \times P_w(u_0, w_0)}{|P_u(u_0, w_0) \times P_w(u_0, w_0)|} \quad (3.21)$$

To evaluate the experimental results, we compared the object surface obtained by B-Surface with the original sphere. Because it is a simulated object, we can easily obtain the exact error of the result. For every surface point, the distance from itself to the center of the sphere is calculated and compared with the radius of the sphere. The normal direction of the B-Surface at surface point was computed using Equation (3.21) and compared with the sphere normal at that position.

The performance of B-Surface algorithm in the experiment is presented in Table 3.1. From Table 3.1, we can see that the compression ratio is high ($400/16 = 25$).

Figure 3.5: Final experimental result of B-Surface algorithm. (a) Initial B-Surface model; (b) 6 surface patches from the all 52 are presented; (c) Final result of the sphere.

Here the B-Surface patch consisted of about 400 surface points in the experiment. The number of control points of this surface patch was only 16. The distance error was only 0.05 voxel. It reached the maximum distance error at 1 voxel only in a few places. Based on the obtained surface, we can compute the normal to the surface patch at the central point. Compared with the ground truth, the error was 3°.

A comparison has been carried out with the traditional 3D discrete deformable model Deformable Surface Algorithm (DFSA) and a B-spline surface based method (Huang’s Deformable B-spline Tube). In this simulated object segmentation experiment, we can see that the accuracies of three methods are all acceptable. The error level was low because no noise was added in the test data. However, the average and maximum distance errors of Huang’s method were highest compared with the proposed B-Surface Algorithm (BSA) and DFSA. It is because Huang’s method is designed for tubal topological object segmentation. It is difficult to segment and represent the apex of the target. And it is also sensitive to the size of target. When the radius of value of target decreases, the accuracy of BSF will decrease as well. Therefore, average and maximum errors at the apex of the target were much higher than BSA and DFSA. The advantage of Huang’s method is its ability to calculate the volume value of target efficiently by setting $dz = 0$ all the time. DFSA can achieve high accuracy in this experiment as well. Compared with the proposed BSA, the compression ratio was low, at 1:1 only. It is difficult for DFSA to compute differential quantities such as normal to surface at each surface point.

Table 3.1: Comparison of B-Surface algorithm with traditional methods in simulated object segmentation.

	BSA	DFSA	Huang’s Method
Average Distance Error	0.05 Voxel	0.05 Voxel	0.1 Voxel
Maximum Distance Error	1 Voxel	1 Voxel	3 Voxel
Compress Ratio	>20	1	-
Differential Quantities Computation	by formula	by approximation	by formula
Normal Error in Central Point	3°	-	-

3.5.2 Experiments on Segmentation of Brain Tumor

In this Section, the deformation steps of 3D B-Surface algorithm in the experiments on segmentation of human brain tumor are described. The experimental results and analysis are presented as well.

Threshold was first set to separate tumor and other organs briefly. Then, 3D edge extraction algorithm was carried out to extract the coarse boundary of the tumor using Equation (3.1). The result of coarse boundary extraction is shown in the right column in Figure 3.6. Based on the coarse edge map, the external forces field of the object surface is calculated using 3D GVF as described in Section 3.2.2. Two slices of the 3D external force field of the tumor we obtained are shown in Figure 3.7.

By setting control points, a B-Surface model near the boundary of the tumor in the 3D external force field was initialized. From Equation (3.10), an initial B-Surface model which consists of 28 B-spline surface patches was obtained. This is shown in Figure 3.8. Under the influence of the external forces, each initial B-spline surface patch was deformed in an iterative process to reach and stop at the surface of the tumor.

Using Equation (3.10), the object could be represented based on the experimental result. After the control point matrix of the surface is obtained, the 3D object could be represented easily with MatLab. The 3D model of the final result is shown in Figure 3.9. A smooth boundary of the tumor was obtained using the proposed B-Surface algorithm. Surface normal and curvature at any point on the surface can be calculated accurately based on the bi-cubic B-spline surface model. Only 448 control points were used to deform the surface model and represent the surface which consisted of over 5,000 surface points. Thus, the compression ratio was over 10.

To evaluate the experimental results, the obtained surface model was put back into the input image. The distances between 100 manually marked boundary points and the obtained surface were measured. The evaluation results are shown in Table

Figure 3.6: Original images and the coarse edges: (a1)-(a3) 3 slices of the original brain image; (b1)-(b3) the coarse edges of the tumor.

Figure 3.7: 3D external force field of the brain tumor. (a) X-Y plane with Z value at 45; (b) Y-Z plane with X value at 45.

Figure 3.8: Initial B-Surface model for brain tumor segmentation.

Figure 3.9: Experimental result on segmentation of brain tumor.

3.2. The maximum distance error was 1 voxel. The average distance error was 0.14 voxel. The normal to the result surface is not compared with the ground truth of the target which is not known in this real application.

A number of slices of the overlapped volume image in Figure 3.10 show that the 3D Deformable B-Surface model can find the surface of the object of interest successfully. When the boundary was not clear in the top part of the tumor, the proposed method could find the boundary properly. Representing the 3D result in 2D slice, the proposed method could successfully decide the width of the contour, which is difficult to judge even for human eyes.

In this experiment of brain tumor segmentation, a comparison has been carried out with DFSA and Huang’s method. The average and maximum distance errors of Huang’s method remained highest compared with the proposed BSA and DFSA. It is because the spherical topology of the brain tumor which is difficult for Huang’s method to segment and represent the apex of the target. Therefore, distance errors

Figure 3.10: Evaluation of the tumor segmentation result: (a1)-(a4) 4 slices of the original brain image; (b1)-(b4) the overlapping image of the boundary of the tumor with the original image.

of Huang’s method at the two apexes of the brain tumor were higher than BSA and DFSA. DFSA can achieve acceptable accuracy in this experiment as well. But it had low data compress ratio, only at 1:1. The difference of accuracy between DFSA and the proposed BSA is because the difference external force is chosen to guide the surface model to find the boundary of the brain tumor. In BSA, modified 3D GVF, which has a large capture range and is able to guide surface model to move into boundary concavities, was utilized to generate the external force field. It is difficult for DFSA to extract differential quantities such as normal to surface at each surface point. It can only approximate these differential quantities based on the surface points it obtained. The proposed B-Surface algorithm can extract these differential quantities accurately using surface normal and surface curvature calculation formula.

Table 3.2: Comparison of B-Surface algorithm with traditional methods in brain tumor segmentation.

	BSA	DFSA	Huang’s method
Average Distance Error	0.14 Voxel	0.5 Voxel	0.6 Voxel
Maximum Distance Error	1 Voxel	2 Voxel	3 Voxel
Compress Ratio	> 10	1	-
Differential Quantities Computation	by formula	by approximation	by formula

3.5.3 Experiments on Segmentation of Brain Ventricle

In this experiment, the proposed B-Surface algorithm was carried out in ventricle segmentation in brain image analysis. The horizontal and vertical view of the original MR image is shown in Figure 3.11.

Figure 3.11: Original MR images of human brain.

Figure 3.12: Edge maps of the MR image.

Threshold was first set to separate background and ventricle briefly. Then 3D edge extraction algorithm was carried out to extract the coarse boundary of the ventricle using Equation (3.1). The result of 3D coarse edge extraction algorithm is shown in Figure 3.12.

After the 3D edge detection algorithm was applied, the external force field of the object surface was calculated using 3D GVF as described in Section 3.2. The 3D external force field we obtained is shown in Figure 3.13. Figure 3.14 is the locally enlarged image of the 3D external force field of the brain ventricle. The arrow

Figure 3.13: 3D external force field of the brain ventricle.

denotes the external force vector at each position in the 3D space. From Figure 3.14, we can see that all the external force vectors point to the boundary of the target object. Thus, this external force field should guide the surface model to reach the boundary of the target. In some parts, the GVF force looks to be zero in one plane (x-y plane). But in another plane (x-z or y-z plane), the GVF force is not zero. The force is normalized in 3D space, which can be zero when it is projected to a 2D plane.

By setting control points, a B-Surface model which consisted of 120 B-spline surface patches was initialized near the boundary of the ventricle in the 3D external force field. From Equation (3.10), an initial B-Surface model which consisted of many B-spline surface patches was obtained. Under the influence of the external forces, each initial B-spline surface patch was deformed iteratively to reach and stop at the surface of the ventricle.

Using Equation (3.10), the object could be represented based on the experimental result. After the control point matrices of the surface are obtained, the 3D object can be represented easily with MatLab. The final result of B-Surface is shown in Figure 3.15. A smooth boundary of the brain ventricle was obtained using the proposed B-Surface algorithm. Surface normal and curvature at any point on the

Figure 3.14: The locally enlarged image of the 3D external force field of the brain ventricle.

surface can be calculated accurately based on the bi-cubic B-spline surface model. Only 1920 control points were used to deform the surface model and represent the surface which consisted of around 25,000 surface points. Thus, the compression ratio was over 10.

To evaluate B-Surface algorithm, the experimental result was put back into the 3D volume image. The overlapped images in Figure 3.16 show that the B-Surface method can find the surface of the object of interest successfully. When the boundary was not clear in the top part of the ventricle, the proposed method could find the boundary properly. Representing the 3D result in 2D slice, the proposed method could successfully decide the width of the contour which is difficult to judge even for human eyes. The segmentation accuracy was obtained by measuring the distance between 100 manually marked boundary points and the obtained surface. As shown in Table 3.3, the maximum distance error was 2 voxel. The average distance error was 0.31 voxel. The normal error was not calculated because it is hard compare the result surface with the ground truth of the target which is not known in this real application.

A comparison of the proposed method with the traditional 3D discrete deformable model DFSA and the B-spline surface based method Huang's method has been carried out in the experiment of brain ventricle segmentation. The average and maximum distance errors of the result are shown in Table 3.3. The maximum distance error of Huang's method remained the same level. The average distance error decreased slightly from 0.6 to 0.5. It is because brain ventricle is more like tubal topological object compared with brain tumors which is suitable for Huang's method to segment. Therefore, distance errors of Huang's method at the apex of the target remained higher than BSF and DFSA. At the tubal part of the target, distance errors of Huang's method decreased. The advantage of BSF is its ability to calculate the volume value of target efficiently by setting $dz = 0$ all the time. The accuracy of DFSA in this experiment was lower than the proposed BSA because different external forces were choose. Compared with the proposed BSA, the compress ration

Figure 3.15: Different view of the final result of the B-Surface in brain ventricle segmentation: (a) left view; (b) top view; (c) 3D view.

Figure 3.16: Evaluation of the result of B-Surface in brain ventricle segmentation. (a1) is the 75th slice of the original MR image; (a2) and (a3) are the 86th and 87th slices of the original MR image respectively; (b1) to (b3) are the overlapping images of the segmentation result with the original MR image.

of DFSA was at 1:1 only. It is difficult for DFSA to compute differential quantities such as normal to surface at each surface point. It approximates these differential quantities based on the surface points it obtained. The proposed B-Surface algorithm can extract these differential quantities of the target accurately using surface normal and surface curvature calculation formula.

Table 3.3: Comparison of B-Surface algorithm with the traditional methods in brain ventricle segmentation.

	BSA	DFSA	Huang’s method
Average Distance Error	0.31 Voxel	0.6 Voxel	0.5 Voxel
Maximum Distance Error	2 Voxel	3 Voxel	3 Voxel
Compress Ratio	> 10	1	-
Differential Quantities Computation	by formula	by approximation	by formula

3.6 Concluding Remarks

We have presented a novel algorithm for object segmentation in 3D volume image. The proposed algorithm uses a 3D edge extraction method to obtain a coarse boundary of the target. An improved 3D external force field combined with the normalized GVF is then computed. After the initialization of a surface model near the target, B-Surface starts the deformation steps to locate the boundary of the object. Firstly, it overcomes the difficulty that comes from analyzing 3D volume image slice by slice. Secondly, the speed of B-Surface deformation is enhanced since the internal forces are not needed to compute at every deformation iteration. Compared with traditional discrete 3D deformable surface models such as DFSA, it is easy for the proposed method to calculate the curvature and normal of the object surface because B-Surface is a continuous deformable model. It has the ability to achieve a high compression ratio by presenting the whole surface with only a relatively small number of control points. Compared with another B-spline surface model based 3D deformable model, the proposed method is not limited to tubal topological ob-

ject segmentation. The experimental results show that the proposed method can locate and represent the boundary of smooth 3D objects with high accuracy and data compression ratio. The proposed B-Surface model is suitable for simple object segmentation. Therefore, it will meet difficulty in segmenting object with more subtle structures because of the intricate boundaries. One of the reasons is a bi-cubic B-spline surface has strong implicit constraints. It will be difficult to segment and represent the object with subtle structures such as amygdale/hippocampus or superior temporal gyrus. One method is to reduce the B-spline surface patch size by decreasing the distance between the control points. But if the resolution of the source image is not high enough, it will not help to increase the accuracy in object segmentation for the proposed B-Surface model.

Chapter 4

A Novel Deformable Model for Segmentation of Complicated 3D Objects

4.1 Introduction

For the segmentation of complicated objects in medical image analysis, a priori knowledge from training data could be incorporated in deformable models. It is because a kind of organs often has a similar shape. And it can be utilized to generate statistical shape models for segmentation [147]. However, the complicated procedure for building a PDM from the training data is difficult, especially for 3D deformable models. And statistical deformable models are not flexible enough due to the limitation of the training sample. If the object for segmentation is not similar to any sample in the training set, the result will be bad because the strong parameter constraints will limit the template to find the correct boundary of the object. For example, the size, shape, location and rotation of a brain tumor vary greatly between different patients. It is very hard to train a PDM in this case. The shape of a brain ventricle, which is widely used to test statistical deformable

in recent published papers, will also vary significantly and unpredictably because of diseases. Statistical deformable models will meet great difficulty in the segmentation of objects with arbitrary shape variation.

Topologically adaptable deformable model is an efficient method to segment complicated objects. The term topology change may include both change in the mesh structure or change of the surface genus which transforms the surface such as it is not homeomorphic to its previous configuration. Besides 2D topologically adaptable active contours [138–140], there have been a lot of research on topologically adaptable deformable models [13, 14, 16, 141, 148, 149]. These 3D deformable surface models achieve topologically adaptable surface extraction at the price of high computational costs. For example, in Lachaud’s deformable meshes [13], the model changes its topology according to the classical Eulerian topological transformations of creation, deletion or inversion. At each deformation iteration, distances between vertices are computed to check the constraints for topological transformations. The total number of the vertices is often over tens of thousands in real applications.

The proposed Growing Deformable Surface Patches (GDSP) model is designed for the segmentation of complicated 3D objects from volume image. It achieves topologically adaptable surface extraction by connecting new surface patches with active patches and automated triangulating the square patch in particular situations. Compared with the existing topologically adaptable deformable surfaces, unnecessary computational costs are removed. Furthermore, surface curvature adaptiveness is achieved in the proposed method by associating the surface curvature with the size of the surface patches.

In Section 4.2, the procedure of GDSP model is presented. In Section 4.3, details are presented on how to associating the surface curvature with the size of the surface patches. A particular internal force for the growing mechanism is also presented. In Section 4.4, experimental results and analysis are given.

Figure 4.1: Each square surface patch, S_i , consists of 16 knots: P_{11} to P_{44} . P_{11} , P_{14} , P_{41} and P_{44} are the four vertices of the square patch.

4.2 3D Topologically Adaptable Growing Deformable Surface Patches (GDSP) Model

4.2.1 Representation of the Model

Many deformable models are based on a triangular mesh surface. The proposed GDSP model is built up with square patches, as shown in Figure 4.1. A square patch is chosen because (1) features of the surface patch such as local curvature, surface normal, tangent direction of the patch edge can be calculated easily in the form of square patch (these features are important in the growing procedure of the proposed method); (2) it is convenient to represent surface model using square patches; (3) we can save data space because in most situations two triangular meshes (6 vertices in all) form one square patch (4 vertices in all); (4) we can triangulate square patches when necessary. It is shown in Figure 4.2.

In the proposed model $\mathbf{M}$, $\mathbf{M} \in \mathbb{R}^3$, the smallest element is a square surface patch S_i , which consists of four vertices: P_{i11} , P_{i14} , P_{i41} and P_{i44} , as shown in Figure 4.1. It is impossible to represent the entire close surface of an object without overlapping or

Figure 4.2: Triangulate a square patch by merging two adjacent vertices.

gap if only square patches are used. This problem is solved by merging two vertices to triangulate a square patch when the average curvature of the surface increases or decreases significantly. It is shown in Figure 4.3. In the situation that surface patches come to a pole of the object when the average curvature of the surface does not change, a square patch is often triangulated as well. This situation is shown in Figure 4.4.

The final result of the proposed model, M , is a closed surface which consists of a number of surface patches S_i (square patches or triangulated patches). Each surface patch is connected with nearby patches, sharing the same edges of the connected patches. The difference between other methods and the proposed method is the growing mechanism, rather than difference of the mesh type used in the surface model.

Figure 4.3: Triangulate a square patch by merging two vertices when the average curvature of the surface changes significantly

Figure 4.4: Triangulate a square patch by merging two vertices when surface patches come to a pole of the object to generate a closed surface

4.2.2 Topologically Adaptable Growing Mechanism

In traditional deformable surface models, a surface model is initialized first. Commonly, this initial surface model is a closed surface model with spherical topology. Under the influence of the internal and external forces, the model is deformed to search the boundary of the object of interest. In each deformation step, every element of the surface model is deformed regardless whether it has reached the boundary of the object. Therefore, the internal and external forces on the elements which have reached the boundary are still calculated in each deformation step. These computations are useless actually. Furthermore, at each deformation iteration, distances between vertices whose total number is often over tens of thousands in real applications have to be computed to check the constraints for topological transformations because the topological transformations may occur at any position in the model. Actually, most of the computation is not necessary because topological transformations only occur locally and limitedly at each deformation iteration.

The proposed algorithm includes two iteration loops (shown in Figure 4.5). The inner loop for the deformation of each surface patch and outer loop for the growth of the entire surface model. The inner loop is to deform surface patches separately with the help of particular internal force which is designed to support deformation with “anchored” edge (AE). The details are presented in Section 4.3. The outer loop, based on the growth mechanism, is designed to achieve topologically adaptable object segmentation.

In the surface model $\mathbf{M}$, any two connected patches, $\mathbf{S}_i, \mathbf{S}_j, i \neq j$, will share a common edge. This edge will be set as “connected” edge. The bare edge which is free and ready for connecting with other patch is labelled as “bare” edge (BE). If a surface patch $\mathbf{S}_i$ has BEs, it is labelled as an “active” surface patch (ASP) and put into ASP pool $\mathbf{M}_{atv}$. If a surface patch $\mathbf{S}_i$ does not have BE, it will be taken away from the ASP pool.

In the outer loop, a new surface patch $\mathbf{S}_{i+1}$ is generated from the ASP pool $\mathbf{M}_{atv,i}$

Figure 4.5: Flowchart of the GDSP model.

in current surface model $\mathbf{M}_i$ and the participant edges are relabelled. After each inner loop, a single surface patch $\mathbf{S}_i$ is deformed to reach the boundary of the object. Surface model $\mathbf{M}_i$ and ASP pool $\mathbf{M}_{atv,i}$ are updated after the end of every inner loop. An ASP $\mathbf{S}$ is selected from the current ASP pool $\mathbf{M}_{atv,i}$ to initialize a new surface patch, $\mathbf{S}_{i+1}$, for the inner loop at the next iteration.

The criteria for termination of deformation loop for each surface patch is similar to traditional deformable models. When the total energy on a surface patch reaches the minimum value, the deformation procedure for this surface patch terminates. The proposed model will stop if the criteria for termination of the growth loop is satisfied, i.e., $\mathbf{M}_{atv,i} \subset \mathbf{M}_i$ is null. It means there is no more ASP in the surface model. The final result of the proposed model, $\mathbf{M}$, is a closed surface which consists of a number of surface patches $\mathbf{S}_i$ (square patches or triangulated patches). Each surface patch is connected with nearby patches, sharing the same edges of the connected patches.

In the deformation procedure of the proposed method, once a surface patch is “anchored”, it will not participate in the deformation step. If it still has edges with BE label, it will provide geometric information for being connected with other patches, but neither internal nor external forces are calculated for this surface patch because it is anchored and will not move any more; if it has no BE, this result is just recorded and taken away from the ASP pool $\mathbf{M}_{atv}$ because it could not provide any information for subsequent deformation steps. In this way, we remove the useless computational cost on the stable elements (it is the part of the surface model that reaches the boundary of the object first) of the surface model. The computational cost is reduced by dividing the entire closed surface into small surface patches. At the same time, the procedure for surface model initialization which is always a difficult problem for deformable models is simplified. Each new surface patch is guaranteed to be very near to the boundary of the object because it is generated base on the “anchored” surface patches (ANSP) which have already found the boundary. Therefore, the existing geometric information is utilized in the new surface patch initialization step. Thus, the iteration number and the computational cost of the

Figure 4.6: Surface patch growth: type 1.

deformation for each surface patch are reduced significantly.

In the proposed algorithm, a new surface patch $\mathbf{S}$ utilizes the geometric information of the existing result directly by growing up from the ASP pool $\mathbf{M}_{atv}$. Surface patches model $\mathbf{M}$ grows up along the boundary of the object of interest. The topological adaptability is achieved in the growing procedure naturally without additional topological transformation such as melting or splitting. A new surface patch will connect to the ASP $\mathbf{S} \in \mathbf{M}_{atv}$ if they are close enough to each other and satisfy other conditions. The details are given in Section 4.2.3. For those patches without a BE, all of the edges of them have been utilized to connected with nearby patches, they will not be connected to any other new patches since we assume that there is no intersection occasion.

4.2.3 Implementation of the Growth of Surface Patches

In the surface model $\mathbf{M}$, each surface patch $\mathbf{S}$ has four edges. In the triangulated surface patch, there are four edges as well. One of the edge is degenerated to a vertex (Figure 4.10). A new surface patch $\mathbf{S}_i$ is generated from the ASP pool $\mathbf{M}_{atv}$. Therefore, it is connected with these surfaces patches. Basically, growth of surface patches can be classified into 7 types, shown in Figure 4.6 to Figure 4.12.

In type 1, the BE l (the dashed edge shown in Figure 4.6) on the top of the ANSP is selected to generate a new surface patch. As shown in Figure 4.7, β is the angle

Figure 4.7: Surface patch growth: type 2.

Figure 4.8: Surface patch growth: type 3.

between the selected BE and the connected BE in other surface patches. The right hatched surface patch is generated from the selected ANSP. In this case, the selected BE does not have a connected BE or the angle β is more than 120° . This edge will be utilized in generating the new surface patch as the common edge between these two patches, then labelled as “connected”. Therefore, the new surface patch has three BEs, and the number of BE of the selected ANSP is reduced by one.

In type 2, the two dashed edges in the left two selected ANSPs shown in Figure 4.7, l_1 and l_2 , are the participant BEs for the growth of the new surface patch. The right hatched surface patch is generated from the two ANSPs. β is the angle between the two BEs l_1 and l_2 . The angle β is more than 60° and less than 120° . Then l_1 and l_2 are utilized as the common edges in generating the new surface patch and then labelled as “connected”. The new surface patch has two BEs and the number of BE of the two selected ANSPs is reduced by one respectively.

Type 3 is similar to type 2. Three dashed lines, l_1 , l_2 and l_3 , in the left three ANSPs

Figure 4.9: Surface patch growth: type 4.

shown in Figure 4.8 are the participant BEs for the growth of the new surface patch. The right hatched surface patch is generated from the three ANSPs. β is the angle between the two BEs. The angle β_1 and β_2 is more than 60° and less than 120° both. Then the three connective BEs l_1, l_2 and l_3 are utilized as the common edges in generating the new surface patch and then labelled as “connected”. The new surface patch has one BE and the number of BE of the two selected ANSPs is reduced by one respectively.

In type 4, four BEs l_1, l_2, l_3 and l_4 , shown in Figure 4.9, are connective to each other and form a empty square surface patch. In the growth procedure, l_1, l_2, l_3 and l_4 are utilized in generating the new surface patch and then labelled as “connected”. All the edges of the new surface patch are labelled as “connected” and the number of BE of the four selected ANSPs is reduced by one respectively. In this case, the new generated surface patch will be recorded after the deformation procedure for it and taken away from the ASP pool since the new generated surface patch does not have any BE.

Surface patch growth type 1 to type 4 are designed for surface patch generation in the smooth portion of the object. We use surface patch growth type 5 to generate triangulated patch when the average curvature of the surface increases or decreases significantly or in the situation that surface patches come to a pole of the object

Figure 4.10: Surface patch growth: type 5.

Figure 4.11: Surface patch growth: type 6.

to generate a closed surface although the average curvature of the surface does not change significantly. The two dashed edges in the left two selected ANSPs shown in Figure 4.10, l_1 and l_2 , are the participant BEs for the growth of the new surface patch. In this case, β , the angle between the selected BE l_1 and its connecting BE l_2 , is less than 60° . The two connecting BEs l_1 and l_2 are utilized in generating the triangulated surface patch. One edge of the new surface patch is degraded to a knot that is the conjunction of l_1 and l_2 .

Surface patch growth type 6 and type 7 are designed for connecting the current patch with opposite ASPs when two edges of the surface come close enough to each other. In type 6, the two vertices of the BE l_1 are opposite to the vertices of l_2 in another ASP as shown in Figure 4.11. The distance between the two centers of the BEs is less than $1.2 \times d_w$. d_w is the desired mean edge length of surface patches.

Figure 4.12: Surface patch growth: type 7.

The two opposite BEs l_1 and l_2 are utilized in generating the new surface patch. The number of BE of the two selected ANSPs is reduced by one respectively. The new surface patch has two BEs.

In type 7, as shown in Figure 4.12, the selected BE l_1 only have one opposite vertex. The distance between the two opposite vertices is less than $1.2 \times d_w$. The vertex and l_1 are utilized in generating the new surface patch. The number of BE of the selected ANSP is reduced by one and the number of BE of the opposite surface patch remains the same. The number of BE of the new surface patch is three.

All these seven types of growth of surface patch and their combinations ensure that a closed surface model can be obtained.

4.3 Deformation of Surface Patches for Object Segmentation

In GDSP model, each surface patch $\mathbf{S}_i$ is deformed by minimizing the total energy

$$E(\mathbf{S}_i) = E_{int}(\mathbf{S}_i) + E_{ext}(\mathbf{S}_i) \quad (4.1)$$

As growing mechanism is introduced for topologically adaptable object segmentation and surface extraction, specific internal force is designed to support the deformation of surface patches with growing mechanism. Each surface patch $\mathbf{S}_i, \mathbf{S}_i = (x(t), y(t), z(t))$ is deformed separately, but not independently. It is because the new generated surface patch has at least one AE which is shared with the connect surface patch. The AEs of the new surface patch will not move in the deformation process of this patch.

4.3.1 Internal Forces

We define three internal forces that depend on the surface patch itself and the mean size of the surface patch which is a global parameter.

$$E_{int} = E_{f_c} + E_{f_e} + E_{f_a} \quad (4.2)$$

The first term is a traditional internal force which smoothes the shape of the surface patch. The second term is another internal force which spreads localized deformations and achieves local curvature adaptiveness. The third term is designed to support deformation procedure with AEs. Here we choose the mean edge length d_w to constrain the size of the surface patches. If $U(x, y, z)$ is a knot of the new surface patch, $\mathbf{u}$ expresses its coordinates, $\bar{\mathbf{u}}$ is the mid-point of the nearby knots of U in the new surface patch.

The first internal force is f_c which smoothes the shape of the surface patch. It brings back knots to their local tangent plane and minimizes surface curvature.

$$\forall U \in \mathbf{S}_i, f_c(U) = \alpha_c(\bar{\mathbf{u}} - \mathbf{u}) \quad (4.3)$$

where α_c is the rigidity coefficient.

The second internal force is f_e which spreads localized deformations and makes the

Figure 4.13: (a) θ is near 180° if the surface of the object is simple; (b) θ decreases when the surface of the object is complicated.

surface patch adaptive to the local curvature. It regularizes the edge lengths along the whole surface and expresses the binding energy. The local curvature of the model will also influence f_e . θ , the angle between two edges of adjacent knots, is near its maximum 180° if the surface of the object is simple, shown in Figure 4.13 (a). When the surface of the object is complicated, its radius of curvature decreases significantly, θ will decrease as well.

Different objects, even different portions of the same object, have different levels of complexity. To make the proposed method adaptive to the complexity of the object, we adjust the size of the individual surface patch with respect to the local curvature by changing the width of the surface patch. Thus, f_e is defined as:

$$\forall U \in \mathbf{S}_i, f_e(U) = \alpha_e \sum_{V \in \mathbf{S}_i, V \neq U} \frac{\mathbf{v} - \mathbf{u}}{\|\mathbf{v} - \mathbf{u}\|} \left(\|\mathbf{v} - \mathbf{u}\| - \frac{d_a}{3} \right) \quad (4.4)$$

where

$$d_a = \frac{1}{2}d_w(1 - \cos\theta)$$

where α_e is the stiffness coefficient, d_w is the desired mean edge length of surface patches, V is a knot in $\mathbf{S}_i$.

When θ decreases, the width of the new generated surface patch d_a will decrease also. It comes from the common sense. In the complicated portion of the object, in order to obtain more details, the size of the surface patch should be reduced according to the increase of local curvature; in the simple portion of the object, the size of the surface patch is enlarged to the desired size to speed up the procedure by reducing the number of surface patches.

The third internal force is f_a which is design to support deformation with AEs. When a new surface patch is generated for deformation, there are two kinds of knots among all the 16 knots in each surface patch. $\mathbf{S}_{ia}$ is a set of “anchored” knots (AK) lie in AEs and $\mathbf{S}_{if}$ is a set of the rest “free” knots. On AKs, the magnitude of f_a equals to the sum of f_c and f_e , but its direction is opposite to the sum of f_c and f_e . Therefore, the sum of internal forces on AKs remains zero. The f_a on “free” knots is relevant to the f_a on nearby AKs with respect to the distance between these two knots. The influence of $f_a(V), V \in \mathbf{S}_{ia}$ to $f_a(U), U \in \mathbf{S}_{if}$ decreases when the distance $\|V - U\|$ increases.

$$\left. \begin{aligned} \forall U \in \mathbf{S}_{ia}, \quad f_a(U) &= -f_c(U) - f_e(U), \\ \forall U \in \mathbf{S}_{if}, \quad f_a(U) &= \alpha_a \sum_{V \in \mathbf{S}_{ia}} \frac{f_a(V)}{\|V - U\|}, \end{aligned} \right\} \quad (4.5)$$

where α_a is the response coefficient.

4.3.2 External Force

The function of external force field $\mathbf{v}, \mathbf{v} \in \mathbb{R}^3$ in 3D GDSP model is to guide the proposed surface model to move towards the boundaries of the target object. We put the new generated surface patch $\mathbf{S}_i$ in the force field $\mathbf{v}$ first. Every point $U, U \in \mathbf{S}_i$ is influenced by the force field. The external force on point U is:

$$\forall U \in \mathbf{S}_i, f_{ext}(U) = \mathbf{v}(U) \quad (4.6)$$

The improved gradient vector flow (GVF) field is chosen as the external force field which has a large capture range and is able to move the surface model into boundary concavities. The way to calculate the external force field of objects of interest has been described in Chapter 3.

In order to support deformation with AEs, particular internal forces are introduced to ensure the total internal forces on the AKs remaining zero. However no extra external force is introduced to set the total external forces on the AKs as zero to stop its movement. It is because the AKs in the new surface patch lie in the AE which is shared with other deformed surface patches. The external forces on these knots has already achieved minimum in the deformation procedure of the ANSP.

4.4 Experimental Results and Discussion

In this Section, the deformation steps of 3D GDSP model are described. Experimental results and discussion are presented as well. In each experiment, an external force field of the target was calculated in the first step. A single surface patch was initialized near the boundary of the target, and an ASP pool was built at the same time. In the inner loop of the proposed method, surface patches were deformed to locate the boundary of the target. The ASP pool was updated as well. In the outer loop, new surface patches were generated based on the ASPs. If there was no more ASP, a closed surface of the target was obtained. The flowchart is shown in Figure 4.5.

4.4.1 Experiments on Segmentation of Torus

Simulated experiments were carried out first to demonstrate that the proposed method could achieve topologically adaptable object segmentation.

A torus which is a typical test object for topologically adaptable deformable model was chosen as the object to segment in this experiment. The simulated volume image size was $256 \times 256 \times 256$. The size of the torus was set as $R = 48$, $r = 16$, as shown in Figure 4.14.

The model parameters were: $d_w = 9$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. In the first stage, four surface patches were initialized in the external force field of the torus. They were deformed to find part of the boundary of the torus then “anchored” at there. New surface patches were generated based on the ANSPs in the subsequent steps, growing along the surface of the torus step by step. The ANSPs will not move in the deformation procedure. If an old ANSP has BEs, it is still “active” because it will be connected with other surface patches in the subsequent growth procedure. When a surface patch does not has a BE, it means all of its four edges have been utilized to connect with other surface patches.

Figure 4.14: Simulated torus for segmentation.

Then it is taken away from the ASP pool. Until all the edges of the existing surface patches are labelled as “connected” , a closed surface model is obtained since there is no ASP.

The surface evolution in the experiment of torus segmentation is shown in Figure 4.15. The initial open surface model grew along the surface of the torus step by step and generate a closed surface model finally. Because it was a simulated torus, the boundary of the target was known. Thus, the accuracy of the proposed method was obtained by comparing segmentation result with the ground truth. The results shown in Table 4.1 indicate that the proposed method can achieve high accuracy because classical method was utilized in the deformation step for each surface patch of the entire model. The internal forces and external force guided each surface patch to find the correct boundary of the object. The average distance error of the proposed method in this experiment was 0.12 voxel. The maximum distance error was 1 voxel.

Figure 4.16 shows the change of the number of the ASPs as the iteration number increases during the deformation procedure. The total number of surface patches of the final result was 473. The number of the ASPs increased at the beginning, then it reached the peak at around 80 and started to decrease. At last, the number of

Figure 4.15: Experimental result of torus segmentation without noise: (a) result with 100 surface patches; (b) result with 150 surface patches; (c) result with 200 Surface Patches; (d) result with 300 surface patches; (e) result with 350 Surface Patches; (f) result with 473 surface patches.

Table 4.1: Experimental results on segmentation of torus.

Maximum Number of ASPs	79
Number of Surface Patches	473
Number of Surface Points	7568
Initial Patch Width	9 voxel
Average Distance Error	0.12 voxel
Maximum Distance Error	1 voxel

the ASPs reduced to zero and the deformation procedure ended.

Discussion

In this experiment, there were 7568 surface points in the final closed surface model. Because the maximum number of ASPs was only 80, it means in order to achieve topologically adaptable object segmentation, only the position relationship among 80 or less ASPs was considered during the whole deformation process. If using traditional topologically adaptable deformable surface models [13], the distances between two knots among the 7568 vertices were calculated to check the constraints for topological transformations at each iteration. In this experiment the ratio of the ASP number to the total vertices number was 0.010. Thus, the computational cost of the proposed method was reduced significantly compared with other methods.

4.4.2 Experiments on Segmentation of Human Ventricle

In the second experiment, the ventricle was segmented from a T2 weighted MR image of human brain. The volume image size was $256 \times 256 \times 138$.

The model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. Compared with the former experiment, the desired mean edge length of surface patches d_w was set with a smaller value. It is because in real applications, targets are often more complicated than a smooth torus. Therefore, d_w was set a small value to prevent losing details.

Figure 4.16: Change of the number of the ASPs as the iteration number increases during the deformation procedure in torus segmentation.

Figure 4.17 shows the growing steps of the deformable model. The surface evolution process of the proposed method shows that after initialize a surface patch near the boundary of the target, the unclosed surface model will grow along the boundary of the target. New surface patches were generated from the ASPs in the outer loop of the algorithm. Each surface patch was deformed and the ASP pool was updated in the inner loop. This process was repeated until a closed surface of the object was obtained. Different views of the final result final result of the ventricle segmentation is shown in Figure 4.18.

To evaluate the proposed algorithm, the surface model obtained in this experiment was put back into the 3D volume image. The distance error of the proposed method was evaluated by measuring the distance between 100 manually marked boundary points and the obtained surface. The performance of the proposed method is shown in Table 4.2. The total number of surface points was 11440 and the total number of surface patches was 715. The maximum number of ASPs was 42. In this ex-

Figure 4.17: The growing steps of the proposed deformable model in human brain ventricle segmentation: (a) result with 100 surface patches; (b) result with 200 surface patches; (c) result with 300 surface patches; (d) result with 400 surface patches; (e) result with 500 surface patches; (f) result with 600 surface patches.

Figure 4.18: Different views of the final result of human brain ventricle segmentation: (a) top view; (b) right view; (c) 3D view.

periment, the average distance error of the proposed method was 0.41 voxel. The maximum distance error was 3 voxel, reaching at the tip of the ventricle. The results indicate that the proposed method can achieve high accuracy in complicated object segmentation in the real application.

Table 4.2: Experimental results on segmentation of human ventricle.

Maximum Number of ASPs	42
Number of Surface Patches	715
Number of Surface Points	11440
Initial Patch Width	6 voxel
Number of Manual Selected Marks	100
Average Distance Error	0.41 voxel
Maximum Distance Error	3 voxel

Figure 4.19 shows the change of the number of the ASPs as the iteration number increases during the deformation procedure in this brain ventricle segmentation experiment. The total number of surface patches of the final result was 715. As

Figure 4.19: Change of the number of the ASPs as the iteration number increases during the deformation procedure in brain ventricle segmentation.

the total number of surface patches increases, the number of the ASPs increased and reached its peak at 42 and started to decrease. At last, the deformation step terminates when the number of the ASPs reduced to zero which means a closed surface model was generated and there was no more surface patch with BEs.

Discussion

In the experiment on brain ventricle segmentation, the target was much more complicated compared with the simulated torus. The total number of surface points was 11440 and the total number of surface patches was 715. The maximum number of ASPs was only 42. It means topologically adaptable object segmentation was achieved by considering the position relationship among 42 or less ASPs during the whole deformation process. If using traditional topologically adaptable deformable surface models [13], the distance between two knots among the 11440 vertices will be calculated to check the constraints for topological transformations at each itera-

tion. In this experiment, the ratio of the ASP number to the total vertices number was 0.0037. Thus, the computational cost of the proposed method was reduced significantly compared with other topologically adaptable deformable models.

4.4.3 Experiments on Segmentation of Human Vertebra

In the third experiment, the human vertebra was segmented from 3D volume image. The original volume image was interpolated to the size of $128 \times 128 \times 128$.

The model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. The desired mean edge length of surface patches d_w was also set as 6 voxel to obtain more details of the target.

The growing steps of the proposed deformable model is presented in Figure 4.20. First, a surface patch was initialized near the boundary of the target. In the inner loop of deformation procedure, this initial surface patch was deformed to locate the boundary of the vertebra. Based on the ASP pool, new surface patches were generated along the boundary of the target and deformed in the outer loop of deformation procedure. This process was repeated until a closed surface model of the human vertebra was obtained. The final result of the vertebra segmentation is shown in Figure 4.21.

To evaluate the proposed algorithm, the surface model obtained in this experiment was put back into the 3D volume image. The distance error of the proposed method was evaluated by measuring the distance between 100 manually marked boundary points and the obtained surface. The performance of the proposed method in human vertebra is shown in Table 4.3. The total number of surface points was 20208. The total number of surface patches of the final result was 1263. The maximum number of the ASPs was 106. The average distance error was 0.44 voxel. The maximum distance error reached 3 voxels at the sharp corner of vertebra. Compared with the target in former experiment, human vertebra is much more complicated than human brain ventricle. The proposed method can obtain good experimental results

Figure 4.20: The growing steps of the proposed deformable model in human vertebra segmentation: (a) result with 200 surface patches; (b) result with 400 surface patches; (c) result with 600 surface patches; (d) result with 800 surface patches; (e) result with 900 surface patches; (f) result with 1000 surface patches.

Figure 4.21: Different view of the final result of human vertebra segmentation: (a) left view; (b) top view; (c) 3D view.

as well. The results indicate that the proposed method can achieve high accuracy in segmenting object with complicated topology in the real application.

Table 4.3: Experimental results on segmentation of human vertebra.

Maximum Number of ASPs	106
Number of Surface Patches	1263
Number of Surface Points	20208
Initial Patch Width	6 voxel
Number of Manual Selected Marks	100
Average Distance Error	0.44 voxel
Maximum Distance Error	3 voxel

Figure 4.22 shows the change of the number of the ASPs as the iteration number increased during the deformation procedure in this vertebra segmentation experiment. The total number of surface points was 20208. As the iteration number increased, the number of the ASPs increased and reached its peak to 106 and then started to decrease. At last, the deformation step terminated when the number of the ASPs reduced to zero which means a closed surface model was obtained and there was no more surface patch with BEs.

Figure 4.22: Change of the number of the ASPs as the iteration number increases during the deformation procedure in human vertebra segmentation.

Discussion

In this experiment on human vertebra segmentation, the target was complicated. It had fingers, a tunnel, and a handle [150].

A surface patch was the smallest element in the proposed method. Thus, in the deformation procedure, we only considered the position relationship among 106 or less surface patches to achieve topologically adaptable object segmentation and surface extraction. If using the mechanism of traditional topologically adaptable deformable models [13], topology transformation check must be carried out among all 20208 surface points, the distance between two knots among all the vertices will be calculated at each iteration. In this experiment, the ratio of the ASP number to the total vertices number was 0.0052. Thus, much computational cost was reduced in the proposed method compared with traditional methods with topologically adaptable ability.

4.5 Concluding Remarks

We have presented a novel method, GDSP model, for the segmentation of complicated objects in 3D volume image. A growing mechanism is introduced in the deformable models to achieve topologically adaptable object segmentation. Only a surface patch, rather than a whole surface model is initialized. Therefore, the surface model initialization step is simplified. This method overcomes the limitations of traditional explicit deformable models. It achieves topologically adaptable surface extraction by connecting new surface patches with ASPs and automated triangulating the square patch in particular situations. Compared with the existing topologically adaptable deformable models, no splitting or merging judgement is carried out among all the vertices at each deformation iteration. Thus, the computational costs are reduced significantly. The proposed method is explicit. Therefore, it is convenient to provide geometrical information such as area, volume, or local curvature of the object. Furthermore, surface curvature adaptiveness is achieved in the proposed deformable model by associating the surface curvature with the size of the surface patches.

Chapter 5

Development of a Robust Algorithm for 3D Object Segmentation

5.1 Introduction

As presented in Chapter 4, the proposed GDSP model is a general deformable surface model designed to achieve flexible object segmentation. The critical and time consuming training procedure for 3D statistical deformable models is not necessary for the proposed method. Although it will be more sensitive to noise or missing data as compared with the well trained statistical deformable models such as 3D ASM, it can segment and represent objects with arbitrary shape variation and complex topology. The unique GDSP model can be utilized in the segmentation of brain tumor, brain ventricle, vertebra or other organs in 3D MR or CT images.

However, the deformation procedure may not work properly when the initialized surface patch is parallel to the direction of the external force vector in 3D space. It is because the knots of the surface patch will come close to each other under the

influence of the external force. At the same time, the traditional internal force works against the external force in this situation. The combined forces cannot let the knots converge to the actual boundary of the object quickly. To solve this problem, an extra internal force is introduced in the proposed method. A relationship between internal force and external force will be utilized in deformable model. This novel internal force could let the deforming surface patch move perpendicularly to the external force vector.

In the past years, many approaches were introduced in traditional 2D or 3D deformable models in order to increase their performance, such as statistical deformable models, attribute matching mechanism, *etc.* Prior knowledge of object is not suitable to incorporate in the proposed method in order to keep the flexible ability. We can also add some other properties to GDSP model. A popular idea is coarse-to-fine or multi-scale deformable model. In order to speed up the process, besides to avoid being trapped by local minima, a hierarchical mechanism is introduced in the proposed robust GDSP model. After a fast estimation of the object in low resolution, the model searches for finer details while increasing the resolution level.

An efficient algorithm for surface area calculation of objects of interest was developed in this Chapter as well. In the explicit robust GDSP model, the smallest element is surface patch. The surface area of the entire model is obtained by summarizing areas of all the surface patches. The experimental results show that it can calculate the surface area of the target with high accuracy.

In Section 5.2, the procedure of robust GDSP model is presented. A novel internal force is developed to remove the sensitivity to initialization. A hierarchical framework is incorporated to reduce local minima. Calculation of area is presented as well. In Section 5.3, experimental results and analysis on the novel internal force, hierarchical framework and area calculation are given.

5.2 Robust Growing Deformable Surface Patches (GDSP) Model

In the proposed robust GDSP model, a novel internal force is developed to remove the sensitivity to model initialization. A hierarchical framework is incorporated to reduce local minima and speed up the algorithm. Based on the segmentation result, an efficient algorithm for surface area calculation of objects of interest was developed as well.

5.2.1 Inclusion of a New Internal Force

The deformation procedure may not work properly when the initialized surface patch is parallel to the direction of the external force vector in 3D space. Knots are forced to move along the direction of the external force. The knots of the surface patch will come close to each other under the influence of the external force. At the same time, the traditional internal force, which tends to spread the knots to reach a desired distance d_0 between each other, works against the external force in this situation. This is shown in Figure 5.1 where "●" is the knot of the surface model, E_{ext} is the external force field, F_{int} is the internal force on each knot, d_0 is the desired distance between each knot. But it cannot let the knots converge to the actual boundary of the object quickly.

To solve this problem, an extra internal force is introduced in the proposed method. This novel internal force could let the deforming surface patch move perpendicularly to the external force vector. This is shown in Figure 5.2. A positive normal of the current deforming surface patch will first be calculated. The difference between this normal and the direction of the external force vector will be a factor of the internal force. Because in the ideal condition, the positive normal of the current deforming surface patch will be the same with the direction of the external force vector. In this way, a relationship between internal force and external force will be utilized in

Figure 5.1: The traditional external and internal forces fail to let the knots converge to the actual boundary of the object.

deformable model.

First, normal to a surface patch in deformation process will be calculated. To any point of a surface S : $\mathbf{x}(u, v)$, we can associate the two vectors

$$\mathbf{x}_u \equiv \frac{\partial \mathbf{x}}{\partial u}, \quad (5.1)$$

and

$$\mathbf{x}_v \equiv \frac{\partial \mathbf{x}}{\partial v}. \quad (5.2)$$

At any regular point P of S these vectors are linearly independent and are tangent to the coordinate curves through P ; they span the tangent plane $E(P)$ to S at P .

The unit vector $\mathbf{n}$ which is orthogonal to those vectors and whose sense is such that $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{n}$ form a right-handed triple of vectors, is called the unit normal vector

Figure 5.2: The additional internal force rotates the surface patch if the initialized surface patch is parallel to the external force vector.

to S at P . This is shown in Figure 5.3.

The straight line through P in the direction of $\mathbf{n}$ is called the normal to the surface S at the point P .

Thus, we can calculate the normal to the surface S at the point P

$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|} \quad (5.3)$$

In the real application, we do not calculate the normal to the surface patch S at every point. Only the normal $\mathbf{n}$ to the surface patch S at the central point is calculated. The difference between n and the average direction of the external force vectors on the surface patch is used as a feedback in the deformation procedure to rotate the surface patch, making it move perpendicularly to the direction of the external force field.

Based on the surface patch model we used, shown in Figure 4.1, we can calculate the normal $\mathbf{n}$ to the surface patch S at the central point as follows:

Figure 5.3: The unit vector $\mathbf{n}$ which is orthogonal to those vectors and whose sense is such that $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{n}$ form a right-handed triple of vectors, is called the unit normal vector to S at P .

$$\begin{aligned}
 t_{u22,21} &= \frac{P_{22}-P_{21}}{|P_{22}-P_{21}|} \\
 t_{u23,22} &= \frac{P_{23}-P_{22}}{|P_{23}-P_{22}|} \\
 t_{u24,23} &= \frac{P_{24}-P_{23}}{|P_{24}-P_{23}|} \\
 t_{u32,31} &= \frac{P_{32}-P_{31}}{|P_{32}-P_{31}|} \\
 t_{u33,32} &= \frac{P_{33}-P_{32}}{|P_{33}-P_{32}|} \\
 t_{u34,33} &= \frac{P_{34}-P_{33}}{|P_{34}-P_{33}|}
 \end{aligned} \tag{5.4}$$

$$\begin{aligned}
 t_{v22,12} &= \frac{P_{22}-P_{12}}{|P_{22}-P_{12}|} \\
 t_{v32,22} &= \frac{P_{32}-P_{22}}{|P_{32}-P_{22}|} \\
 t_{v42,32} &= \frac{P_{42}-P_{32}}{|P_{42}-P_{32}|} \\
 t_{v23,13} &= \frac{P_{23}-P_{13}}{|P_{23}-P_{13}|} \\
 t_{v33,23} &= \frac{P_{33}-P_{23}}{|P_{33}-P_{23}|} \\
 t_{v43,33} &= \frac{P_{43}-P_{33}}{|P_{43}-P_{33}|}
 \end{aligned} \tag{5.5}$$

$$\mathbf{n}_{22} = (t_{u22,21} + t_{u23,22}) \times (t_{v22,12} + t_{v32,22}) \tag{5.6}$$

$$\mathbf{n}_{23} = (\mathbf{t}_{u23,22} + \mathbf{t}_{u24,23}) \times (\mathbf{t}_{v23,13} + \mathbf{t}_{v33,23}) \quad (5.7)$$

$$\mathbf{n}_{32} = (\mathbf{t}_{u32,31} + \mathbf{t}_{u33,32}) \times (\mathbf{t}_{v32,22} + \mathbf{t}_{v42,32}) \quad (5.8)$$

$$\mathbf{n}_{33} = (\mathbf{t}_{u33,22} + \mathbf{t}_{u34,33}) \times (\mathbf{t}_{v33,23} + \mathbf{t}_{v43,33}) \quad (5.9)$$

$$\mathbf{n}_{i,j} = \frac{\mathbf{n}_{i,j}}{|\mathbf{n}_{i,j}|}, \quad i, j = 2, 3. \quad (5.10)$$

We use the average of these four normals as the normal to the surface patch at the central point. This is shown in Figure 5.4.

$$\mathbf{n} = \sum_{i=2}^3 \sum_{j=2}^3 \frac{\mathbf{n}_{i,j}}{4}. \quad (5.11)$$

α , the difference between the direction of the normal $\mathbf{n}$ and the external force $\mathbf{F}_{ext}$ shown in Figure 5.5, is calculated

$$\alpha = \mathbf{n} - |\mathbf{F}_{ext}|, \quad (5.12)$$

where $\mathbf{F}_{ext}$ is the external force on central point of the surface patch S .

α works as a feedback to let the surface patch move perpendicularly to the external force field. The normal to the surface patch at the central point should have the same direction as the external force at that position. Therefore, in the deformation procedure, α is minimized

$$\hat{\alpha} \Rightarrow 0, \quad (5.13)$$

Figure 5.4: The unit vector $\mathbf{n}_{22}$, $\mathbf{n}_{23}$, $\mathbf{n}_{32}$ and $\mathbf{n}_{33}$ are normals to the surface patch at point P_{22} , P_{23} , P_{32} and P_{33} .

Figure 5.5: α is the difference between the direction of the normal and the external force.

Thus,

$$\begin{aligned}\hat{\mathbf{n}} &= \mathbf{F}_{ext} - \hat{\alpha} \\ &= \mathbf{F}_{ext},\end{aligned}\tag{5.14}$$

Then, a surface plane $\hat{S}$ is build using the unit external force vector $|\mathbf{F}_{ext}|$ at the central point as the normal $\hat{\mathbf{n}}$. Rotate the surface patch S to match $\hat{S}$. The surface patch S is rotated as the sequence as Z axis rotation, Y axis rotation and finally X axis rotation. Therefore, α is separated in three components: α_z , α_y and α_x .

For Z axis rotation,

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} = \begin{bmatrix} \cos \alpha_z & -\sin \alpha_z & 0 \\ \sin \alpha_z & \cos \alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}\tag{5.15}$$

For Y axis rotation,

$$\begin{bmatrix} P''_x \\ P''_y \\ P''_z \end{bmatrix} = \begin{bmatrix} \cos \alpha_y & 0 & -\sin \alpha_y \\ 0 & 1 & 0 \\ -\sin \alpha_y & 0 & \cos \alpha_y \end{bmatrix} \cdot \begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix}\tag{5.16}$$

Finally, for X axis rotation,

$$\begin{bmatrix} P'''_x \\ P'''_y \\ P'''_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_x & -\sin \alpha_x \\ 0 & \sin \alpha_x & \cos \alpha_x \end{bmatrix} \cdot \begin{bmatrix} P''_x \\ P''_y \\ P''_z \end{bmatrix}\tag{5.17}$$

Thus, the new surface patch $\hat{S}$ is

$$\hat{P}_{i,j} = R_x R_y R_z \cdot P_{i,j},\tag{5.18}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_x & -\sin \alpha_x \\ 0 & \sin \alpha_x & \cos \alpha_x \end{bmatrix},$$

$$R_y = \begin{bmatrix} \cos \alpha_y & 0 & -\sin \alpha_y \\ 0 & 1 & 0 \\ -\sin \alpha_y & 0 & \cos \alpha_y \end{bmatrix},$$

$$R_z = \begin{bmatrix} \cos \alpha_z & -\sin \alpha_z & 0 \\ \sin \alpha_z & \cos \alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Thus, f_n is defined as

$$\forall P \in \mathbf{S}_i, f_n(U) = \alpha_n \sum_{P \in \mathbf{S}_i} \frac{P' - P}{\|P' - P\|} \quad (5.19)$$

At each deformation iteration, the surface patch model is updated by

$$P'_{ij} = P_{ij} + \Delta P_{ij} \quad (5.20)$$

ΔP_{ij} is the change of surface point.

$$\Delta P_{ij} = a_1 F_{int} + a_2 F_{ext} \quad (5.21)$$

where

$$F_{int} = f_c + f_e + f_a + f_n \quad (5.22)$$

$$F_{ext} = f_g + f_i \quad (5.23)$$

5.2.2 Inclusion of a Hierarchical Framework

Hierarchical mechanism can be introduced into deformable models to speed up the algorithm and reduce the local minima. Pyramidal image representations as proposed by Tanimoto [151] have been the first ones to define and exploit image reduction. Pyramids of frequency decomposition presented in Burt [152] provide a set of images at decreasing resolutions which are close to the visual perception of an observer at increasing distance. 3D image pyramids of any reduction factor were generated by Lachaud [13] because in their method the reduction factor of the resampling must be coherent with both the surface representation and its refinement.

In the construction of a classical Gaussian pyramid, the successive levels of that kind of pyramid are computed with the convolution of a Gaussian kernel of side 5 pixels. A low-cost filter without a phase translation linked to a reduction factor of 2 is guaranteed for each image dimension.

Let I_0 be the original image and the base of the pyramid and I_h the image of level h in the pyramid. The computation of I_{h+1} from I_h is given by the discrete convolution formula:

$$I_{h+1}(i', j', k') = \sum_{m,n,p=-2}^2 w(m, n, p) \cdot I_h(2i' + m, 2j' + n, 2k' + p), \quad (5.24)$$

where w is a Gaussian convolution kernel of size 5 voxels

$$\left[\frac{1}{16} \quad \frac{4}{16} \quad \frac{6}{16} \quad \frac{4}{16} \quad \frac{1}{16} \right]^3 \quad (5.25)$$

Figure 5.6: I_0 is the original image and the base of the pyramid. I_1 is the image of level 1 in the pyramid. I_2 is the image of level 2 in the pyramid.

In the proposed multi-scale GDSP model, a list of volumetric discrete images $I_0, I_1, \dots, I_m$ representing the 3D pyramid will be determined. I_0 is the original image at the real size. I_m is the image that includes only the lowest frequencies.

Specifically, three different levels are used in the experiments. This corresponds to the original image resolution I_0 , a subsampled image I_1 by a factor of 2, and a subsampled image I_2 by a factor of 4. Images of 3 different levels are shown in Figure 5.6.

For the two subsampled images I_1 and I_2 , we first calculate the external force field utilizing 3D GVF because of the wide search region of GVF.

The external force field of I_2 in $\mathbb{R}^3$ is obtained by minimizing the energy function

$$\varepsilon = \int \int \int \mu |\nabla \mathbf{v}|^2 + |\nabla f_2|^2 |\mathbf{v} - \nabla f_2|^2 dx dy dz, \quad (5.26)$$

where f_2 is the coarse edge of the subsampled image I_2 .

Similarly, the external force field of I_1 in $\mathbb{R}^3$ is obtained by minimizing the energy functional:

$$\varepsilon = \int \int \int \mu |\nabla \mathbf{v}|^2 + |\nabla f_1|^2 |\mathbf{v} - \nabla f_1|^2 dx dy dz \quad (5.27)$$

where f_1 is the coarse edge of the subsampled image I_1 .

The initial model in the original image, which is the surface extraction results in lower resolution, is already close enough to the real boundary of the object. The wide search region of 3D GVF is not necessary considering its high computational cost. Therefore, a traditional method is utilized to calculate the external force field for the original image I_0 .

Thus, the external force field of I_0 in $\mathbb{R}^3$ is obtained by minimizing the energy functional

$$\varepsilon = \int \int \int P(\mathbf{v}) dx dy dz \quad (5.28)$$

$$P(\mathbf{v}) = -|\nabla(G_\sigma * I_0)| \quad (5.29)$$

where ∇ is the gradient operator and $G_\sigma * I_0$ denotes the image convolved with a Gaussian smoothing filter whose characteristic width σ controls the spatial extent of the local minima of P . By defining such a potential function P , the surface model will be attracted to intensity edges in the original image I_0 .

In summary, a multistage segmentation mechanism is introduced in the 3D GDSP model. First, two subsampled 3D images I_1 and I_2 are obtained using 5.24. For the two subsampled images I_1 and I_2 , the external force field is calculated utilizing 3D GVF because of its wide search region. In order to save computational costs,

a traditional method is utilized to calculate the external force field for the original image I_0 . At the beginning stage, in order to reduce the unnecessary computational cost, bigger surface patches are set for coarse boundary extraction. Each patch is examined and the patches are divided into smaller size to obtain the detail of the target. The experimental results show that the computational costs are reduced with the help of the hierarchical framework.

5.2.3 Computation of the Surface Area

An efficient algorithm for surface area calculation of objects of interest is developed as well in this Chapter. The surface area calculation method is based on the segmentation result of the proposed robust GDSP model.

In the proposed robust GDSP model $\mathbf{M}$, the smallest element is a surface patch S_i which consists of 16 knots. The final result $\mathbf{M}$ consists of a number of surface patches S which are connected to each other in sequence without overlapping or gap.

For each surface patch S_i , Lagrange interpolation is utilized to generate the surface model from the 16 knots in matrix $\mathbf{L}_i$. Based on [153], every surface point $P(u, w)$ is located as shown below.

$$P(u, w) = \mathbf{u} \mathbf{M}_L \mathbf{L}_i \mathbf{M}_L^T \mathbf{w}^T \quad (5.30)$$

$$P(u, w) = (x(u, w), y(u, w), z(u, w)) \quad (5.31)$$

$$x(u, w) = \mathbf{u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \mathbf{w}^T \quad (5.32)$$

$$y(u, w) = \mathbf{u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \mathbf{w}^T \quad (5.33)$$

$$z(u, w) = \mathbf{u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \mathbf{w}^T \quad (5.34)$$

where

$$\mathbf{L}_i = \begin{bmatrix} U_{i-1,j-1} & U_{i-1,j} & U_{i-1,j+1} & U_{i-1,j+2} \\ U_{i,j-1} & U_{i,j} & U_{i,j+1} & U_{i,j+2} \\ U_{i+1,j-1} & U_{i+1,j} & U_{i+1,j+1} & U_{i+1,j+2} \\ U_{i+2,j-1} & U_{i+2,j} & U_{i+2,j+1} & U_{i+2,j+2} \end{bmatrix},$$

$$\mathbf{u} = [u^3 \quad u^2 \quad u \quad 1],$$

$$\mathbf{w} = [w^3 \quad w^2 \quad w \quad 1],$$

and

$$\mathbf{M}_L = \begin{bmatrix} -\frac{9}{2} & \frac{27}{2} & -\frac{27}{2} & \frac{9}{6} \\ \frac{9}{2} & -\frac{45}{2} & 18 & -\frac{9}{2} \\ -\frac{11}{2} & 9 & -\frac{9}{2} & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

$\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are the three x , y , z components of the surface point matrix $\mathbf{L}_i$.

The area of each surface patch S_i , A_i , can be calculated conveniently using the following equation:

$$A_i = \int_0^1 \int_0^1 \sqrt{P_u^2 P_w^2 - (P_u P_w)^2} du dw, \quad (5.35)$$

where

$$\begin{aligned} P_u^2 &= \left(\frac{\partial P(u, w)}{\partial u} \right)^2 \\ &= \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \mathbf{w}^T \right)^2 + \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \mathbf{w}^T \right)^2 + \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \mathbf{w}^T \right)^2 \\ &= ([3u^2 \quad 2u \quad 1 \quad 0] \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \mathbf{w}^T)^2 \\ &\quad + ([3u^2 \quad 2u \quad 1 \quad 0] \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \mathbf{w}^T)^2 \\ &\quad + ([3u^2 \quad 2u \quad 1 \quad 0] \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \mathbf{w}^T)^2 \end{aligned} \quad (5.36)$$

$$\begin{aligned} P_w^2 &= \left(\frac{\partial P(u, w)}{\partial w} \right)^2 \\ &= \left(\mathbf{u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right)^2 + \left(\mathbf{u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right)^2 + \left(\mathbf{u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right)^2 \\ &= (\mathbf{u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T [3w^2 \quad 2w \quad 1 \quad 0]^T)^2 \\ &\quad + (\mathbf{u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T [3w^2 \quad 2w \quad 1 \quad 0]^T)^2 \\ &\quad + (\mathbf{u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T [3w^2 \quad 2w \quad 1 \quad 0]^T)^2 \end{aligned} \quad (5.37)$$

$$\begin{aligned}
P_u P_w &= \frac{\partial x}{\partial u} \frac{\partial x}{\partial w} + \frac{\partial y}{\partial u} \frac{\partial y}{\partial w} + \frac{\partial z}{\partial u} \frac{\partial z}{\partial w} \\
&= \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \mathbf{w}^T \right) \left(\mathbf{u} \mathbf{M}_L \mathbf{X}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right) \\
&\quad + \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \mathbf{w}^T \right) \left(\mathbf{u} \mathbf{M}_L \mathbf{Y}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right) \\
&\quad + \left(\frac{\partial \mathbf{u}}{\partial u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \mathbf{w}^T \right) \left(\mathbf{u} \mathbf{M}_L \mathbf{Z}_i \mathbf{M}_L^T \frac{\partial \mathbf{w}^T}{\partial w} \right)
\end{aligned} \tag{5.38}$$

Because the final result, $\mathbf{M}$, consists of a number of surface patches, S . All the surface patches are connected to each other in sequence without overlapping or gap. Therefore, for the entire surface model, area of the surface model $A_{\mathbf{M}}$ is obtained by summarizing all the surface areas of each surface patch

$$A_{\mathbf{M}} = \sum_{i=1}^n A_i \tag{5.39}$$

where n is the total number of surface patches in $\mathbf{M}$.

5.3 Experimental Results and Discussion

5.3.1 Effect of the New Internal Force

In the real application, if the position of the first surface patch is not initialized properly, the method presented in Chapter 4 sometimes will fail to obtain correct segmentation result. The worst situation is to initialize a surface patch model perpendicular to the boundary of the target. Without the help of the proposed new internal force f_n , in the case of segmenting object with sharp corner in 3D space, the deforming surface patch may not perform very well. The shape of surface patches may be irregular and the accuracy will be low as well. In the following experi-

ments, several results with and without the newly introduced internal force f_n are presented.

First, in the experiment on the segmentation of torus, we set $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_n = 0.0$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. Thus, the new internal force f_n was not included. The mesh was part of the target to segment. The surface patch in Figure 5.7 (a) is the initial surface model which is perpendicular to the true boundary. The experimental results from Figure 5.7 (b) to Figure 5.7 (d) show that the knots are forced to move along the direction of the external force. The knots of the surface patch will come close to each other under the influence of the external force. At the same time, the traditional internal forces worked opposite to the external force. However, they cannot let the knots converge to the actual boundary of the object quickly. At last, the surface patch vibrated perpendicularly to the actual boundary of the object, and can not allocate the correct boundary by rotating itself. From Figure 5.7 (d), we can see the accuracy is not acceptable. Because the subsequent surface patches will be generated based on the first one, the segmentation accuracy of the first surface patch will influence the initialization of the subsequent surface patches directly. In this case, the first surface patch failed to locate the boundary of the target due to bad initialization. This surface patch was still perpendicular to the true boundary after deformation. Thus the subsequent surface patches will be also perpendicular to the true boundary after they were generated based on the first one. It caused the subsequent surface patches fail to extract the correct boundary as well. That is how the process will fail if the first surface patch is badly initialized.

Because the traditional internal forces work against the external force, they prevent the bad initialized surface patch from converging to the boundary of target. One method to solve that problem is to reduce the influence of traditional internal forces by decreasing α_c and α_e . In the second experiment, without f_n , we set $\alpha_c = 0.05$, $\alpha_e = 0.05$, $\alpha_a = 0.5$, $\alpha_n = 0.0$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. The experimental results from Figure 5.8 (b) to Figure 5.8 (h) show that the knots are forced to move along the direction of the external force and reach the boundary of the target. However,

Figure 5.7: Surface evolution without the new introduced internal force f_n in the segmentation of torus: (a) initial surface patch model; (b) surface patch at iteration 5; (c) surface patch at iteration 10; (d) surface patch at the last iteration.

all the knots of the surface patch came close to each other since the influence of the spreading internal force was reduced. At last, the surface patch turned into a narrow line on the boundary of the object. As mentioned before, the subsequent surface patches will be generated based on the result of the first one. When the first surface patch is deformed and turns into a narrow line on the boundary of the object, the initialization process of the subsequent surface patches will have problems. It is because it is difficult to predict the growth direction of the four new surface patches based on only a line. In this case, two new surface patches will be generated as two narrow lines as well. These two lines still can locate the boundary of the target. However, the other new surface patches will fail to obtain direction information from the first surface patch. In the experiments, these two subsequent surface patches are often deformed and overlap the first surface patch in the end. Thus the growth mechanism fails, no closed surface model will be obtained in the end.

Finally, we set $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_n = 0.175$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. Here, the new internal force f_n was included. We remained the magnitude of the traditional internal forces by setting α_c and α_e as normal value.

The experimental results from Figure 5.9 (b) to Figure 5.9 (d) show that the surface patch rotates under the influence of f_n when the initial surface patch was in the worst position. From Figure 5.9 (d), we can see that the badly initialized surface patch located the boundary of the target successfully, and the segmentation error was within 1 voxel. The experimental result shows that the new internal force helps when initialization is bad. It is because the difference between the normal and the direction of the external force vector is utilized by including this new internal force in every deformation step. When the direction of the normal and external force vector is different, the new internal force will start to minimize this difference by rotating the surface patch. Thus, every knot was forced to move towards the actual boundary of the object quickly, at the same time, the surface patch can adjust its position by rotating itself. At last, the surface patch found the actual boundary of the object.

Figure 5.8: Surface evolution without the new introduced internal force f_n in ring segmentation, but α_c and α_e are reduced to remove the strong opposite force against the external force: (a) initial surface patch model; (b) surface patch at iteration 4; (c) surface patch at iteration 7; (d) surface patch at the last iteration.

Figure 5.9: Surface evolution with the new introduced internal force f_n in ring segmentation: (a) initial surface patch model; (b) surface patch at iteration 3; (c) surface patch at iteration 6; (d) surface patch at iteration 8.

The proposed novel internal force was applied in human vertebra segmentation. In this experiment, our goal is to extract and represent the vertebra boundary from the input image. Moreover, a comparison between method with and without the novel internal force was carried out to demonstrate the improvement due to the proposed new introduced internal force.

We used the data set, which was the same as in Section 4.4.3, as the input image. In the first group of experiments which utilized internal force f_n , the model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$, $\alpha_2 = 0.9$ and $\alpha_n = 0.175$. In the second group of experiments without f_n , the model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$, $\alpha_2 = 0.9$ and $\alpha_n = 0$. The influence of f_n was ignored in the second group of experiments by setting α_n , the coefficient of f_n , as 0. All other parameters remained the same with the first group of experiments.

In every group of the two experiments, the first initial surface patch of the proposed deformable model was set respectively from the best position to ordinary position and finally to the worst position. In the best initialization situation **I**, the surface patch was initialized in parallel to the target. In the worst initialization situation **II**, the surface patch was initialized perpendicularly to the target. In the ordinary situation **II**, we set the angle between the normal to the initial surface patch and the normal to the target boundary as 45 degree.

The experimental results of these three groups listed in Table 5.1 show that when the first surface patch was badly initialized, the deformable surface model cannot work without f_n . That is because the first surface patch must find the correct boundary briefly. Subsequent new generated surface patches are all based on this result directly or indirectly. If the first surface extraction result is totally wrong, subsequent surface patches cannot find correct boundary and the method fails. The new introduce internal force f_n could guide the first surface patch find the correct boundary even in the worst initialization situation. Then, subsequent surface patches can find the correct boundary of the target based the first surface patch. Therefore, the proposed

topologically adaptable deformable model is not sensitive to the initialization of the first surface patch with the help of the new introduced internal force f_n . When the first surface patch is initialized properly, both methods can segment the target successfully but with difference accuracy.

Table 5.1: Experimental results of internal force in the segmentation of human vertebra.

Initialization	Group I		Group II		Group III	
Coefficient of f_n	$\alpha_n = 0.175$	$\alpha_n = 0$	$\alpha_n = 0.175$	$\alpha_n = 0$	$\alpha_n = 0.175$	$\alpha_n = 0$
Avg Dist Error	0.31 voxel	0.44 voxel	0.31 voxel	0.44 voxel	0.31 voxel	-
Max Dist Error	2 voxels	3 voxels	2 voxels	3 voxels	2 voxels	-

Figure 5.10 shows the experimental result in human vertebra segmentation Group I. The surface models in the first column (a1) and (a2) are parts of the surface extraction result with f_n , the surface models in the second column (b1) and (b2) are parts of the surface extraction result without f_n . The new introduced internal force f_n can improve the segmentation accuracy at the portion with sharp corner. It is because the internal force f_n can guide surface patches move perpendicularly to the direction of the external force at that position, while tradition external force can only change the position of surface points locally. The external force field of sharp corner is shown in Figure 5.11. Therefore, f_n can also improve the accuracy of the proposed method when the first surface patch is well initialized if the target have constructs with sharp corner.

In the following experiment, the brain ventricle was segmented from a T2 weighted MR image of human brain. The volume image size was $256 \times 256 \times 138$. A comparison between method with and without the novel internal force f_n was carried out to demonstrate the improvement due to the proposed new introduced internal force. In the first group of experiments which utilize internal force f_n , the model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$, $\alpha_2 = 0.9$ and

Figure 5.10: (a1) and (a2) are parts of the surface extraction result with f_n ; (b1) and (b2) are parts of the surface extraction result without f_n .

Figure 5.11: External force field of sharp corner.

$\alpha_n = 0.175$. In the second group of experiments without f_n , the model parameters were: $d_w = 6$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_1 = 0.1$, $\alpha_2 = 0.9$ and $\alpha_n = 0$. The influence of f_n was ignored in the second group of experiments by setting α_n , the coefficient of f_n , as 0. All other parameters remained the same with the first group of experiments.

In every group of the two experiments, the first initial surface patch of the proposed deformable model was set respectively from the best position to ordinary position and finally to the worst position. In the best initialization situation **I**, the surface patch was initialized in parallel to the target. In the worst initialization situation **III**, the surface patch was initialized perpendicularly to the target. In the ordinary situation **II**, we set the angle between the normal to the initial surface patch and the normal to the target boundary as 45 degrees.

The surface evolution process of the proposed method shows that the unclosed surface model will grow along the boundary of the target. New surface patches were generated from the active surface patches. This process was repeated until a closed

surface of the object is obtained.

Table 5.2: Experimental results of internal force in the segmentation of brain ventricle.

Initialization	Group I		Group II		Group III	
Coefficient of f_n	$\alpha_n = 0.175$	$\alpha_n = 0$	$\alpha_n = 0.175$	$\alpha_n = 0$	$\alpha_n = 0.175$	$\alpha_n = 0$
Avg Dist Error	0.30 voxel	0.41 voxel	0.30 voxel	0.41 voxel	0.30 voxel	-
Max Dist Error	2 voxels	3 voxels	2 voxels	3 voxels	2 voxels	-

The experimental results of these three groups which is listed in Table 5.2 show that when the first surface patch is badly initialized, the deformable surface model cannot work without f_n . That is because the first surface patch must find the correct boundary. Subsequent new generated surface patches are all based on this result directly or indirectly. If the first detection result is totally wrong, subsequent surface patches cannot find correct boundary and the method fails. The new introduce internal force f_n can guide the badly initialized surface patch to find the correct boundary even in the worst initialization situation. Subsequent surface patches can find the correct boundary of the target based on the first surface patch. Therefore, the proposed topologically adaptable deformable model is not sensitive to the initialization of the first surface patch with the help of the new introduced internal force f_n . When the first surface patch is initialized properly, both methods can segment the target successfully but with different accuracy.

The experimental result Group I is shown in Figure 5.12. The surface model (a) is the detection result with f_n , the surface models (b) is the detection result without f_n . The proposed internal force f_n can increase the detection accuracy at the portion of the object with sharp corner. It is because the internal force f_n can guide surface patches to move perpendicularly to the direction of the external force at those positions. While the tradition external force can only change the position of surface points locally.

Figure 5.12: (a) is the ventricle detection result with f_n ; (b) is the ventricle detection result without f_n .

5.3.2 Effect of the Hierarchical Framework

Specifically, three different levels were used in the experiments. This corresponded to the original image resolution I_0 , a subsampled image I_1 by a factor of 2, and a subsampled image I_2 by a factor of 4. Thus, d_w , the desired mean edge length of surface patches, had three levels as well: d_{w0} , d_{w1} and d_{w2} . In these group of experiments, the new introduced internal force f_n was included by setting its parameter $\alpha_n = 0.175$.

Case 1: Segmentation of Brain Ventricle

In order to test this method, it was applied in the segmentation of brain ventricle from MR images. We set d_{w0} as 6, d_{w1} as 12 and d_{w2} as 24. External force field $E_{f_{ext2}}$ and $E_{f_{ext1}}$ were computed using modified 3D GVF. $E_{f_{ext0}}$ was computed using the traditional gradient force in order to reduce the computational cost.

First, the model parameters were set as: $d_w = 24$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_n = 0.175$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. A coarse boundary of the brain ventricle was obtained based on the low level surface model and external force field $E_{f_{ext2}}$. This is shown in Figure 5.13.

Next, the model parameters remained the same except d_w was reduced to 12 voxels. Using the coarse boundary of the target obtained from the low resolution image I_2 as the initial surface model, a boundary of the brain ventricle M_1 was obtained based on the median level surface model and external force field $E_{f_{ext1}}$. It is shown in Figure 5.14. In this procedure, every surface patch in the surface model M_2 which was the coarse boundary of the target obtained from the low resolution image I_2 was divided into 4 surface patches.

Figure 5.13: A coarse boundary of the brain ventricle is obtained based on the low level surface model and external force field $E_{f_{ext}2}$.

In the last step, the model parameters remain the same except d_w was reduced to 6 voxels. M_1 , the boundary of the brain ventricle obtained from the median resolution image I_1 , was utilized as the initial surface model for surface extraction in original image. Every surface patch in the surface model M_1 was divided into 4 surface patches. Finally, an accurate boundary of the brain ventricle was obtained based on the high level surface model and external force field $E_{f_{ext}0}$, shown in Figure 5.15.

Figure 5.16 compares the behavior of both methods with and without multi-scale frame work in this experiment. The behavior of the first approach without the help of the multi-scale frame work is clear. The kinetic energy curve shows the slow convergence of the model and the small variations of its number of vertices. The behavior of the second approach with the help of the multi-scale frame work is also displayed in Figure 5.16. The kinetic energy curve shows that in the second

Figure 5.14: A coarse boundary of the brain ventricle is obtained based on the median level surface model and external force field $E_{f_{ext}1}$.

Figure 5.15: A refined boundary of the brain ventricle is obtained based on the low level surface model and external force field $E_{f_{ext}0}$.

Figure 5.16: A comparison of the two methods of the segmentation of brain ventricle (with and without multi-scale frame work).

method, the surface model can converge to the true boundary of the object much more quickly.

Case 2: Segmentation of Human Vertebra

The proposed method was applied in the segmentation of human vertebra from MR images. We set $d_{w0} = 6$, $d_{w1} = 12$ and $d_{w2} = 24$. External force field $E_{f_{ext}2}$ and $E_{f_{ext}1}$ were computed using modified 3D GVF. $E_{f_{ext}0}$ was computed using the traditional gradient force in order to reduce the computational cost.

First, the model parameters were set as: $d_w = 24$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_n = 0.175$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. A coarse boundary of the human vertebra was obtained based on the low level surface model and external force field $E_{f_{ext}2}$, shown in Figure 5.17.

Figure 5.17: A coarse boundary of the human vertebra is obtained based on the low level surface model and external force field $E_{f_{ext}2}$.

Next, keeping the other parameters the same, d_w was reduced to 12 voxels. Using the coarse boundary of the target obtained from the low resolution image I_2 as the initial surface model, a boundary of the human vertebra M_1 was obtained based on the median level surface model and external force field $E_{f_{ext}1}$. This shown in Figure 5.18. In this procedure, every surface patch in the surface model M_2 which was the coarse boundary of the human vertebra obtained from the low resolution image I_2 was divided into 4 surface patches.

In the last step, d_w was reduced to 6 voxels. M_1 , the boundary of the target obtained from the median resolution image I_1 , was utilized as the initial surface model for surface extraction in original image. Every surface patch in the surface model M_1 was divided into 4 surface patches. Finally, an accurate boundary of the human vertebra was obtained based on the high level surface model and external force field $E_{f_{ext}0}$. This is shown in Figure 5.19.

Figure 5.20 compares the behavior of both methods with and without multi-scale

Figure 5.18: A coarse boundary of the human vertebra is obtained based on the median level surface model and external force field $E_{f_{ext}1}$.

Figure 5.19: A refined boundary of the human vertebra is obtained based on the low level surface model and external force field $E_{f_{ext}0}$.

Figure 5.20: Two methods of human vertebra segmentation are compared on the graph (with and without multi-scale frame work).

frame work in the experiment of human vertebra segmentation. The behavior of the first method without the help of the multi-scale frame work is clear. The kinetic energy curve shows the slow convergence of the model and the small variations of its number of vertices. The behavior of the second approach with the help of the multi-scale frame work is also displayed. The kinetic energy curve shows that in the second method, the surface model can converge to the true boundary of the object much more quickly.

5.3.3 Computation of the Surface Area

In order to compute the surface area of the object of interest, a segmentation result $\mathbf{M}$ was obtained first using the proposed robust GDSP model. $\mathbf{M}$, the closed surface of the object of interest, consists n non-overlapping surface patches S . n was the total number of the surface patches in proposed robust GDSP model $\mathbf{M}$.

After a closed surface $\mathbf{M}$ was obtained, the surface area of each surface patch S was calculated using Equation 5.35. By summing A_i , $i = 1, \dots, n$, the surface area of the entire surface model $\mathbf{M}$ was obtained using Equation 5.39.

To validate the proposed algorithm of area calculation, this method was applied to several simulated images. It is because we know the ground truth of the target by calculating the theoretical area value of each target. Here, a sphere, a cubic, a cylinder and a torus were chosen as the targets. The theoretical area value of each target was derived respectively as the ground truth.

For the sphere, we set the radius R as 1cm. Thus, the theoretical area of the target was $S = 4\pi r^2 = 12.56\text{cm}^2$. For the cubic, we set the length a as 3cm. Thus, the theoretical area of the cubic was $S = 4a^2 = 36\text{cm}^2$. We set the radius r of the cylinder as 1cm, the height h as 3cm. Thus, the theoretical area of the cylinder was $S = 2\pi r^2 + 2\pi rh = 25.12\text{cm}^2$. For the torus, we set R as 1.5, r as 0.5, denoted in Figure 4.14. The theoretical area of the torus was $S = 4\pi^2 Rr = 29.58\text{cm}^2$.

We used a $200 \times 200 \times 200$ 3D image space to represent the $5\text{cm} \times 5\text{cm} \times 5\text{cm}$ real space. Thus, for each voxel, it was a $0.025\text{cm} \times 0.025\text{cm} \times 0.025\text{cm}$ cubic. After the boundary of the simulated object was extracted from the 3D image space, the actual position of each surface point in real space can be obtained because the size of each voxel is known.

Each target was segmented respectively using the proposed method. We set $d_w = 9$ voxels, $\alpha_c = 0.175$, $\alpha_e = 0.175$, $\alpha_a = 0.5$, $\alpha_n = 0.175$, $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$. Using the proposed robust GDSP model, surface model of each target was obtained respectively, shown in Figure 5.21.

Based on the algorithm presented in Section 5.2.3, the surface area of each target was calculated. The calculation error of the surface area can be obtained by comparing the calculated value with the ground truth, i.e., the theoretical value. Table 5.3 lists the area values and percent errors.

Figure 5.21: The segmentation results of the simulated objects.

Table 5.3: The comparison of the theoretical area value and the area value obtained by the proposed algorithm.

	Ground truth	Obtained Value	Error
Sphere	28.27 cm^2	28.44 cm^2	0.60%
Cubic	36.00 cm^2	36.14 cm^2	0.39%
Cylinder	25.12 cm^2	25.48 cm^2	1.49%
Torus	29.58 cm^2	30.09 cm^2	1.72%

As shown in Table 5.3, we can see that the proposed method can achieve high accuracy in calculating the surface area after the segmentation of the target. This method can be utilized in feature extraction of the target. For example, it can provide geometric information on the target to evaluate the efficiency of particular treatment in medical image analysis.

5.4 Concluding Remarks

The deformation procedure may not work properly when the initialized surface patch is parallel to the external force vector in 3D space. To solve this problem, an additional internal force, f_n , is introduced in the robust GDSP model. This novel internal force could rotate the deforming surface patch and let it move perpendicularly to the external force vector. The experimental results show that the proposed method can obtain the correct boundary of the target even if the initialized surface patch is in the worst situation. Thus, the proposed method is not sensitive to the surface patch initialization. Also, it can improve segmentation accuracy in segmenting object with sharp corner. The experiments show that the proposed robust GDSP model achieve better results then the original method. In the proposed robust GDSP model, a hierarchical mechanism is introduced to speed up algorithm as well. An efficient algorithm for surface area calculation of objects of interest is developed as well in this Chapter. The experimental results show that it can calculate the surface area of the target with high accuracy.

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

In this thesis, novel approaches for 3D object segmentation are presented. A number of new ideas have been introduced to improve the performance of current techniques. A continuous deformable model, named B-Surface algorithm, is proposed for the segmentation of smooth 3D objects. For the segmentation of complicated 3D objects, a GDSP model is proposed. A robust GDSP model is proposed as well to remove the sensitivity to the surface model initialization and reduce the computational cost. These methods can be applied to the analysis of medical images such as segmentation of brain ventricle, brain tumor and human vertebra.

Firstly, the methods in the newly developing area of object segmentation have been reviewed in this thesis, which offers many applications to medical image analysis. Much progress has been made recently in this area, but many important problems remain open. For example, because of the success of 2D statistical deformable models, many researchers work on 3D statistical deformable models. However, the computational cost increases in 3D case. Even the building procedure of PDM becomes a difficulty for 3D statistical deformable models. Compared with 2D cases, more training data sets which are obtained mainly by manual labelled from experts

are necessary. It is because the high dimension (in terms of thousands) of the feature vector is required in the construction of 3D PDM. Even if a PDM is constructed successfully for each kind of object that we want to segment, there are also some other difficulties. For example, in medical image analysis, the target often has a significant shape variation because of some disease. In this situation, it is very hard to find the correct boundary for 3D statistical deformable models because of the strong constraints which come from the limited training data.

One contribution of this thesis is to develop a novel B-Surface algorithm for smooth 3D object segmentation. It overcomes the difficulty that comes from analyzing 3D volume image slice by slice. Next, it is easy to obtain the curvature and normal of the object surface because B-Surface is a continuous deformable model. It has the ability to achieve a high compression ratio (ratio of data to parameters) by presenting the whole surface with only a relatively small number of control points. Compared with traditional 3D deformable models, the computational cost of B-Surface deformation is reduced since the internal forces are not needed to compute at every deformation iteration. However, the proposed B-Surface model is suitable for simple object segmentation. Therefore, it will meet difficulty in segmenting object with more subtle structures because of the intricate boundaries. One of the reasons is a bi-cubic B-spline surface has strong implicit constraints. It will be difficult to segment and represent the object with subtle structures such as amygdale/hippocampus or superior temporal gyrus.

GDSP model is proposed for the segmentation of complicated 3D objects. In the proposed method, a growing mechanism is introduced to achieve topologically adaptable object segmentation. As a general deformable surface model, it can be applied to complicated object segmentation without building a specific template for each target in advance. The experimental results on the segmentation of brain ventricle segmentation and human vertebra in 3D medical images show that this method overcomes the limitations of traditional deformable models. It achieves topologically adaptable surface extraction by connecting new surface patches with active patches

and automated triangulating the square patch in particular situations. Compared with the existing topologically adaptable deformable surfaces, no splitting or merging judgement is carried out among all the vertices at each deformation iteration. Topologically adaptable surface extraction is achieved by analyzing the possibility of patch connection among the “active” surface patches only. Thus, the computational cost of the proposed method is reduced significantly. The surface model initialization step is simplified. Only a surface patch, rather than a whole surface model is initialized. The proposed method is explicit. Therefore, it is convenient to provide geometrical information such as area, volume, or local curvature of the object. Furthermore, surface curvature adaptiveness is achieved in the proposed deformable model by associating the surface curvature with the size of the surface patches. The size of surface patches is reduced automatically to obtain the details when they reach the portion of the object with small radius of curvature.

A robust algorithm is developed for 3D complicated object segmentation. The deformation procedure may not work properly when the initialized surface patch is parallel to the direction of the external force vector in 3D space. To solve this problem, a novel internal force is introduced in the proposed method. This novel internal force could rotate the deforming surface patch and let it move perpendicularly to the direction of the external force vector. The experimental results show that the proposed method can obtain the correct boundary of the target even if the initialized surface patch is in the worst situation. Thus, the proposed method is not sensitive to the surface patch initialization. It can also improve segmentation accuracy in segmenting object with sharp structure. In the proposed robust GDSP model, a hierarchical mechanism is introduced to speed up algorithm. An efficient algorithm for surface area calculation of objects of interest was developed as well in this Chapter. The experiments show that the proposed robust GDSP model yields better results than the original method. It can also calculate the surface area of the target with high accuracy.

In conclusion, the new algorithms proposed in this thesis will bring one step closer

to the real-time segmentation of 3D object. This will be very useful in the analysis of medical image.

6.2 Recommendations for Future Work

While good progress has been demonstrated in object segmentation from 3D images in the past few years, much research work can be continued.

6.2.1 B-Surface Algorithm Based on Closed Surface Model

The current B-Surface algorithm works well for the segmentation of smooth 3D object. The proposed algorithm can be extended in the following ways. The surface model for the proposed method forms a closed boundary of objects by overlapping many bi-cubic B-spline surface patches. Differential quantities of the target could be obtained conveniently on every surface points on each bi-cubic B-spline surface patch. However, in the overlapping portion of two adjacent bi-cubic B-spline surface patches, the coincidence of differential quantities are not guaranteed. One method to overcome this shortcoming is to build a closed surface model using bi-cubic B-spline surface for particular targets such as left ventricle of human heart, brain ventricle, artery, etc. First, utilizing computer graphics technique, a closed surface model of particular target is built with a bi-cubic B-spline surface model or a triangle B-spline surface model. Then, under the influence of external force and other constraints, this initial surface model can be deformed by updating the position of the control points to locate the boundary of one kind of objects.

6.2.2 Growing B-Surface Algorithm without Overlapping

Another approach is to generate a growing B-Surface model without overlapping. In the smooth portion of the surface model, growing mechanism could be used to ex-

tend the unclosed surface model. When two bi-cubic B-spline surface patches come close to each other to generate a closed surface model, the technique of multiple control points could be used to connect the two patches. However, when using multiple control points to connect two bi-cubic B-spline surface patches, the two connected patches can only achieve C^0 class continuity. Thus, the coincidence of differential quantities such as slope and curvature are not guaranteed when using multiple control points. A closed bi-cubic B-spline surface model without overlapping or leakage is a challenge for B-Surface algorithm. To solve this problem, the possible method is to find the correspondence of control points and merge the corresponding control points of two bi-cubic B-spline surface patches and generate a smooth connection.

6.2.3 Statistical GDSP Model

GDSP model has a flexibility to achieve topologically adaptable object segmentation and surface extraction. Although it does not utilize a prior knowledge from the training set, the geometric and statistical information from the nearby surface patches can be utilized in the deformation procedure of a new surface patch to improve its robustness in the noisy condition. First, an attribute vector can be utilized to distinguish object boundary. The attribute vector can be calculated from moments, Wavelets, or others. A classification method could be used in the proposed GDSP model for determining the similarity of the model's attribute vector and subject's attribute vector.

6.2.4 On-line Learning GDSP Model

Normally, statistical deformable models learn the "experience" off-line. In order to generate the prior knowledge for statistical deformable models, a set of training data must be analyzed in advance. The proposed method could achieve on-line training process. For every input data, an attribute vector field could be calculated first. The first surface patch is initialized near the peak of the field which means the

most distinguished boundary. Then the open surface model grows up based on the initial surface patch along the boundary of the object. In the growing procedure of model, statistical information could be collected efficiently along the growing surface patches. When the surface model reaches blurred portion, the statistical information of nearby surface patches will be utilized to find the boundary of the object.

Author's Publications

- [1] Xujian Chen and Eam Khwang Teoh, "Growing Deformable Surface Patches for Segmentation of 3D Medical Images," *submitted to Pattern Recognition*, Jul 2007.
- [2] Xujian Chen and Eam Khwang Teoh, "Active Growing Deformable Surface Patches," *submitted to SPIE Symposium on Medical Imaging*, Jul 2007.
- [3] Xujian Chen and Eam Khwang Teoh, "3D Object Segmentation Using B-Surface," *Image and Vision Computing*, 23(14): 1237-1249 2005.
- [4] Xujian Chen and Eam Khwang Teoh, "An Improved Growing Deformable Surface Patches Model for Analysis of Medical Images," in *Proceedings of IAPR Conference on Machine Vision Applications'05*, Tsukuba Science City, Japan, May 2005.
- [5] Xujian Chen and Eam Khwang Teoh, "Growing Deformable Surface Patches for Topology-Adaptive Object Detection in MR Images," in *Proceedings of SPIE Symposium on Medical Imaging'05*, San Diego, USA, Feb 2005.
- [6] Xujian Chen and Eam Khwang Teoh, "3D Growing Deformable B-Surface Model for Object Detection," in *Proceedings of International Conference on*

Automation, Robotics, Control and Vision (ICARCV '04), Kunming, China,
Dec 2004, pp. 357-362.

Bibliography

- [1] M. Kass and A. Witkin, "Snake: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.
- [2] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: recovering 3D shape and nonrigid motion," *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, 1988.
- [3] C. Davatzikos and R. N. Bryan, "Using a deformable surface model to obtain a shape representation of the cortex," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 785–795, 1996.
- [4] M. Jackowski, M. Satter, and A. Goshtasby, "Approximating digital 3D shapes by rational gaussian surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 1, pp. 56–69, 2003.
- [5] S. Pizer, T. Fletcher, S. Joshi, and A. Thall, "Deformable m-reps for 3D medical image segmentation," *International Journal of Computer Vision*, vol. 55, pp. 85–106, 2003.
- [6] A. Klein, F. Lee, and A. Amini, "Quantitative coronary angiography with deformable spline models," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 468–482, 1997.
- [7] T. Cham and R. Cipolla, "Automated b-spline curve representation incorporating mdl and error-minimizing control point insertion strategies," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 49–53, 1999.

- [8] Y. Wang, E. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image and Vision Computing*, vol. 22, pp. 267–280, 2004.
- [9] J. Huang and A. A. Amini, "Anatomical object volumes from deformable B-spline surface models," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 732–736, 1998.
- [10] T. F. Cootes and C. J. Taylor, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38–59, 1995.
- [11] N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images. an improved active shape model," *IEEE Transactions on Medical Imaging*, vol. 17, no. 6, pp. 1049–1062, 1998.
- [12] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [13] J.-O. Lachaud and A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction," *Medical Image Analysis*, vol. 3, no. 2, pp. 187–207, 1998.
- [14] R. Szeliski, D. Tonnesen, and D. Terzopoulos, "Modeling surfaces of arbitrary topology with dynamic partiles," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY*, pp. 82–87, June, 1993.
- [15] D. DeCarlo and J. Gallier, "Topological evolution of surfaces," *Graphics Interface*, pp. 194–203, 1996.
- [16] T. McInerney and D. Terzopoulos, "Topology adaptive deformable surfaces for medical image volume segmentation," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 840–850, 1999.

- [17] Y. Duan and H. Qin, "Intelligent balloon: a subdivision-based deformable model for surface reconstruction of arbitrary topology," in *Proceedings of Sixth ACM symposium on Solid modeling and applications*, pp. 47–58, ACM Press, 2001.
- [18] J. C. Rajapakse, R. G. N. Meegama, and L. S. Ho, "NURBS-based analysis of gender-related variability of the human brain surface," in *Proceedings of International Conference on Control, Automation, Robotics and Vision*, pp. 520–524, 2002.
- [19] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," *Pattern Recognition*, vol. 11, pp. 191–204, 1979.
- [20] K. Mardia and T. Hainsworth, "A spatial thresholding method for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 919–927, 1988.
- [21] P. K. Sahoo, S. Soltani, A. K. Wong, and Y. C. Chen, "A survey of thresholding techniques," *Computer Vision Graphics Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.
- [22] T. Taxt, P. Flynn, and A. Jain, "Segmentation of document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1322–1329, 1989.
- [23] J. R. Parker, "Gray level thresholding in badly illuminated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 813–819, 1991.
- [24] N. Papamarkos and B. Gatos, "A new approach for multilevel threshold selection," *CVGIP: Graphical Models Image Process*, vol. 56, no. 5, pp. 357–370, 1994.

- [25] J. Yen, F. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370–378, 1995.
- [26] O. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 312–315, 1995.
- [27] M. Cheriet, J. N. Said, and C. Y. Suen, "A recursive thresholding technique for image segmentation," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 918–921, 1998.
- [28] L. Yang and J. Yen, "An adaptive simplex genetic algorithm," in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2000)* (D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, eds.), (Las Vegas, Nevada, USA), p. 379, Morgan Kaufmann, 10-12 2000.
- [29] L. Cohen, L. Vinet, P. Sander, and A. Gagalowicz, "Hierarchical region based stereo matching," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 416–421, 1989.
- [30] H. G. Mussmann, M. Hotter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing, Image Communication*, vol. 1, pp. 117–138, 1989.
- [31] T. Pavlidis and Y. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 225–233, 1990.
- [32] Kurita, "An efficient agglomerative clustering algorithm using a heap," *PATREC: Pattern Recognition*, Pergamon Press, vol. 24, pp. 205–209, 1991.

- [33] L. D. Griffin, A. C. F. Colchester, and G. P. Robinson, "Scale and segmentation of grey-level images using maximum gradient paths," *Image and Vision Computing*, vol. 10, pp. 389–402, 1992.
- [34] D. Noguet, A. Merle, and D. Lattard, "A data dependent architecture based on seeded region growing strategy for advanced morphological operators," in *Proceedings of In Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 235–243, 1996.
- [35] D. L. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Computation*, vol. 9, no. 4, pp. 805–836, 1997.
- [36] K. Haris, S. N. Efstratiadis, and N. Maglaveras, "Hybrid image segmentation using watersheds and fast region merging," *IEEE Transactions on Image Processing*, vol. 7, no. 12, pp. 1684–1699, 1998.
- [37] X. Yang and K. Ramchandran, "A low-complexity region-based video coder using backward morphological motion field segmentation," *IEEE Transactions on Image Processing*, vol. 8, no. 3, 1999.
- [38] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *Proceedings of International Conference on Computer Vision (2)*, pp. 926–932, 1999.
- [39] M. Levine and A. Nazif, "Dynamic measurement of computer generated image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 155–164, 1995.
- [40] F. Hansen and H. Elliott, "Image segmentation using simple Markov random field models," *Computer Graphics Image Processing*, vol. 20, pp. 101–132, 1982.
- [41] I. Herlin, D. Bereziat, G. Giraudon, C. Nguyen, and C. Graffigne, "Segmentation of echocardiographic images with Markov random fields," in *Proceedings of European Conference on Computer Vision (2)*, pp. 201–206, 1994.

- [42] K. Held, E. Kops, B. Krause, W. Wells, R. Kikinis, and H. Muller-Gartner, "Markov random field segmentation of brain MR images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, pp. 878–886, 1997.
- [43] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 648, IEEE Computer Society, 1998.
- [44] K. V. Leemput, F. Maes, F. Bello, D. Vandermeulen, A. C. F. Colchester, and P. Suetens, "Automated segmentation of MS lesions from multi-channel MR images," in *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*, pp. 11–21, 1999.
- [45] M. Svensen, F. Kruggel, and D. von Cramon, "Probabilistic modeling of single-trial fMRI data," *IEEE Transactions on Medical Imaging*, vol. 19, no. 1, pp. 25–35, 2000.
- [46] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, "The application of markov random field models to wavelet-based image denoising," pp. 43–70, 2001.
- [47] D. Xu, J.-N. Hwang, and C. Yuan, "Segmentation of multi-channel image with Markov random field based active contour model," *Journal of VLSI Signal Processing-Systems*, vol. 31, no. 1, pp. 45–55, 2002.
- [48] A. Ghosh, N. Pal, and S. Pal, "Image segmentation using a neural network," *Biological cybernetics*, vol. 66, pp. 151–158, 1991.
- [49] S. Cagnoni, G. Coppini, M. Rucci, D. Caramelia, and G. Valli, "Neural network segmentation of magnetic resonance spin echo images of the brain," *Journal of Biomedical Engineering*, vol. 15, no. 9, pp. 355–362, 1993.
- [50] A. Ossen, T. Zamzow, H. Oswald, and E. Fleck, "Segmentation of medical images using neural-network classifiers," in *Proceedings of International Con-*

- ference on Neural Networks and Expert Systems in Medicine and Healthcare*, pp. 427–432, 1994.
- [51] L. Jzau-Sheng, C. Kuo-Sheng, and M. Chi-Wu, “Segmentation of multispectral magnetic resonance image using penalised fuzzy competitive learning network,” *Computers and Biomedical Research*, vol. 29, pp. 314–326, 1996.
- [52] A. Hasegawa, S.-C. B. Lo, J.-S. Lin, M. T. Freedman, and S. K. Mun, “A shift-invariant neural network for the lung field segmentation in chest radiography,” *Journal of VLSI Signal Processing Systems*, vol. 18, no. 3, pp. 241–250, 1998.
- [53] J. O. Glass, W. E. Reddick, Y. V. Goloubeva, O., and R. G. Steen, “Hybrid artificial neural network segmentation of precise and accurate inversion recovery (PAIR) images from normal human brain,”
- [54] Z. Dokur, “Segmentation of MR and CT images using a hybrid neural network trained by genetic algorithms,” *Neural Processing Letters*, vol. 16, no. 3, pp. 211–225, 2002.
- [55] Y. Jiang and Z.-H. Zhou, “Som ensemble-based image segmentation,” *Neural Processing Letters*, vol. 20, no. 3, pp. 171–178, 2004.
- [56] H. Derin, H. Elliott, R. Cristi, and D. Geman, “Bayes smoothing algorithms for segmentation of binary images modeled by markov random fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 707 – 720, 1984.
- [57] M. Asker and H. Derin, “A recursive algorithm for the Bayes solution of the smoothing problem,” *IEEE Transactions on Automatic Control*, vol. 26, pp. 558 – 561, 1981.
- [58] W. Blanz and S. Gish, “A connectionist classifier architecture applied to image segmentation,” in *Proceedings of 10th International Conference on Pattern Recognition*, pp. 272–277, 1990.

- [59] A. Amini, T. Weymough, and J. R., "Using dynamic programming for solving variational problem in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855–867, 1990.
- [60] D. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol. 55, pp. 14–26, 1992.
- [61] K. Lam and H. Yan, "Fast greedy algorithm for active contours," *Electronics Letters*, vol. 30, no. 1, pp. 21–23, 1994.
- [62] G. Xu, E. Segawa, and S. Tsuji, "Robust active contours with insensitive parameters," *Pattern Recognition*, vol. 27, no. 7, pp. 879–884, 1994.
- [63] G. Iannizzotto and L. Vita, "Fast and accurate edge-based segmentation with no contour smoothing in 2-D real images," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1232–1237, 2000.
- [64] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [65] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [66] L. D. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, 1997.
- [67] A. Chakraborty and J. S. Duncan, "Game-theoretic integration for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 12–30, 1999.
- [68] Y. Chen, H. D. Tagare, S. Thiruvankadam, F. Huang, D. Wilson, K. S. Gopinath, R. W. Briggs, and E. A. Geiser, "Using prior shapes in geometric active contours in a variational framework," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 315–328, 2002.

- [69] A. Rexhepi, F. Mokhtarian, and A. Rosenfeld, "Sparse, variable-representation active contour models," in *Proceedings of 17th International Conference on Pattern Recognition, Volume 3*, pp. 683–686, Proceedings of IEEE Computer Society, 2004.
- [70] S. Osher and J. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [71] R. Malladi, J. Sethian, and B. Vemuri, "Shape modelling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, 1995.
- [72] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, no. 2, pp. 269–277, 1995.
- [73] R. T. Whitaker, "A level-set approach to 3D reconstruction from range data," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.
- [74] L. H. S. Jing Yang and J. S. Duncan, "Neighbor-constrained segmentation with level set based 3-d deformable models," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 940–948, 2004.
- [75] J. Sethian, "Tracking interfaces with level sets," *American Scientist*, vol. 85, no. 3, 1997.
- [76] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, "A PDE based fast local level set method," *Journal of Computational Physics*, vol. 155, pp. 410–438, 1999.
- [77] C. Samson, L. Blanc-Fraud, G. Aubert, and J. Zerubia, "A level set model for image classification," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 2000.

- [78] L.-T. Cheng, P. Burchard, B. Merriman, and S. Osher, "Motion of curves constrained on surfaces using a level-set approach," *Journal of Computational Physics*, vol. 175, no. 2, pp. 604–644, 2002.
- [79] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.
- [80] Fedkiw and Osher, *Level Set Methods and Dynamic Implicit Surfaces*. Berlin: Springer, 2002.
- [81] N. Lin, W. Yu, and J. S. Duncan, "Combinative multi-scale level set framework for echocardiographic image segmentation," *Medical Image Analysis*, vol. 7, no. 4, pp. 529–537, 2003.
- [82] J. E. Cates, A. E. Lefohn, and R. T. Whitaker, "GIST: an interactive, GPU-based level set segmentation tool for 3D medical images," *Medical Image Analysis*, vol. 8, no. 3, pp. 217–231, 2004.
- [83] I.B., Ayed, A. Mitiche, and Z. Belhadj, "Multiregion level-set partitioning of synthetic aperture radar images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 793 – 800, 2005.
- [84] B. Cernushi-Frias and D. Cooper, "Estimation of location and orientation of 3-D surfaces using a single 2-D image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 13–23, 1994.
- [85] D. Marr, "Analysis of occluding contour," in *Proceedings of of Royal Society, Ser. B*, vol. 197.
- [86] G. Greiner and H. Seidel, "Modeling with triangular B-splines," *Computer Graphics and Applications*, vol. 14, pp. 56 – 60, 1994.
- [87] F. S.Cohen and J. Wang, "Modeling image curves using invariant 3D object curve models - a path to 3D recognition and shape estimation from image

- contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1–23, 1994.
- [88] Z. Huang and F. Cohen, "Affine-invariant B-spline moments for curve matching," *IEEE Transactions on Image Processing*, vol. 5, pp. 1473–1480, 1996.
- [89] Y. Wang, D. Shen, and E. Teoh, "A novel lane model for lane boundary detection," in *Proceedings of IAPR Workshop on Machine Vision Applications*, pp. 27–30, 1998.
- [90] Y. Wang and E. Teoh, "Lane detection using B-snake," in *Proceedings of IEEE International Conference on Information, Intelligence and Systems, Washington DC*, 1999.
- [91] M. A. T. Figueiredo and J. M. N. Leitao, "Unsupervised contour representation and estimation using B-splines and a minimum description length criterion," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1075–1087, 2000.
- [92] Y.-H. Gu and T. Tjahjedi, "Coarse-to-fine planar object identification using invariant curve features and B-spline modeling," *Pattern Recognition*, vol. 33, pp. 1411–1422, 2000.
- [93] Y. Wang, D. Shen, and E. Teoh, "Lane detection using spline model," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 677–689, 2000.
- [94] Y. Wang, E. Teoh, and D. Shen, "Structure-adaptive B-snake for segmenting complex objects," in *Proceedings of International Conference on Image Processing*, pp. 769–772, 2001.
- [95] M. Unser, "Splines: a perfect fit for medical imaging," in *Proceedings of SPIE Symposium on Medical Imaging, San Diego, CA.*, 2002.
- [96] D. Terzopoulos and H. Qin, "Dynamic NURBS with geometric constraints for interactive sculpting," *ACM Transactions on Graphics*, vol. 13, no. 2, pp. 103–136, 1994.

- [97] D. Terzopoulos and H. Qin, "Dynamic sculpting of triangular NURBS objects," in *Proceedings of Fourth SIAM Conference on Geometric Design, Nashville, TN*, November 1995.
- [98] D. Terzopoulos and H. Qin, "D-nurbs," in *Proceedings of Third Pacific Conference on Computer Graphics and Applications, Seoul, Korea*, pp. 455–474, August 1995.
- [99] H. Qin and D. Terzopoulos, "D-NURBS: A Physics-Based Framework for Geometric Design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 1, pp. 85–96, 1996.
- [100] H. Qin and D. Terzopoulos, "Dynamic NURBS swung surfaces for physics-based shape design," *Computer-Aided Design*, vol. 27, no. 2, pp. 111–127, 1995.
- [101] D. Terzopoulos and H. Qin, "Dynamic NURBS with geometric constraints for interactive sculpting," *ACM Transactions on Graphics*, vol. 13, no. 2, pp. 103–136, 1994.
- [102] R. Meegama and J. Rajapakse, "NURBS-based segmentation technique for MR head scans," in *Proceedings of Indian Conference on Vision, Graphics and Image Processing*, pp. 376–381, 2002.
- [103] R. Meegama and J. Rajapakse, "NURBS-based detection of age-related variability of the human brain surface," in *Proceedings of IEEE International Conference Neural Information Processing, Singapore*, pp. 22–34, 2002.
- [104] R. Meegama and J. Rajapakse, "NURBS snakes," *Image and Vision Computing*, vol. 21, no. 6, pp. 551–562, 2003.
- [105] R. Meegama and J. Rajapakse, "NURBS-based segmentation of the brain in medical images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 6, pp. 995–1009, 2003.

- [106] P. Saint-Marc, H. Rom, and G. Medioni, "B-spline contour representation and symmetry detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1191–1197, 1993.
- [107] M. Eck and H. Hoppe, "Automatic reconstruction of B-spline surfaces of arbitrary topological type," *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 325–334, 1996.
- [108] P. Radeva, A. Amini, and J. Huang, "Deformable B-solids and implicit snakes for 3D localization and tracking of SPAMM MRI data," *International Journal on Computer Vision and Image Understanding*, vol. 66, pp. 163–178, 1997.
- [109] M. Figueiredo, J. Leitao, and A. Jain, "Adaptive B-spline and boundary estimation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 724–729, 1997.
- [110] A. A. Amini, Y. Chen, R. Curwen, V. Mani, and J. Sun, "Coupled B-snake grids and constrained thin-plate splines for analysis of 2D tissue deformations from tagged MRI," *IEEE Transactions on Medical Imaging*, vol. 17, no. 3, pp. 344–356, 1998.
- [111] A. A. Amini, P. Radeva, M. Elayyadi, and D. Li, "Measurement of 3D motion of myocardial material points from explicit B-surface reconstruction of tagged MRI data," in *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*, pp. 110–118, 1998.
- [112] J. Huang, D. Abendschein, V. G. Davila-Roman, and A. A. Amini, "Spatio-temporal tracking of myocardial deformation with a 4D B-spline model from tagged MRI," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 957–972, 1999.
- [113] A. A. Amini, Y. Chen, M. Elayyadi, and P. Radeva, "Tag surface reconstruction and tracking of myocardial beads from SPAMM-MRI with parametric

- B-spline surfaces," *IEEE Transactions on Medical Imaging*, vol. 20, no. 2, pp. 94–103, 2001.
- [114] Y. Chen and A. A. Amini, "A MAP framework for tag line detection in SPAMM data using Markov random fields on the B-spline solid," *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, pp. 1110–1122, 2002.
- [115] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," *Image and Vision Computing*, vol. 12, no. 6, pp. 242–263, 1994.
- [116] P. Smyth, C. Taylor, and J. Adams, "Automatic measurement of vertebral shape using active shape models," in *Proceedings of British Machine Vision Conference*, pp. 705–714, 1996.
- [117] S. Solloway, C. Taylor, C. Hutchinson, and J. Waterton, "The use of active shape models for making thickness measurements from MR images," in *Proceedings of 4th European Conference on Computer Vision*, pp. 400–412, 1996.
- [118] S. Solloway, C. Hutchinson, J. Waterton, and C. Taylor, "The use of active shape models for making thickness measurements of articular cartilage from MR images," *Magnetic Resonance in Medicine*, vol. 37, no. 6, pp. 943–952, 1997.
- [119] G. Behiels, D. Vandermeulen, F. Maes, P. Suetens, and P. Dewaele, "Active shape model-based segmentation of digital X-ray images," in *Proceedings of Lecture Notes in Computer Science, Berlin Germany, MICCAI*, pp. 128–137, 1999.
- [120] F. Vogelsang, M. Kohnen, J. Mahlke, F. Weiler, M. W. Kilbinger, B. Wei, and R. Gunther, "Model based analysis of chest radiographs," in *Proceedings of SPIE Symposium on Medical Imaging*, pp. 1040–1052, 2000.

- [121] G. Hamarneh and T. Gustavsson, "Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images," *Computers in Cardiology*, vol. 27, pp. 115–118, 2000.
- [122] S. Mitchell, B. Lelieveldt, H. van der Geest, R. and Bosch, J. Reiber, and M. Sonka, "Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac MR images," *IEEE Transactions on Medical Imaging*, vol. 20, no. 5, pp. 415–423, 2001.
- [123] G. Behiels, F. Maes¹, D. Vandermeulen, and P. Suetens, "Evaluation of image features and search strategies for segmentation of bone structures in radiographs using active shape models," *Medical Image Analysis*, vol. 6, no. 1, pp. 47–62, 2002.
- [124] B. van Ginneken, A. Frangi, J. Staal, B. ter Haar Romeny, and M. Viergever, "Active shape model segmentation with optimal features," *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 924–933, 2002.
- [125] M. Kohnen, A. Mahnen, J. Kesten, E. Koepfel, R. Gunther, and B. Wein, "A three dimensional knowledge based surface model for segmentation of organic structures," in *Proceedings of SPIE Symposium on Medical Imaging*, vol. 4684, pp. 485–494, 2002.
- [126] T.F.Cootes, G. Edwards, and C.J.Taylor, "Active appearance models," in *Proceedings of European Conference on Computer Vision*, vol. 2.
- [127] T. Cootes, G. Wheeler, K. Walker, and C. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, pp. 657–664, 2002.
- [128] T. Cootes, G. J. Edwards, and C. J. Taylor, "A comparative evaluation of active appearance model algorithms," in *Proceedings of British Machine Vision Conference*, vol. 2.

- [129] G. Edwards, T.F.Cootes, and C.J.Taylor, "Face recognition using active appearance models," in *Proceedings of European Conference on Computer Vision*, vol. 2.
- [130] T. Cootes, K. Walker, and C.J.Taylor, "View-based active appearance models," in *Proceedings of International Conference on Face and Gesture Recognition*, pp. 227–232, 2000.
- [131] H. Bosch, S.C.Mitchell, P.F.Boudewijn, P.F.Lelieveldt, F.Nijland, O. Kamp, M. Sonka, and J. Reiber, "Active appearance-motion models for endocardial contour detection in time sequences of echocardiograms," in *Proceedings of SPIE Symposium on Medical Imaging*, vol. 4322, pp. 257–268, 2001.
- [132] B. Lelieveldt, S. Mitchell, R. van der Geest, H. Bosch, and J. R. M. Sonka, "Time continuous segmentation of cardiac MR images using active appearance motion models," in *Proceedings of Computer Assisted Radiology and Surgery*, pp. 917–921, 2001.
- [133] T. Cootes and P. Kittipanya-ngam, "Comparing variations on the active appearance model algorithm," in *Proceedings of British Machine Vision Conference*, pp. 837–846, 2002.
- [134] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "3-D active appearance models: Segmentation of cardiac MR and ultrasound images," *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, pp. 1167–1178, 2002.
- [135] J. G. Bosch, S. C. Mitchell, B. P. F. L. F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1374–1383, 2002.

- [136] R.H.Davies, C.J.Twining, T.F.Cootes, J.C.Waterton, and C.J.Taylor, "3D statistical shape models using direct optimisation of description length," in *Proceedings of European Conference on Computer Vision*, vol. 3.
- [137] M. Roberts, T. Cootes, and J. Adams, "Linking sequences of active appearance sub-models via constraints: an application in automated vertebral morphometry," in *Proceedings of British Machine Vision Conference*, vol. 1.
- [138] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proceedings of Fifth International Conference on Computer Vision*, pp. 840–845, June 1995.
- [139] T. McInerney and D. Terzopoulos, "Medical image segmentation using topologically adaptable snakes," in *Proceedings of First International Conference on Computer Vision*, vol. 905.
- [140] T. McInerney and D. Terzopoulos, "Topology adaptive snakes," *Medical Image Analysis*, vol. 4, pp. 73–91, 2000.
- [141] T. McInerney and D. Terzopoulos, "Medical image segmentation using topologically adaptable surfaces," in *Proceedings of First Joint Conference of Computer Vision, Virtual Reality, and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, vol. 1205.
- [142] J.-O. Lachaud and B. Taton, "Deformable model with adaptive mesh and automated topology changes," in *Proceedings of 4th International Conference on 3-D Digital Imaging and Modeling , Banff, Alberta, Canada* (M. Rioux, P. Boulanger, and G. Godin, eds.), IEEE Computer Society Press, 2003.
- [143] L. Staib and J. Duncan, "Deformable fourier models for surface finding in 3D images obtain a shape representation of the cortex," in *Proceedings of Second Conference on Visualization in Biomedical Computing*, vol. 1808, 1992.

- [144] C. Davatzikos, J. Prince, and X. Tao, "Using a statistical shape model to extract sulcal curves on the outer cortex of the human brain," *IEEE Transactions on Medical Imaging*, vol. 21, no. 5, pp. 513–524, 2002.
- [145] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1131–1147, 1993.
- [146] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 359–369, 1998.
- [147] M. R. Kaus, V. Pekar, C. Lorenz, R. Truyen, S. Lobregt, and J. Weese, "Automated 3-D PDM construction from segmented images using deformable models," *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 1005–1013, 2003.
- [148] J.-O. Lachaud and E. Bainville, "A discrete adaptive model following topological modifications of volumes," in *Proceedings of 4th Discrete Geometry for Computer Imagery, Grenoble, France*, pp. 183–194, Sep 1994.
- [149] J.-O. Lachaud and B. Taton, "Deformable model with adaptive mesh and automated topology changes," in *Proceedings of 4th International Conference 3-D Digital Imaging and Modeling*, 2003.
- [150] X. Han, C. Xu, U. Braga-Neto, and J. L. Prince, "Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm," *IEEE Transactions on Medical Imaging*, vol. 21, no. 2, pp. 109–121, 2002.
- [151] S. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Computer Graphics and Image Processing*, vol. 4, pp. 104–119, 1975.
- [152] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. COM-31,4, pp. 532–540, 1983.

-
- [153] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*.
Berlin: Springer-Verlag, 1988.