2D object segmentation and matching using B-spline model

Wang, Yue

2005

Wang, Y. (2005). 2D object segmentation and matching using B-spline model. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3707>

<https://doi.org/10.32657/10356/3707>

Nanyang Technological University

Downloaded on 26 Apr 2023 13:28:47 SGT

2D Object Segmentation and Matching Using B-Spline Model

Wang Yue

School of Electrical & Electronic Engineering

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

2005

TA
1637
W24
2005

Statement of Originality

I hereby certify that content of this dissertation is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

15 / 9 / 2005
Date

王越
Wang Yue

Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisor, Associate Professor Teoh Eam Khwang, for his enthusiastic encouragement, excellent guidance and support throughout the duration of this research work and the writing of this thesis. Without his assistance and direction, this research would not be possible.

I am also sincerely and deeply grateful to Assistant Professor Shen Dinggang of the Department of Radiology, School of Medicine, University of Pennsylvania, USA, for his many fruitful discussions and comments on this research.

Special thanks are dedicated to all staff and friends in the Intelligent Machines Research Laboratory at the School of Electrical and Electronic Engineering, Nanyang Technological University, for their logistic support and the help rendered.

I am greatly indebted to my family for their kind support throughout my academic life.

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 PhD study.

Summary

This thesis presents a research work on object segmentation and matching using a B-spline model. The B-spline is selected here as a curve representation to describe the object contour as it possesses very attractive properties such as compact representation, invariance under affine transformation, local shape controllability, smoothness and continuity. In this thesis, the problem of object segmentation is first addressed and followed by curve matching.

In object segmentation, we concentrate on developing a new knowledge-based B-snake model where a series of novel algorithms are proposed and implemented. Firstly, a cubic B-snake model is developed for 2-D object segmentation, with an adaptive control point insertion algorithm that is suggested to increase the flexibility of B-snake to describe complex shape. A minimum energy method, which is called Minimum Mean Square Error (MMSE), is proposed for B-snake to push it to the target boundary. The internal forces are not required in deforming B-snake since the representation of B-spline has guaranteed smoothness by hard implicit constraints. This method overcame the problems that existed in other B-spline based models. These models require the number of control points to be set beforehand and hence have less flexibility to describe unknown complex shapes. The proposed B-snake model has been tested on object contour extraction such as human brain ventricle in Magnetic Resonance (MR) images and complex lane detection in 2D road images. The experimental results demonstrate the capability of structure-adaptive shape description and the potential power of the proposed model used for object segmentation.

Next, in order to use statistical information on the geometry and appearance to improve the B-snake model in object segmentation, two algorithms, called the Dynamic B-snake

Model (DBM) and the Active B-snake Model (ABM), are developed for modeling the features of the object in training set and guiding the B-snake in the deforming procedure.

In DRM, the samples in the given training set are reconstructed by a fixed number of knot points and they are aligned before one performs statistics on them. We use the affine-invariant alignment method, where the attribute vectors with different levels of geometric information are employed. By using an error minimization process, the affine transformation between two sets of knot points of B-snakes can be established. A statistical model is then suggested which is able to constrain the deformation of B-snake into the expected statistical shape space of the training samples.

In ABM, both the statistical geometric information and the appearance information are extracted. An affine-invariant landmark point assignment strategy is proposed for B-snake to extract the geometric structure and appearance information of the object being studied. Compared to other deformable models which require manual assignment of landmark points, such as Active Shape Model (ASM), our method is fully automatic and greatly reduces the processing time. A statistical framework is embedded into ABM by combining the Principal Component Analysis (PCA) for modeling the shape distribution and appearance characteristics of landmark points in the training samples. The ABM can keep its geometric shape and search for the appearance similarly to those in the training set during the process of segmentation. This helps to significantly improve the robustness of the search capability and avoid getting trapped in the local minima. The final positions of the control points are refined by the proposed knot insertion strategy and MMSE approach. It accommodates fine deformation that allows the B-snake to accurately adapt to the desired object boundaries in the image.

The two models have been implemented for the segmentation of human brain ventricles in real MR images with a training set of 105 samples. Both qualitative and quantitative comparisons with traditional snake and ASM are presented. The experimental results show both models have a robust capability to describe complex object contours while requiring only a few parameters. It can also obtain a more accurate object boundary in object segmentation over the traditional snake and ASM.

In the area of curve matching, a new affine-invariant matching algorithm based on B-spline smoothing and Curvature Scale Space (CSS) matching is proposed. The new algorithm overcomes the problem of non-uniqueness of the control points of B-spline curve, which means that different sets of control points of B-spline may represent the same shape and makes the curve recognition process very difficult when comparing the respective control points of the B-spline to be matched. Our method combines the advantages of the B-spline, i.e. continuous curve representation and affine invariance, and the robustness of the CSS matching with respect to noise and affine transformation. In addition, it avoids other matching algorithms that have to use re-sampled points on the curve, thus reducing the curve matching error. The proposed method has been simulated for finding the similar shapes from a prototype database which contains a vast variety of shapes of marine creatures. The experimental results show the efficiency of the proposed method in B-spline curve smoothing and affine-invariant CSS image construction, as well as the robustness in B-spline curve matching without estimation of affine transformation.

In summary, this thesis presents the research work on object segmentation using B-snake model and object matching using B-spline model. A novel 2-D B-snake model is developed with an adaptive control point insertion algorithm and a MMSE method. Moreover, two algorithms, DBM and ABM, are presented for modeling the features of object in the training set and guiding the B-snake in the object segmentation. Finally, an affine-invariant B-spline curve matching algorithm is proposed to overcome the problem of non-uniqueness of control point of B-spline curve. Our future work will focus on extending our models from 2-D to 3-D for object segmentation and matching.

Contents

Acknowledgments	i
Summary	ii
List of Figures	x
List of Tables	xv
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Analysis of Medical Image	1
1.1.2 Object Shape Matching	4
1.2 Objectives	5
1.3 Original Contributions	6
1.4 Thesis Organization	7
2 A Literature Review	9
2.1 Introduction.....	9
2.2 Object Segmentation – An Overview	10
2.2.1 Histogram Thresholding.....	10
2.2.2 Region Growing.....	11
2.2.3 Classifier	12
2.2.4 Markov Random Field Models	14
2.2.5 Artificial Neural Networks.....	15

Contents	vi
2.2.6 Deformable Models.....	17
2.2.7 Atlas-Guided Approaches	17
2.2.8 Other Approaches.....	19
2.3 Deformable Models for 2D Object Segmentation	19
2.3.1 Point-Based Deformable Model.....	20
2.3.2 B-Spline Based Deformable Models.....	31
2.4 Object Curve Matching.....	41
2.4.1 Curve Matching Using B-Spline Model	42
2.5 Concluding Remarks	46
3 B-Spline and B-Snake Models – An Overview	48
3.1 Introduction.....	48
3.2 On B-Spline Model.....	49
3.2.1 Uniform Cubic B-Splines.....	49
3.2.2 Insertion of Knot Points	58
3.2.3 Why B-Splines ?	60
3.3 On Active Contour - Snake.....	62
3.4 On the B-Snake Model	64
3.5 Application of B-Snake for Object Boundary Detection.....	66
3.6 Concluding Remarks	68
4 Estimation of B-Snake Parameters from Image Data	69
4.1 Introduction.....	69
4.2 Shape Extraction Algorithm Using B-Snake.....	71
4.2.1 Gradient Vector Flow.....	71

Contents	vii
4.2.2 External Forces for B-Snake	73
4.2.3 Minimum Mean Square Error Approach	76
4.2.4 Adaptive Knot Point Insertion Strategy	79
4.2.5 The Complete Object Contour Detection Algorithm	80
4.3 Experimental Results and Discussion.....	80
4.3.1 Experimental Results on Complex Lane Detection - The Case of Open B-Snake Model.....	81
4.3.2 Experimental Results on Object Detection - The Case of Closed B- Snake Model.....	91
4.4 Concluding Remarks	98
5 Dynamic B-Snake Model (DBM) for Object Segmentation	99
5.1 Introduction.....	99
5.2 B-Snake Model Using Statistical Geometric Information.....	100
5.2.1 Re-Construction of B-Spline Curve	101
5.2.2 Shape Alignment Strategy.....	103
5.2.3 Modeling the Shape Distribution from the Training Set.....	107
5.3 The Complete DBM Algorithm.....	109
5.4 Experimental Results and Discussion.....	110
5.4.1 Experimental Results on Object Segmentation.....	110
5.4.2 Experiments on Comparing DBM with the Traditional Snake and ASM	116
5.5 Concluding Remarks	118
6 Active B-Snake Model (ABM) for Object Segmentation	120
6.1 Introduction.....	120

Contents	viii
6.2 Active B-Snake Model (ABM).....	121
6.2.1 Landmark Assignment for B-Snake Model	121
6.2.2 Extraction of Statistical Appearance and Geometric Information	131
6.2.3 The Complete Algorithm for Object Segmentation Using ABM	134
6.3 Experimental Results and Discussion.....	135
6.3.1 Experimental Results of Landmark Point Assignment and Shape Alignment.....	136
6.3.2 Experimental Results Using Active B-Snake Model (ABM)	138
6.3.3 Comparison of ABM with the Traditional Snake and ASM.....	142
6.4 Concluding Remarks	144
7 B-Spline Model for Object Contour Matching	146
7.1 Introduction.....	146
7.2 Curvature Scale Space Matching.....	148
7.3 The Algorithm for B-Spline Curve Matching	151
7.3.1 The Algorithm for B-Spline Curve Smoothing.....	152
7.3.2 Construction of The CSS Image	157
7.3.3 Extraction of Maxima of CSS Contour	160
7.3.4 Curvature Scale Space Matching	161
7.4 Experimental Results and Discussion.....	162
7.4.1 Experimental Result on Affine Invariant Construction of CSS Image	162
7.4.2 Experimental Result on Object Curve Matching	166
7.5 Concluding Remarks	172
8 Conclusion and Recommendations	173

Contents	ix
8.1 Conclusion	173
8.2 Recommendations for Future Research	175
Author's Publications	178
Bibliography	180

List of Figures

Figure 1-1 A transaxial MR image of the human brain.....2

Figure 2-1 Lumen extraction using region growing. 11

Figure 2-2 Results of adaptive segmentation..... 13

Figure 2-3 Segmentation of the MR image. 15

Figure 2-4 Unsupervised segmentation using ANN..... 16

Figure 2-5 Sulci detection using Atlas-guided approach..... 18

Figure 2-6 Search the cartilage using Active Shape Model on an MR image of
the knee 21

Figure 2-7 An example on endocardium detection in MR heart image..... 22

Figure 2-8 A right thalamus segmentation. 23

Figure 2-9 A labeled training image gives a shape-free patch and a set of points
in the AAM 24

Figure 2-10 The face AAM approaches to each face 25

Figure 2-11 Two examples of ventricle in the MR images where the AAM fails..... 25

Figure 2-12 The left and right field of lung segmentation using the ASM with
optimal features..... 26

Figure 2-13 Contour extraction using wavelet deformable model in CT image. 27

Figure 2-14 Segmentation with T-snake in an MR brain image..... 28

Figure 2-15 Brain segmentation 29

Figure 2-16 Two examples of optic disk shape detection..... 30

List of Figures	xi
Figure 2-17 An open B-snake model for vessel boundaries approaching.	32
Figure 2-18 Figure illustrating the PERM strategy. (Adopted from [105].).....	33
Figure 2-19 Results of PERM strategy. (Adopted from [105].)	33
Figure 2-20 Lane detection using B-snake model. (Adopted from [127].)	34
Figure 2-21 Evolution of the contour shape for 5, 8, 10 and 11 control points.....	35
Figure 2-22 The description length corresponding to the example in Figure 2-20.	36
Figure 2-23 Multi-resolution approach vs. single-resolution approach.....	37
Figure 2-24 PET-scan of the brain.....	37
Figure 2-25 Movement of a curve point $v(s)$ towards the desired control point when the associated weight is increased	38
Figure 2-26 Vessel boundary detection results obtained with the proposed algorithm [111] on the trauma sequence.....	39
Figure 2-27 The inflection points on the B-spline curve	44
Figure 2-28 Boundaries of objects with marked corners	45
Figure 3-1 An open cubic B-spline curve.....	50
Figure 3-2 Uniform cubic B-spline basic functions.....	52
Figure 3-3 A closed cubic B-spline curve.....	53
Figure 3-4 Forming a corner by tripling a control point.....	54
Figure 3-5 B-spline needs to insert a new knot point.	59
Figure 3-6 B-spline after knot point insertion.	59
Figure 3-7 Comparison of two B-splines curves before and after control points insertion.....	60
Figure 3-8 Using B-snake to detect the ventricle in MR image.	68
Figure 4-1 Canny edge detection.	71

List of Figures	xii
Figure 4-2 External force using the GVF. The arrows represent the direction and magnitude of the GVF force.....	73
Figure 4-3 Demonstration of $\hat{F}_B(s)$	74
Figure 4-4 External force $F_B(s)$	75
Figure 4-5 External forces $\hat{F}_B(s)$	75
Figure 4-6 An initialization of B-snake lane model for “S” shape road with painted lane.	84
Figure 4-7 Deformation of B-snake with three control points.....	85
Figure 4-8 Result of B-snake with three control points.....	85
Figure 4-9 The magnitude of external forces on the B-snake with three control points.....	86
Figure 4-10 One new knot point (or control point) has been inserted.....	86
Figure 4-11 B-snake deformation with four control points.	87
Figure 4-12 Final result of B-snake with four control points.	87
Figure 4-13 The magnitude of external forces on the B-snake with four control points.....	88
Figure 4-14 Lane detection for wet and painted road.	89
Figure 4-15 Lane detection for unpainted lane.....	90
Figure 4-16 The closed B-snake with four control points	91
Figure 4-17 Result of the B-snake with 4 control points.....	92
Figure 4-18 External force of the B-snake.....	92
Figure 4-19 Result of B-snake deformation using 5 control points.....	93
Figure 4-20 Results of B-snake deformation with 5 to 17 control points.....	94
Figure 4-21 External force of the B-snake using 17 control points.....	96
Figure 4-22 B-snake used to extract the object contour in noisy background.....	97

List of Figures	xiii
Figure 5-1 A re-constructed B-spline with 40 control points.	102
Figure 5-2 The shapes of B-spline curve with different number of control points.	102
Figure 5-3 Two more re-constructed B-spline curves.	103
Figure 5-4 Illustration of the concept of the “attribute vector” on the i th knot point.	104
Figure 5-5 Alignment results	107
Figure 5-6 Some aligned results of B-snake model.	107
Figure 5-7 Deformation of the DBM in ventricle segmentation.	111
Figure 5-8 External force of B-snake in Figure 5-7(f).	112
Figure 5-9 Some contour extraction results of different MR brain images using DBM.	113
Figure 5-10 A failed case of DBM.	115
Figure 5-11 External force of B-snake in Figure 5-10.	115
Figure 5-12 Qualitative Comparison of DBM with the traditional snake and ASM.	117
Figure 6-1 Inflection points on the B-spline curve.	126
Figure 6-2 Demonstration of A_i and α_i	127
Figure 6-3 The result of removing the faulty inflection points.	129
Figure 6-4 Two examples of removing the faulty inflection points.	129
Figure 6-5 Landmark point assignment.	131
Figure 6-6 Illustration of the concept of the “attribute vector” on the i -th landmark point.	132
Figure 6-7 Aligned and affine transformed images.	134
Figure 6-8 Examples on landmark point assignment.	137

List of Figures	xiv
Figure 6-9 Some aligned results of B-snake model.....	138
Figure 6-10 ABM in ventricle segmentation.	139
Figure 6-11 The external force of B-snake after convergence.	140
Figure 6-12 More segmentation results of different MR brain images using ABM.....	141
Figure 6-13 Qualitative comparison of ABM with the traditional snake and ASM.	143
Figure 7-1 CSS image construction	150
Figure 7-2 Example of degree increasing of B-spline curve.	153
Figure 7-3 Example on smoothing the B-spline curve based on the proposed method.....	157
Figure 7-4 CSS image construction for a fish shape.	159
Figure 7-5 The CSS image for the affine transformed shape of Figure 7-4(a).....	164
Figure 7-6 Comparison of the maxima of CSS of the shapes with and without affine transformed.	165
Figure 7-7 Matching results of shark contour.....	167
Figure 7-8 Matching result of eel contour.	169
Figure 7-9 Matching result of a fish contour.	170
Figure 7-10 Matching result of a fish contour.	171

List of Tables

Table 5.1 Quantitative comparison in term of DBM with the traditional snake and ASM based on the results presented in Figure 5-12..... 118

Table 6.1 Quantitative Comparison in term of ABM with the traditional snake and ASM based on the results presented in Figure 6-13..... 144

Table 7.1 The Maxima of CSS contour presented in Figure 7-4(i)..... 160

Table 7.2 The Maxima of CSS contour presented in Figure 7-5(b)..... 165

Table 7.3 Matching values of Figure 7-7(a) to Figure 7-7(b)-(g). 167

Table 7.4 Matching values of Figure 7-8(a) to Figure 7-8(b)-(g). 169

Table 7.5 Matching values of Figure 7-9(a) to Figure 7-9(b)-(g). 170

Table 7.6 Matching values of Figure 7-10(a) to Figure 7-10(b)-(g). 171

Chapter 1

Introduction

Over the past decade, there has been increasing research activity on object segmentation and object matching for image analysis, computer vision, and medical imaging. Object segmentation is aimed at helping us to understand the various properties of objects in the images, while object shape matching is aimed at measuring how similar between object shapes. Both of object segmentation and matching require deriving a mathematical description of object boundaries from images, this work is also known as a fundamental step in image processing and computer vision. The applications of object segmentation and matching include motion tracking, shape matching and identification, object recognition, and image registration and warping.

1.1 Background and Motivation

Below the brief introduction of background of object segmentation and matching is given.

1.1.1 Analysis of Medical Image

With the increasingly important role of medical imaging playing in the diagnosis and treatment of disease, the medical image analysis has become more concentrated on the challenging problem of organ segmentation, with the assistance of computers, clinically useful information about anatomic structures imaged through Computed Tomography

(CT), Magnetic Resonance (MR), Positron Emission Tomography (PET), ultrasound, X-ray projection radiography and other modalities. Although modern imaging devices provide exceptional views of internal anatomy, the use of computers to quantify and analyze the embedded structures with accuracy and efficiency is limited. Accurate, repeatable, quantitative data must be efficiently extracted in order to support the spectrum of biomedical investigations and clinical activities from diagnosis, to radiotherapy and surgery.

Object segmentation remains a difficult task due to the variability of object shapes and diverse image sources. For example, one important task in medical imaging is the boundary extraction of the brain ventricle from MR images (Figure 1-1). Image noise and sampling artifacts especially may cause the boundaries of objects of interest to be indistinct and disconnected. How to integrate these boundaries into a coherent and consistent mathematical description is a challenging problem that a boundary mapping technique has to address.

Figure 1-1 A transaxial MR image of the human brain.

Object segmentation methods abound in the literature of image analysis, computer vision, and medical imaging. Dedicated solutions for the segmentation of a wide range of objects in a wide range of images have been proposed. These methods for medical image segmentation include deformable models, region growing, Markov random field

models, artificial neural networks, and many others [1]. The more details will be given in next chapter. The current research in the segmentation of medical images will strive towards improving the accuracy, precision, and computational speed of segmentation methods, as well as reducing the amount of manual interaction. Accuracy and precision can be improved by incorporating prior information from atlases and by combining discrete and continuous-based segmentation methods.

Active contour model, also called snake, is a deformable model first introduced by Kass *et al.* [2] as a new technique for segmentation. The snake is described as a point-based energy minimizing deformable model and guided by external and internal forces to approach object boundaries. Such models consist of a large number of disconnected points to define the curve, and have the large number of coefficients to be optimized. From then on, it has been studied intensively. Such models, which can adapt themselves to fit objects more closely, have been demonstrated to be effective in object segmentation.

Based on the original approach of the snake, B-spline has been implemented as a representation of the curve. Such a formulation of a deformable model has several advantages, such as compact representation, shape invariance under affine transformation, local controllability, smoothness and continuity.

However, currently there are very few papers that have been published for object segmentation using B-spline model. This is due to the problems listed below:

- (1) In the B-snake Model, it is difficult for the user to decide the number of control points for B-spline. The more control points used, the more complex shape that B-spline curve could form. However, the choice of the number of control points to adapt to the structure of the object still remains a challenge.
- (2) To improve deformable model, the statistical information extracted from a training set can be used to guide the deformation of the snake. However, implementation of the prior knowledge extracted from a training set in the B-snake model is an open problem.

Motivated by above mentioned problems, we extend the basic concept of B-snake and develop a new knowledge-based B-snake model for object segmentation in order to

improve the efficiency, speed and flexibility for B-snake. The new B-snake model should possess the capabilities listed below:

- (1) It is able to perform the control point insertion to adapt the structure of object rather than just fix the number of control points.
- (2) It is able to incorporate *a priori* knowledge about the shape to be segmented, rather than to just let the snake be attracted by edges or ridges in the image. The various features about the object, such as statistical geometry and intensity information, can be embedded in the model to perform the global optimization.

1.1.2 Object Shape Matching

Object shape matching deals with transforming a shape, and measuring the resemblance with another one, using some similarity measure. It is a very important ingredient in shape retrieval, object recognition and classification, alignment and registration, and approximation and simplification. Shape matching or discrimination refers to methods for comparing shapes. Shape similarity measurements are an essential ingredient in shape matching. Although the term “similarity” is often used, “dissimilarity” corresponds to the notion of distance: small distance means small dissimilarity, and large similarity. It is used in model-based object recognition where a set of known model objects is compared to an unknown object detected in the image. For this purpose, a shape descriptor vector should be sought for each model shape and unknown shape in the scene. The unknown shape is matched to one of the model shapes by comparing the shape descriptor vector using a metric. Application of shape matching include industrial inspection, fingerprint matching and content-based image retrieval.

However, up to the present, the B-spline curve has been seldom used for matching or recognition purposes. This is because the B-spline curve is not uniquely described by a single set of control points which makes the curve matching process difficult when comparing the respective parameters of the curves to be matched.

Motivated by above mentioned problems, a novel algorithm is developed as well for affine-invariant curve matching using the B-spline, in spite of the problem of the non-

uniqueness property of the B-spline curve. This algorithm should be able to employ the advantage of the B-spline which gives continuous expression of the curve, is affine invariant, and avoids performing the curve matching based on the discrete re-sampled points on the curve.

1.2 Objectives

In this research project, one of our goals is to construct a new B-snake model for extracting objects from 2D images. Compared to other B-spline based model, this new B-snake model shall possess the capability to describe more complex shape by adaptively increasing the number of control point. Therefore, a strategy of automatic control-point insertion, adaptive to the structure of the studied object, should be developed. It should be simple and fast, and be able to use the existing condition. It should be able to be used in the B-snake deformation procedure.

The method for this B-snake model should be explored to extract the object boundaries. A minimum energy method shall be investigated for pushing the B-snake model to desired object edges in the image. A Minimum Mean Square Error (MMSE) approach has been developed in [127] for an open B-snake lane model. We will extend this MMSE approach to the closed B-snake model.

Moreover, to avoid getting trapped at the local minimum and achieve the global approach, an automatic landmark point assignment algorithm shall be developed for extracting the statistical information of various object features and guiding the deformation of B-snake toward the desired object in the images. This landmark point assignment algorithm should be based on those features that are affine invariant.

Furthermore, a statistical framework should be embedded into our B-snake model in order to use the prior knowledge of the studied object. Constrained by the prior geometric knowledge that reflects shape and appearance characteristics of the model around each landmark point, the new B-snake model should be able to keep its geometric shape and search for similar appearance during the procedure of segmentation. Associated with the knot insertion strategy and MMSE approach, the fine deformation

should be achieved for allowing the B-snake to accurately adapt to the desired object boundary in the image.

Finally, a new affine invariant curve matching algorithm will be developed using B-spline as curve representation. This new matching algorithm shall employ the advantages of the B-spline that are continuous curve representation and affine invariant. Moreover, to reduce the curve matching error, the algorithm should be based on the whole curve rather than those discrete sampled points. Here an affine invariant curve descriptor shall be used and then the matching performed. This algorithm should be applicable and efficient to curve matching in large scale database.

1.3 Original Contributions

This thesis deals with object segmentation using B-snake model and object matching using B-spline model. In particular, the author's contributions can be listed as follows:

- (1) A novel B-snake based deformable model is developed for closed 2-D object segmentation using Gradient Vector Flow (GVF) as B-snake external force. A minimum energy method of MMSE, which finds the correspondence between B-snake and the real edge image, is presented to determine the parameters of B-spline iteratively.
- (2) A simple and faster control point insertion algorithm is proposed for the B-snake model to increase the flexibility of describing complex shapes. In this algorithm, a control point is adaptively added by checking the projection of external force at its normal axis of each sampled point on the B-spline. This method overcomes the problems that exist in other B-spline based models that previously require a decision on the number of control points of the B-spline and hence, less flexible to describe unknown complex shapes. Compared to other methods existing in curve modeling for determining the number of control points, it is simple and largely reduces the processing time.
- (3) In order to use statistical information of geometry for improving the B-snake model in object segmentation, an algorithm called the Dynamic B-snake Model (DBM) is

developed. In the DBM, the samples in the given training set are reconstructed by a fixed number of knot points and then aligned before performing statistics on them. An affine-invariant alignment method is used. A statistical model is then suggested, which is able to constrain the deformation of B-snake into the expected statistical shape space of the training samples. The experimental results show that the DBM can achieve superior extraction results of human brain ventricle over the traditional snake and ASM in real MR brain images.

- (4) In order to improve the B-snake model in object segmentation, an algorithm called the Active B-snake Model (ABM) is developed. In the ABM, both the statistical geometric and the appearance information are extracted. A fully automatic affine-invariant landmark point assignment strategy is proposed. Moreover, a statistical framework is embedded into the ABM by combining the Principal Component Analysis (PCA) for modeling the shape distribution and appearance characteristics of landmark points in the training samples. The ABM can keep its geometric shape and search for appearances similar to those in the training set during the process of segmentation. This helps to significantly improve the robustness of search capability and avoid the local minima. The experimental results show that the ABM can achieve superior extraction results of human brain ventricle over the traditional snake and ASM and it is very robust to avoid the local minima.
- (5) A new affine-invariant curve matching algorithm based on a new B-spline smoothing algorithm and Curvature Scale Space (CSS) matching is proposed. It combines the advantages of the B-spline that are continuous curve representation and affine invariant, and the robustness of the CSS matching with respect to noise and affine transformation. In addition, it avoids the use of re-sampled points on the curve like other curve matching algorithms. Thus, it reduces the curve matching error. The experimental results show the proposed algorithm is promising.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- **Chapter 2** The review on the existing segmentation and matching techniques for the B-snake or B-spline curve is presented, and the analysis of the shortcomings of each technique is also given.
- **Chapter 3** An overview of the B-spline curve is first presented in this chapter. A novel B-spline model is then proposed. The concept of the B-snake is also introduced in this chapter.
- **Chapter 4** A minimum energy method called MMSE is developed for accurate object boundary extraction by iteration. GVF is applied here as the B-snake's external force. An adaptive knot insertion strategy is proposed to increase the flexibility of B-snake. The various experimental results of the proposed B-snake are presented as well.
- **Chapter 5** The DBM, which uses the statistical geometric information to guide the deformation, is presented in this chapter. The method for the B-spline curve reconstruction is presented first, followed by a shape alignment strategy and a shape distribution modeling by the PCA. The experimental results and the comparison of the proposed B-snake model with the traditional snake and ASM are given as well.
- **Chapter 6** The ABM is presented in this chapter. A novel landmark point assignment algorithm is proposed first, then a statistical framework for modeling the shape geometry and appearance, and the complete algorithm of the ABM is presented. The experimental results and the comparison of the ABM with the traditional snake and ASM is also presented in this chapter.
- **Chapter 7** A novel affine invariant shape matching algorithm using B-spline as curve representation is presented. The original CSS matching algorithm is also introduced. The experimental results using the proposed matching algorithm for finding the similar shapes from a database of marine animals with a vast variety of shapes are also provided.
- **Chapter 8** This chapter provides the conclusion of the thesis and recommends the direction of future work in this area.

Chapter 2

A Literature Review

2.1 Introduction

In Chapter 1, a global view of this research work was given with discussion on motivations, objectives and major contributions. However, in order to relate and compare our contributions with the existing techniques, it is necessary to discuss relevant works in details. In this chapter, detailed discussion on object segmentation and curve matching using B-spline model is presented.

As we know, segmentation and matching of objects of interest are of prime importance in the application of image processing and machine vision. The goal of segmentation is the process to extract the object from the image while object matching is to find the similar objects from a database for an input object. Model-based methods implement a prior model of object being studied in the image and typically make an effort to search for the best match of the model to the data from a new image. Once the model is matched, the measurement or test is performed to check whether the desired object is actually present. The model-based techniques are “top-down” methods which are in contrast to “bottom-up” or “data-driven” methods [4]-[9][81][101][131]. In the latter methods, the local structures such as edges are searched at a low level in the image, and are clustered together in an attempt to identify the object of study. Many varieties of model-based methods have been proposed [15]-[18][139]-[150]. This thesis will concentrate on the B-spline model.

In this chapter, the deformable model for object segmentation is introduced and followed with an emphasis of the B-spline deformable model. A review for the object matching algorithms using the B-spline model is then presented.

2.2 Object Segmentation – An Overview

Segmentation is an important topic in the field of image analysis. Object segmentation methods abound in the literature of image analysis, computer vision, and medical imaging. Dedicated solutions for the segmentation of a wide range of objects in a wide range of images have been proposed [1]. These methods for medical image segmentation are described in the following subsections.

2.2.1 Histogram Thresholding

Histogram thresholding approaches segment scalar images by creating a binary partitioning of the image intensities. A thresholding procedure attempts to determine an intensity value, called the threshold, which separates the desired classes. The segmentation is then achieved by grouping all pixels with intensity greater than the threshold into one class, and all other pixels into another class. Two potential thresholds can be determined by the valleys of the histogram. Determination of more than one threshold value is a process called multithresholding [3]. Histogram thresholding is a simple yet often effective means for obtaining a segmentation in images where different structures have contrasting intensities or other quantifiable features. The partition is usually generated interactively, although automated methods do exist [3]. For scalar images, interactive methods can be based on an operator's visual assessment of the resulting segmentation since the thresholding operation is implementable in real-time. Histogram thresholding is often used as an initial step in a sequence of image processing operations. Its main limitations are that in its simplest form only two classes are generated and it can not be applied to multi-channel images. In addition, thresholding typically does not take into account the spatial characteristics of an image. This causes it to be sensitive to noise and intensity inhomogeneities, which can occur in magnetic resonance images. Both these artifacts essentially corrupt the histogram of the image,

making separation more difficult. For these reasons, variations on classical thresholding have been proposed for medical image segmentation that incorporate information based on local intensities [19][20], connectivity [21] and entropy [22].

2.2.2 Region Growing

Region growing [9]-[14] [23]-[31] is a technique for extracting a region of the image that is connected based on some predefined criteria. These criteria can be based on intensity information and/or edges in the image [23][24]. In its simplest form, region growing requires a seed point that is manually selected by an operator, and extracts all pixels connected to the initial seed with the same intensity value. Figure 1-1 shows an example of lumen extraction using region growing in [24]. Like thresholding, region growing is not often used alone but within a set of image processing operations, particularly for the delineation of small, simple structures such as tumors and lesions [25][26].

Figure 2-1 Lumen extraction using region growing. (a) The original endoscopic image. (b) the result of lumen extraction. (Adopted from [24].)

Its primary disadvantage is that it requires manual interaction to obtain the seed point. Thus, for each region that needs to be extracted, a seed must be planted. Split and merge algorithms are related to region growing but do not require a seed point [3]. Region growing can also be sensitive to noise, causing extracted regions to have holes or even

become disconnected. Conversely, partial volume effects can cause separate regions to become connected. To help alleviate these problems, a homotopic region growing algorithm has been proposed that preserves the topology between an initial region and an extracted region [27][28]. Fuzzy analogies to region growing have also been developed [29]-[31].

2.2.3 Classifier

Classifier methods are pattern recognition techniques that seek to partition a feature space derived from the image using data with known labels [32][33]. A feature space is the range space of any function of the image, with the most common feature space being the image intensities themselves. A histogram is an example of a 1D feature space. All pixels with similar associated features would be grouped into one class. Classifiers are known as supervised methods since they require training data that are manually segmented and then used as references for automatically segmenting new data.

Standard classifiers require that the structures to be segmented possess distinct quantifiable features. Because training data can be labeled, classifiers can transfer these labels to new data as long as the feature space sufficiently distinguishes each label as well. Being non-iterative, they are relatively computationally efficient and unlike thresholding methods, they can be applied to multi-channel images. A disadvantage of classifiers is that they generally do not perform any spatial modeling. An adaptive segmentation method in [34] has addressed this weakness and extended classifier methods to segmenting images that are corrupted by intensity inhomogeneities. See Figure 2-2 for an example. Neighborhood and geometric information were also incorporated into a classifier approach in [35]. Another disadvantage is the requirement of manual interaction for obtaining training data. Training sets can be acquired for each image that requires segmenting, but this can be time consuming and laborious. On the other hand, use of the same training set for a large number of scans can lead to biased results which do not take into account anatomical and physiological variability between different subjects.

Clustering algorithms essentially perform the same function as classifier methods without the use of training data. Thus, they are called unsupervised methods. In order to

compensate for the lack of training data, clustering methods iterate between segmenting the image and characterizing the properties of the each class. In a sense, clustering methods train themselves using the available data.

Figure 2-2 Results of adaptive segmentation: (a) white matter probability and (b) segmentation. (Adopted from [34].)

Three commonly used clustering algorithms are the K -means algorithm [36], the fuzzy: C -means algorithm [33], and the expectation-maximization (EM) algorithm [37][38]. The K -means clustering algorithm clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean [47]. The fuzzy: C -means algorithm generalizes the K -means algorithm [33], allowing for soft segmentations based on fuzzy set theory [74]. The EM algorithm applies the same clustering principles with the underlying assumption that the data follows a Gaussian mixture model. It iterates between computing the posterior probabilities and computing maximum likelihood estimates of the means, covariances, and mixing coefficients of the mixture model.

Although clustering algorithms do not require training data, they do require an initial segmentation (or equivalently, initial parameters). The EM algorithm has demonstrated greater sensitivity to initialization than the K -means or fuzzy: C -means algorithms [42]. Like classifier methods, clustering algorithms do not directly incorporate spatial modeling and can therefore be sensitive to noise and intensity inhomogeneities. This lack

of spatial modeling, however, can provide significant advantages for fast computation [45]. Work on improving the robustness of clustering algorithms to intensity inhomogeneities in MR images has demonstrated excellent success [44][55]. Robustness to noise can be incorporated using the Markov random field modeling as described in the next part.

2.2.4 Markov Random Field Models

Markov random field (MRF) modeling itself is not a segmentation method but a statistical model which can be used within segmentation methods. MRFs model spatial interactions between neighboring or nearby pixels. These local correlations provide a mechanism for modeling a variety of image properties [50]. In medical imaging, they are typically used to take into account the fact that most pixels belong to the same class as their neighboring pixels. In physical terms, this implies that any anatomical structure that consists of only one pixel has a very low probability of occurring under a MRF assumption.

MRFs are often incorporated into clustering segmentation algorithms such as the K -means algorithm under a Bayesian prior model [44][46][56][57]. The segmentation is then obtained by maximizing the a posteriori probability of the segmentation given the image data using iterative methods. A difficulty associated with MRF models is proper selection of the parameters controlling the strength of spatial interactions [50]. Too high a setting can result in an excessively smooth segmentation and a loss of important structural details. In addition, MRF methods usually require computationally intensive algorithms. Despite these disadvantages, MRFs are widely used not only to model segmentation classes, but also to model intensity inhomogeneities that can occur in MR images [46] and texture properties [60]. Figure 2-3 gives an example on ventricle segmentation in MR image using a MRF algorithm proposed in [46].

Figure 2-3 Segmentation of the MR image. (a) pd and (b) T_2 weighted transversal slices of a 3Dbrain image. (c) One iteration and (d) six iterations of proposed algorithm in [46]. (e) Final result. (Adopted from [46].)

2.2.5 Artificial Neural Networks

The Artificial neural networks (ANNs) are massively parallel networks of processing elements or nodes that simulate biological learning. Each node in an ANN is capable of performing elementary computations. Learning is achieved through the adaptation of weights assigned to the connections between nodes.

The ANNs represent a paradigm for machine learning and can be used in a variety of ways for image segmentation. The most widely applied use in medical imaging is as a classifier [43], where the weights are determined using training data, and the ANN is then used to segment new data. The ANNs can also be used in an unsupervised fashion

as a clustering method [33][59], as well as for deformable models [64]. An example of unsupervised segmentation using ANN [59] is shown in Figure 2-4.

Figure 2-4 Unsupervised segmentation using ANN. The input images to unsupervised segmentation neural network are (a) T1-weighted , (b) T2-weighted and (c) PD-weighted. Only the image data within the automatically defined octagonal region of interest was used to train the segmentation network. (d) The segmentation result. (Adopted from [59].)

Because of the many interconnections used in a neural network, spatial information can easily be incorporated into its classification procedures. Although the ANNs are

inherently parallel, their processing is usually simulated on a standard serial computer, thus reducing this potential computational advantage.

2.2.6 Deformable Models

Deformable models are physically motivated, model-based techniques for delineating region boundaries using closed parametric curves or surfaces that deform under the influence of internal and external forces. To delineate an object boundary in an image, a closed curve or surface must first be placed near the desired boundary and then allowed to undergo an iterative relaxation process. Internal forces are computed from within the curve or surface to keep it smooth throughout the deformation. External forces are usually derived from the image to drive the curve or surface towards the desired feature of interest.

The main advantages of deformable models are their ability to directly generate closed parametric curves or surfaces from images and their incorporation of a smoothness constraint that provides robustness to noise and spurious edges. A disadvantage is that they require manual interaction to place an initial model and choose appropriate parameters. Reducing sensitivity to initialization has been a topic of research that has demonstrated excellent success [40][52][131]. The traditional deformable models can also exhibit poor convergence to concave boundaries. This difficulty can be alleviated somewhat through the use of pressure forces and other modified external force models [131]. Another important extension of deformable models is the adaptivity of model topology using an implicit representation rather than an explicit parameterization, such as the G-snake [40], T-snake [89]-[91] and level set approach [52][53]. The details introduction on deformable models are presented in the section 2.3.

2.2.7 Atlas-Guided Approaches

Atlas-guided approaches are a powerful tool for medical image segmentation when a standard atlas or template is available. The atlas is generated by compiling information on the anatomy that requires segmenting. This atlas is then used as a reference frame for segmenting new images. Conceptually, atlas-guided approaches are similar to classifiers

except they are implemented in the spatial domain of the image rather than in a feature space.

The standard atlas-guided approach treats segmentation as a registration problem [51]. It first finds a one-to-one transformation that maps a pre-segmented atlas image to the target image that requires segmenting. This process is often referred to as atlas warping. The warping can be performed using linear transformations [49][58] but because of anatomical variability, a sequential application of linear and non-linear transformations [41][61] is often used, Figure 2-5 shows the detected sulci by the algorithm proposed in [61]. Because the atlas is already segmented, all structural information is transferred to the target image.

Figure 2-5 Sulci detection using Atlas-guided approach. (a) Transverse slice through extracted binary brain image. (b) The extracted cortical surface. (c) The sulci detected by the morphological algorithm proposed in [61]. (Adopted from [61].)

The atlas-guided approaches have been applied mainly in MR brain imaging. An advantage of atlas-guided approaches is that labels are transferred as well as the segmentation. They also provide a standard system for studying morphometric properties [48][63]. Even with non-linear registration methods however, accurate segmentations of complex structures is difficult due to anatomical variability. Thus, atlas-guided approaches are generally suitable for segmentation of structures that are stable over the population of study.

2.2.8 Other Approaches

The model-fitting is a segmentation method that typically fits a simple geometric shape such as an ellipse or parabola to the locations of extracted image features in an image [54]. It is a technique which is specialized to the structure being segmented but is easily implemented and can provide good results when the model is appropriate. A more general approach is to fit spline curves or surfaces [39] to the features. The main difficulty with model-fitting is that image features must first be extracted before the fitting can take place.

The watershed algorithm uses concepts from mathematical morphology [135] to partition images into homogeneous regions. This method can suffer from over segmentation, which occurs when the image is segmented into an unnecessarily large number of regions. Thus, watershed algorithms in medical imaging are usually followed by a post-processing step to merge separate regions that belong to the same structure [62].

2.3 Deformable Models for 2D Object Segmentation

Among the methods mentioned in section 2.2, the deformable model which is popularly known as snake [2] has attracted the most attention to date. It is originally introduced by M. Kass [2], which is described as an energy minimizing curve guided by external and internal forces. The external force is given as user-supplied input function. The internal forces are determined solely by the shape of the curve and are usually defined in terms of first and second order derivatives. The snake gives an elegant method to simulate an elastic material, which can dynamically conform to local image features.

The snake has two main features:

- (1) A large number of data points that defines the curve;
- (2) An explicit formulation of smoothness constraint.

The following problems have been recognized in the literature [109][144]:

- (1) Slow convergence speed because of the large number of coefficients to be optimized;

- (2) Difficulty in determining the weights associated with the smoothness constraints;
- (3) The curve is represented by a finite set of disconnected points;
- (4) High-order derivatives on the discrete curve may not be accurate in noisy environment [79].
- (5) Snakes are very sensitive to the initial position and to the image because the specific shape information about the object is not encoded in the model.

Based on the original construction of the snake, different solutions have been suggested to render the snake more stable and to yield faster convergence results [80]-[83]. The statistical information has been implemented to achieve a good segmentation of object being studied. An alternative approach was developed by using a parametric B-spline representation of the curve. Such a formulation of a deformable model allows for the local control and a compact representation. Moreover, this formulation has a fewer number of parameters to control and the smoothness requirement has been implicitly built into the model.

The deformable models are able to achieve the best results, as they possess the flexibility to approach the object boundaries. Below, we introduce the point-based deformable model, and then the B-spline based deformable model.

2.3.1 Point-Based Deformable Model

The ASM, introduced by Cootes *et al.* [143], is a statistically based technique for building deformable shape templates and segmenting various organs from 2D medical images. In this model, the shape models represent objects by sets of fixed number of landmark points which have to be manually placed in the same way on an object boundary in each input image. The points can be connected to form a deformable contour. For different shapes in a training set, after compensating for the differences by the object posed by affine transformation, the Point Distribution Model (PDM) is constructed to model the landmark points' distribution of these shapes. A local structure model for each landmark point is constructed as well. The statistical parameterization provides global shape constraints and allows the model to deform only in ways implied

by the training set. For example, to extract the left ventricle from echocardiograms, one may choose points around the ventricle boundary, the nearby edge of the right ventricle, and the top of the left atrium. By examining the statistics of training sets of hand-labeled medical images and using principal component analysis, a shape model is derived that describes the average positions and the major modes of variation of the object points. New shapes are generated using the mean shape and a weighted sum of the major modes of variation. The ASM segmentation method involves finding the new landmark points that best match the local structure model. A new shape is generated that is similar to the shapes in the training set constructed from the PDM. This process is repeated until convergence. Figure 2-6 demonstrates using the ASM of the cartilage to locate the structure in a new image. The ASMs have been used for several segmentation tasks in medical images [145]-[150]. As the landmark points in the ASM have to be assigned manually, in the case where there are many samples in the training set, extremely heavy workload would result.

Figure 2-6 Search the cartilage using Active Shape Model on an MR image of the knee. (Adopted from [144].)

In [87], a PDM approach is developed for boundary finding where the correspondence of a subset of boundary points to a model is simultaneously determined. Global shape parameters derived from the statistical variation of object boundary points in a training set are used to model the object. A Bayesian framework is employed to find the object boundary with its subset points in correspondence with boundaries in the training set or the mean boundary. Figure 2-7 shows an example on endocardium detection in MR image using this approach. In this algorithm, the landmark points can move to anywhere in the image with the neighboring points moving together. Shapes are not restricted by a training set and thus the model is not specific but allows variability. Also, the critical points no longer correspond. No texture information has been used in this algorithm and it cannot handle the case of affine transform.

Figure 2-7 An example on endocardium detection in MR heart image. Left: original MR image; Middle: initial contour; Right: final contour. (Adopted from [87].)

Nicolae Duta and Milan Sonka present a method to improve ASM by constructing a knowledge-based PDM in [15]. Three additional features characteristic to MR brain images were included in the model: Gray-level appearance, border strength and mean position of landmark points. The searching procedure developed for this PDM approach to image segmentation and interpretation is based on the model fitting strategy. Their model fitting procedure substantially differs from the ASM of Cootes and Taylor [143]. The difference is twofold. First, their 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 (no image preprocessing is done). 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 2-8.

Figure 2-8 A right thalamus segmentation. (a) Optimal model position after piecewise linear transform. (b) Outlier detection—the two detected outliers marked by not connected white squares. (c) Final point adjustment. (d) Final outlier removal. (Adopted from [15].)

Cootes *et al.* present a method of matching statistical models of appearance to images[16][144], called the Active Appearance Model (AAM). A set of model parameters control the modes of shape and gray-level variation learned from a training set. To build this model, it requires a training set of annotated images where

corresponding points have been marked on each example (see Figure 2-9). Procrustes analysis is applied to align the sets of landmark points (each represented as a vector, x) 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 which includes the object intensity distribution but not the object contour. 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 Cootes [16] suggested using a gradient descent method that relates model coefficients with the difference between a synthesized model image and the target image. Figure 2-10 shows the procedures of face AAM search for each face, each starting with the mean model displaced from the true face center.

Figure 2-9 A labeled training image gives a shape-free patch and a set of points in the AAM. (Adopted from [144].)

Compared to the ASM search, the AAM search is slower. The procedure tends to be more robust than the ASM search alone, since more of the image evidence are used. The AAM is very suitable for approaching object with constant contents such as human face, because the organs and their locations on every person's face are relatively constant. But for those objects with various appearance contents such as ventricles of MR images shown in Figure 2-11, the AAM will fail due to significant difference of structure within the ventricles.

Figure 2-10 The face AAM approaches to each face. (Adopted from [144].)

Figure 2-11 Two examples of ventricle in the MR images where the AAM fails.

An ASM segmentation scheme is presented in [18] that is steered by optimal local features, contrary to normalized first order derivative profiles as in the original formulation of the ASM. A nonlinear k NN-classifier is used, instead of the linear Mahalanobis distance, to find optimal displacements for landmarks. Training data were selected from 5×5 neighborhoods around each landmark point. A Taylor expansion is used in this scheme to approximate a function f around the landmark point by a

polynomial of (some) order N , and use a filter bank of multiscale Gaussian derivatives to describe local image structure. After that it extracts features for each location by taking the first few moments of the local distribution of image intensities (the histogram) around each location. For each of the landmarks that describe the shape, at each resolution level taken into account during the segmentation optimization procedure, a distinct set of optimal features is determined. The selection of features is automatic, using the training images and sequential forward and backward feature selection. Examples of lung segmentation using this model are shown in Figure 2-12.

Figure 2-12 The left and right field of lung segmentation using the ASM with optimal features. (Adopted from [18].)

A method for automatically building statistical shape models from a training set of example boundaries or surfaces is presented in [17]. In order to establish a set of dense correspondence landmarks between all members of a set of training shapes instead of manual operation, an algorithm which minimizes the description length of the training set is proposed. The correspondence problem is converted to one of finding the parameterization for each shape in the training set that builds the “best” model. The “best” is defined as that which minimizes the description length of the training set. However, as the authors mentioned in this paper, the proposed algorithm requires a large number of function evaluations, each result takes several hours to produce. Therefore, it becomes impractical when a larger set of training shapes is used because of its complexity.

By using the wavelet transform, Chuang *et al.* developed a hierarchical planar curve descriptor [88] that decomposes a curve into components of different scales, so that the coarsest scale components carry the global approximation information while the finer

scale components contain the local detailed information. It has a multiscale analysis capability. The application of the wavelet descriptor to model-based contour extraction from computed tomography (CT) image is showed in Figure 2-13.

Figure 2-13 Contour extraction using wavelet deformable model in CT image. (a) Initialization. (b)-(c) Deformation of model. (d) Extraction result. (Adopted from [88].)

The snakes cannot adapt to complex shapes of object boundaries because the number of data points remains constant during the evolution. Although the snake is simple to implement, the accuracy of the extracted boundary is limited because the snake is handicapped by the fixed number of data points. A deformation scheme that extends the geometrical flexibility of the snake to deal with complex shapes, referred to as topologically adaptive snakes (T-snakes), is revealed in [89]-[91]. An Affine Cell Decomposition (ACID) framework has been embedded in T-snake. The ACID framework provides an elegant mechanism for contour reparameterization, allowing T-snakes to segment and reconstruct objects with significant protrusions, tubular objects, or objects with bifurcations. The shape of the contour is dictated by the energy grid also provides a mathematically sound basis for robust functional topological transformations. This property enables T-snakes to seamlessly split or merge, adapting to the topology of a target object. The ACID reparameterization mechanism makes the model relatively insensitive to initial placement, significantly improving the efficiency and automation of the segmentation process. Figure 2-14 shows an example on segmentation in MR brain image using T-snake. However, both traditional snake and T-snake suffer from the drawback that the movement of a single data point affects the entire length of the contour; a small change in the position of a data point tends to strongly propagate throughout the entire snake. This can be described as a global propagation of change of a particular data point.

The watershed algorithm from mathematical morphology takes a very different approach, compared to the energy-based methods described. The watershed algorithm has been proven powerful for contour detection [96][97] as well as image segmentation [92][94]. In the watershed algorithm, the input is a relief function representing edge evidence, where the morphological gradient is a common choice for computing such a relief. By viewing this function as a mountain landscape, object boundaries are determined as watershed lines. In [92][93], the watershed algorithm is implemented via region growing, where seeds are the regional minima of the relief. To prevent over segmentation due to a possibly large number of minima in the original edge evidence function, the watershed algorithm from selected markers is usually employed [92][94]. This technique selects markers from significant minima and modifies the relief accordingly such that nonsignificant minima are filled up. The markers are usually extracted as low gradient zones in the image, smoothed by a morphological filter such as opening by reconstruction [95]. Compared to the other methods mentioned, the watershed has several advantages, including the proper handling of gaps and the placement of boundaries at the most significant edges. It is difficult, however, to impose a priori information, in particular smoothness, on the watershed lines. As the algorithm relies purely on the edge evidence, the boundaries between regions may be noisy. A simple post processing smoothing of the watershed line does not take into account the observed information and, therefore, important corners with high edge evidence may be lost.

Figure 2-14 Segmentation with T-snake in an MR brain image. (a) Initialization of T-snake. (b) – (d) T-snake segmenting gray-matter / white-matter interface and ventricles. (Adopted from [91]).

The watersnake is first introduced by Nguyen *et al.* [98][99], as a new segmentation method which is implemented by a multiscale algorithm. It combines the watershed segmentation from mathematical morphology and energy-based segmentation methods. In the watersnake, the distance-based approach is used for the definition of the watershed line. An energy function that is equivalent to the watershed segmentation has been derived. Therefore, a priori knowledge on boundary smoothness or shape can be taken into account. In particular, the smooth watersheds are obtained by adding the length of the region boundary to the energy function. An example of brain segmentation using watersnake is shown in Figure 2-15.

Figure 2-15 Brain segmentation. (a) CT brain images. (b) The relief computed from morphological gradient. (c) The markers extracted. (d) The result of the watersnake. (Adopted from [98]).

Li presented a method in [100] which is model based and constrained by the model obtained from the training set. It can automatically locate the optic disk and estimate its shape in color retinal images. The shape of optic disk is obtained by an active shape method in which affine transformation is used to transform the shape model from shape space to image space. The PCA is applied to the candidate regions at various scales to locate the optic disk. The minimum distance between the original retinal image and its projection onto “disk spaces” indicates the center of the optic disk. However, the model-based algorithm still requires that the optic disk should be distinguishable from any lesions on the retina, and have most of the boundary identifiable. Two examples of optic disk shape detection using this algorithm are showed in Figure 2-16.

Figure 2-16 Two examples of optic disk shape detection: (a) from a retinal image of the left eye and (c) from the right eye. (a) and (c) Initialization of shape model in the image space and (b) and (d) the corresponding results of shape detection. (Adopted from [100]).

2.3.2 B-Spline Based Deformable Models

In [84], Feng *et al.* presented a method to find a cubic spline model to optimally approximate a given planar shape. This problem is solved by treating the control points which defined the spline as variables and apply an optimization technique to minimize an error norm so as to find the best locations of the control points. The error norm, which is defined as the total squared distance of the curve sample points from the spline model, reflects the discrepancy between the spline and the original curve. An extension of this method is also mentioned, which allows the addition of the control points to the spline model based on local error measures to represent the given shape within a pre-specified tolerance. However, the detail of control point insertion algorithm has not been provided.

A deformable B-spline algorithm was presented in [101] for determining vessel boundaries and enhancing their centerline features. A bank of even and odd Stretched-Gabor filter [102] 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 the cubic B-snake and are optimized on filter outputs with the Dynamic Programming: Beginning with the first segmentation of the B-spline, the locations of all four control points are varied systematically within a limited window and the energy corresponding to each conformation is calculated. These energies are minimized over the first control point. The procedure is repeated for the next segmentation, the conformation of which is controlled by the last three control points of the previous segmentation and the next control point in the sequence. Energies corresponding to all of the possible locations of these four control points are calculated from the spline energy plus the minimum energies from the first segmentation. This scheme continues until all the control points have been varied and the minima of all the remaining conformations can be sought. Stepping back through stored optimal control point locations reveals the lowest energy state conformations. This iterative process is repeated until the overall spline energy reaches a minimum, i.e., there is no lower energy state than the one occupied by the snake. Figure 2-17 shows an example. There are few drawbacks in this algorithm: (1)

the locations of starting and end points of the B-spline curve have to be fixed; (2) Dynamic Programming is very slow for the B-snake deformation; (3) the number of control points for the B-snake has to be fixed; (4) no statistic information is being used.

Figure 2-17 An open B-snake model for vessel boundaries approaching. (a) initialized; (b) after five iterations of the Dynamic Programming algorithm; (c) after ten iteration; (d) after ten iteration; (e) after 20 iteration; (d) optimum reached at 23 iterations, line connect initial control point locations and optimized locations. (Adopted from [101].)

Cham and Cipolla presented an automatic scheme for spline-fitting [105], the proposed method incorporates cubic B-spline active contours, the Minimum Description Length (MDL) principle and a control point insertion strategy based on maximizing the Potential for Energy-Reduction Maximization (PERM). However, the maximum of PERM has to be found by exhaustive search along the entire B-spline contour after the active contour optimization has converged. Hence, this method is slow. Figure 2-18 illustrates the PERM strategy while Figure 2-19 shows the spline fitting of PERM strategy.

Figure 2-18 Figure illustrating the PERM strategy. (Adopted from [105].)

Figure 2-19 Results of PERM strategy. (Adopted from [105].)

Wang *et al.* presented a B-snake based lane model for lane detection in [125]-[127]. In this model, a new B-snake lane model has been 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 providing a good initial position for the B-snake lane model. A Minimum Mean Square Error (MMSE) method has been formulated with dual external forces to update the control points of B-snake for approaching the real road boundaries. However, although the results are good in lane detection, the number of control points is fixed to three in this model. The results for lane detection are shown in Figure 2-20.

Figure 2-20 Lane detection using B-snake model. (Adopted from [127].)

Mário presented an approach method to adaptive estimation of parametric deformable contours based on the B-spline for unsupervised contour representation [106]-[107]. The problem is formulated in a statistical framework with the likelihood function derived from a region-based image model. The parameters of the image model, the contour parameters and the B-spline parameterization order (i.e., the number of control points) are all considered unknown. The parameterization order is estimated via a Minimum Description Length (MDL) type criterion. A deterministic iterative algorithm is developed to implement the derived contour estimation criterion. However, the method presented in this paper has some shortcomings: (1) To determine the number of control

points for B-spline is by exhaustive search over a range of values (Figure 2-21) and searching the minimum of the MDL (Figure 2-22). As we know, it is very difficult to define a range of number of control points for an uncertain object shape. Also, it is time consuming to perform such exhaustive search; (2) The initial B-spline model has to be put inside the region where the object is located. This requires extra manual involvement; (3) Every pixel in the image must be assigned an indicator to indicate whether it belongs to the inside or outside region of the contour. The likelihood function involves this indicator. This requires the prior knowledge of the object region characters and more processing time; (4) The proposed method can only handle images where the region inside the object being studied is homogenous. This condition may not be satisfied in some objects, particularly in medical image analysis.

Figure 2-21 Evolution of the contour shape for 5, 8, 10 and 11 control points. The dashed line in (a) is the initial contour. The control points are not showed in figure. (Adopted from [107].)

Figure 2-22 The description length corresponding to the example in Figure 2-21. (Adopted from [107].)

Patrick Brigger presented an alternative formulation for the B-snakes that can be used as a tool for fast and intuitive contour outlining [109], in which the intrinsic scale of the spline model adjusted a priori. This leads to a reduction of the number of parameters to be optimized and eliminates the need for internal energies (i.e., the regularization term). In other words, it is to control the elasticity of the spline implicitly and rather intuitively by varying the spacing between the spline knots. The theory is embedded into a multi-resolution formation demonstrating improved stability in noisy image environments (Figure 2-23). Validation results are presented showing the similar performance of the traditional snake using energies and the proposed approach without internal energies. The number of control points is fixed in this model. The proposed method was applied to segmentation of a PET image sequence as well. The results are shown in Figure 2-24.

Figure 2-23 Multi-resolution approach vs. single-resolution approach. (Adopted from [109].)

Figure 2-24 PET-scan of the brain. (Adopted from [109].)

In order to increase the local flexibility of contour, a Non-Uniform Rational B-spline (NURBS) snake is proposed in [110]. This model is attached with an additional weighting parameter that can be adjusted to control the flexibility of the curve at each control point. During the evolution, the weight associated with each distinct control point is modified depending on the curvature of the points on the relevant spline segment. As shown in Figure 2-25, such a modification of weights influences the snake to attract towards high curvature regions without adding more control points.

Figure 2-25 Movement of a curve point $v(s)$ towards the desired control point when the associated weight is increased. (Adopted from [110].)

In this method, a control point insertion strategy has been suggested as well. It used both the Euclidean distance and the curvature of NURBS snake to increase (or decrease) the number of control points if both Euclidean Distance and curvature exceed the relevant thresholding limits. However, checking of the Euclidean distance and the curvature is performed on the current model curve only, but not the object contour in the image. Therefore, the number of control points decided might not be best fitted to the real object contour. Moreover, this model is using balloon force to push or pull the snake model to object boundaries subject to the condition that the initial snake model is inside or outside of the object. This needs extra manual involvement, and in case the initialized snake model is partly inside the object and the rest is outside the object, the balloon force

method faces problems in being defined. Furthermore, no prior knowledge has been used in this model to guide the snake deformation to avoid local minima.

Tang and Acton present a B-snake model with Multiscale Gradient Vector Flow as external force in [111] for vessel boundaries and detection and tracking. The number of control points for this B-snake model is fixed to 8 and the approaching estimation for control points is using least squares estimation. Actually, the curve representation of this model is similar to the B-snake lane model presented by Wang *et al.* in [125]-[127], except that no parallel mechanism has been used to describe the relationship of both side boundaries of vessel to enhance the robustness of vessel detection. Some results of vessel boundary detection are shown in Figure 2-26.

Figure 2-26 Vessel boundary detection results obtained with the proposed algorithm [111] on the trauma sequence. (a) Final active contour at the first frame. (b) Final active contour at the 81st frame. (c) Final active contour at the 181st frame. (d) Final active contour at the 241st frame. (Adopted from [111].)

Based on our studies of deformable models, from the curve description point of view, some recently proposed active contours, such as ASM, T-snake, watersnake and wavelet approach, are all belonged to the point-based snake models, but the deformation mechanism are various. We found that, the similarities between the point-based snake and B-snake are:

- (1) A curve descriptor has to be used in these snake models for object shape description, it is based on either a set of points or a spline curve.
- (2) A deformation mechanism is applied in the curve deformation, although it is various in the different models.
- (3) There is an energy term embedded in the minimum functions for deforming the model, most of them are involved the edge information.

However, the main differences between the point-based snake and B-snake are:

- (1) The point-based snake models use much more parameters compared to B-snake model. In the point-based snake model, the curve is presented directly by a set of points. For the cubic B-snake, the B-spline curve is consisted of few segments, each segment is represented by four parameters (control points).
- (2) Internal forces are required in point-based snake model to maintain the smoothness, but not required in B-snake model, since the representation of B-spline has guaranteed smoothness by implicit constraints.
- (3) The B-spline curve is shape invariance under affine transformation, it is a very attractive curve model to adapt to the case that the affine transformation is assumed between two shapes.

Although B-snake model possesses some very attractive properties such as spatial uniqueness, boundedness and continuity, and local shape controllability, there are few drawbacks that have been encountered. They are:

- (1) Most of B-snake models do not have the flexibility to describe an unknown complex contour as they are using a pre-defined fixed number of control points.

- (2) Very few B-snake models have the capability to describe a complex contour with a different number of control points although the contour is achieved by exhaustive search over a number of control points. Therefore, such methods are very inefficient.
- (3) No statistical information about the object being studied is implemented in B-snake model to guide the deformation process.

Therefore, B-snake can be improved by solving the above-mentioned problems.

2.4 Object Curve Matching

Object matching means establishing correspondences between object features in different images. Since an object may undergo various affine transformations including rotation, scaling and translation, object matching may also mean computing any transformation parameters based on 2D or 3D object representations.

If an object's shape is simple, a direct approach is to match object boundary curves extracted from images, where the objects may undergo various affine transformations and small deformations. One method to compare two object curves is to match the corresponding points on two curves assuming affine objects. However, the method is both sensitive to noise because of imperfect data and imprecise correspondences. Instead of point matching, a variety of less noise-sensitive approaches have been proposed based on comparing corresponding line segments and polygonal arcs.

Basically, the existing methods for shape matching can be classified into two categories: (a) feature-based matching, and (b) intensity-based matching.

(a) Feature-Based Matching

Feature-based matching uses the spatial arrangements of extracted features such as edge elements or junctions. Silhouettes have been described (and compared) using Fourier descriptors [65], skeletons derived using medial axis transform [66], or directly matched using dynamic programming e.g. [67]. Since silhouettes are limited as shape descriptors for general objects, other approaches [68][69] treat the shape as a set of points in the 2D

image using an edge detector. Key points or landmark points are used in [70] to recognize the objects using the spatial arrangements of point sets. In [71], the shape context for each point is applied for the estimation of shape similarity and correspondences.

(b) Intensity-Based Matching

Intensity-based matching involves the use of the pixel intensity value, instead of focusing on the shape of the occluding contour or other extracted features. Several approaches [72][73][16] first attempt to find correspondences between the two images, before doing the comparison. This turns out to be quite a challenge, as differential optical flow techniques do not cope well with the large distortions that must be handled due to pose/illumination variations. Errors in finding correspondence will cause downstream processing errors in the recognition stage. As an alternative, there are a number of methods that build classifiers without explicitly finding correspondences. In such approaches, one relies on a learning algorithm having enough examples to acquire the appropriate invariance.

2.4.1 Curve Matching Using B-Spline Model

Up to the present, very little work has been done on the use of the B-spline curve for matching or recognition purposes, partly because of the non-uniqueness of the B-spline parameters of the curves [171]-[172]. This makes the comparison between two curves based on the control points a nontrivial task.

A 3D B-spline curve model (note that it is not a Surface but a curve in 3D space) is proposed in [171] for 3D object curve recognition and shape estimation. A minimum mean square error estimation technique is presented for B-spline control point estimation. The recognition is based on the Fourier descriptors extracted from the normalized control points. As the curve matching is achieved by using a similarity measure based on the B-spline knot points, this method can neither handle the affine transformation between the curves nor the occlusion.

Two different solutions are proposed in [172] for curve matching using B-spline curve representation. The first method is to re-construct a new B-spline with a fixed number of knot points (control points) and a control-point re-assignment progress, and then find the corresponding knot point of sample and different prototypes. The drawback of this method is that the number of control points to re-construct a new B-spline curve and the start point of new B-spline may be very hard to determine for accurately representing the original curve. These problems would affect the final result. The second method is a residual error-based method that compares the sum of the residual error between the sample and the different prototype curves. The parameters of affine transform can be estimated by sampling the same number of points in both sample and prototype curves, and then using a minimum mean square error method. Finally, the curve matching is achieved by minimizing the sum of the residual errors of the sample to different prototypes. The problem for this method is that the authors did not give out any criteria on selecting the affine variant for sampling the points on the B-spline curve for calculation. If those sampled points have a lack of good correspondences and do not possess affine invariant property, obviously, the estimated affine transformation based on these sample points and the prototype is not reliable. Moreover, the calculation of residual error between sample and prototype has to be done for every sampled point and thus, not suitable for cases where there are many prototypes that need to be matched.

An algorithm is presented in [173] for estimating both closed and open B-spline control points by using a class of weighted B-spline curve moments. It is an extension of the research work of [171]. One of the proposed weighted B-spline curve moments is using an affine invariant kernel which is exactly same as the affine length. However, in this method, in order to find the corresponding points between prototype curve and the affine-transformed curve, the curves are broken into a certain number of segments, each having the same arc length, and then a coarse exhaustive search for a smallest error measure through all possible combinations of those segments is performed. As the arc length is not affine-invariant, this method may not get a precise result.

A cubic B-spline curve model is presented in [104] for registering the rat's brain external contour by a Bayesian decision approach. In this model, the rat brain external contour was represented by a cubic B-spline curve. The registering algorithm is based on a set of

local absolute affine invariants derived from the set of ordered inflection points on the external contour. These invariants are constructed from the sequence of area patches bounded by the contour and the line connecting two consecutive inflection points. See the demonstration in Figure 2-27. The number of control points is fixed by a ratio (1:10) of data points. Obviously, a fixed ratio is not flexible for a smooth or complex shape. In addition, to determine that ratio is another problem. Moreover, as the local absolute affine invariants are directly derived from the set of inflection points, the noise effects or having few inflection points on the contours under study would degrade this proposed algorithm.

Figure 2-27 The inflection points on the B-spline curve. $|A(i)|$ indicates the area of patch i . (Adopted from [104].)

In paper [108], Gu and Tjahjadi presented a hybrid algorithm for coarse-to-fine-matching of affine-invariant B-spline object curves and simultaneous estimation of transformation parameters. The knot points of B-spline object curve are chosen using a constrain B-spline curve model under minimum mean square error criterion. For coarse-matching, two dissimilar measures are exploited by using the significant corners of object boundaries and the sharpest corner as the start point to remove candidate objects with large dissimilarity to a target object. For fine-matching, a point interpolation approach and a simple gradient-based algorithm are applied to B-spline object curves under minimum mean square error criterion. In this algorithm, the arc-length is assumed to be

affine-invariant, it has been proved that the assumption is wrong [177]. The significant corners (shown in Figure 2-28) are extracted to form the affine-invariant object features in this algorithm. Unfortunately, the significant corners or high curvature points are not affine-invariant [139] and may result in getting the wrong parameters for affine transformation. Since the fine-matching step uses the B-spline approximated curves, this algorithm requires that objects must be rigid or only with small deformations, this would reduce the flexibility in the practical application. The affine transformations considered in this algorithm are limited to only rotations, scalings and translations, but not shearing. Therefore, this method is invariant to similarity transformations.

Figure 2-28 Boundaries of objects with marked corners. Mark “*” denotes the Significant corner, while “o” denotes the sharpest corner. (Adopted from [108].)

In this section, we have presented a review on the curve matching using B-spline model. It includes mainly journal papers in the literature. We know that, up to present, very little research work has been explored in the B-spline curve matching. One possible reason might be due to the fact that the B-spline curve is not uniquely described by a single set of parameters (control points) which made the curve matching process difficult when comparing the respective parameters of the curves to be matched. The encountered problems in current existing methods can be summarized as follows:

- (1) Research work so far tends not to employ the advantage of continuous curve description of B-spline. Discrete points which are sampled from the B-spline curve have been used to represent the curve in matching processing. In performing curve matching this way, the B-spline curve is not different from the point-based curve model.
- (2) These algorithms are not affine invariant. In order to estimate the minimum error, some methods are using either the corner points or the discrete points re-sampled from B-spline curve with a certain arc-length. As these points are not affine invariant, these methods may fail to handle the affine transformation.
- (3) These approaches are sensitive to noise. Error minimum estimations for curve matching are directly based on the points of curve, rather than the entire curve. Therefore, they are very sensitive to noise and would introduce sampling error.

Actually, the problems mentioned above are mainly caused by the non-uniqueness property of B-spline curve. Hence, once this obstacle is removed, the B-spline will show its advantage in curve matching.

2.5 Concluding Remarks

In this chapter, reviews of object segmentation using the B-spline deformable model and B-spline curve matching have been presented.

In review of the summary in section 2.3.2, a new B-snake model should be developed for solving the problems mentioned in this section. In the next few chapters, a new

knowledge-based B-snake model will be presented. It possesses a strategy of automatic control-point insertion that is adaptive to the structure of the studied object. A method of Minimum Mean Square Error (MMSE) to iteratively estimate the position of these control points in the B-snake model, and a statistical model for geometrical and appearance modeling to guide the deformation overcome the local minima.

For object matching using the B-spline model, as mentioned in section 2.4.1, it remains a difficult task due to the problem of the non-uniqueness of the control points in the B-spline representation. In Chapter 7, this problem will be addressed and a novel B-spline curve matching algorithm will be presented. It employs the advantages of the B-spline that are continuous curve representation and affine invariant, and the robustness of the CSS matching with respect to noise and affine transformation. In addition, it avoids the conventional CSS matching that has to use re-sampled points on the curve, thus reducing the error.

In the next chapter, an overview on the B-spline and B-snake model is given. Some properties of B-spline curve are introduced as well.

Chapter 3

B-Spline and B-Snake Models –

An Overview

3.1 Introduction

A number of approaches have been proposed for curve modeling such as Fourier descriptor [163], chain codes [164], polygonal approximation [165], curvature primal sketch [166], medial axis transform [167], parametric algebraic curve [168], shape axis tree [169], shape space [170] and B-spline [125]-[130], [171]-[173], among others.

The B-splines stand as one of the most efficient curve representation and possess very attractive properties such as compactness and continuity, local shape controllability, and invariance to affine transformations. These properties made it very attractive for curve representation and consequently, it has been extensively used in computer-aided design and computer graphics. Moreover, B-spline possesses the advantages over other curve representations in the area of data compression, storage and transmission.

In this chapter, the formulation and major properties of the uniform cubic B-spline are introduced. The reasons for using it as an appropriate model for curve representation are then discussed. Finally, the traditional snake and B-snake are introduced.

3.2 On B-Spline Model

A more economical realization for curve representation can be obtained using far fewer state variables by B-splines [123][152]. The “B” stands for “Bezier”. These are deformable curves represented by four or more state variables (control points). The curves may be open or closed as required. The flexibility of the curve increases as more control points are added; each additional control point allows either one more inflection in the curve or, when multiple knots are used [118], reduced continuity at one point.

The B-splines are piecewise polynomial functions that provide local approximations to contours using a small number of parameters (control points). They have been extensively used in the representations of curves and surfaces [119] in computer-aided designs [120] and computer graphics [121]. The theory for B-splines was first suggested by Schoenberg [122].

The 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. The cubic B-splines exhibit a favorable tradeoff between their shape complexity for describing natural curves and computational burden required to solve them. Therefore, we are concerned with the implementation of cubic B-spline. The uniform cubic B-spline is a simple and particularly useful class of B-spline, the term “uniform” means that the knots are equally spaced. In the following sections, the structure of the cubic B-spline and its properties are introduced.

3.2.1 Uniform Cubic B-Splines

A k -th order uniform B-spline is C^{k-1} continuous. As we deal with the curves that are C^2 continuous, i.e., that have both their slopes and curvatures continuous, the uniform cubic B-splines are the mainly focus in this thesis.

(A) Curve Formula of Uniform Cubic B-Spline

For a given set of $n+1$ control points, $\{Q_0, Q_1, \dots, Q_n\}$, the uniform cubic B-splines can be either the open or closed curves. The open uniform cubic B-spline curve (Figure 3-1) consists of $(n-2)$ connected curve segments $g_i(s) = (x_i(s), y_i(s))$, $i = 1, 2, \dots, (n-2)$, each of which is a linear combination of four cubic polynomials in the parameter s , where s is normalized for each such segment between 0 and 1 ($0 \leq s \leq 1$).

Figure 3-1 An open cubic B-spline curve.

The i -th segment of a cubic B-spline is given by [152]:

$$g_i(s) = X_0(s)Q_{i-1} + X_1(s)Q_i + X_2(s)Q_{i+1} + X_3(s)Q_{i+2}, \quad i = 1, 2, \dots, n-2 \quad \dots(3.1)$$

where the functions $X_k(s)$ are nonnegative with amplitude in the range of $(0,1)$, and $X_0(s)$, $X_1(s)$, $X_2(s)$ and $X_3(s)$ are defined as follows:

$$\begin{aligned} X_0(s) &= \frac{1}{6}(1-s)^3 \\ X_1(s) &= \frac{1}{2}s^3 - s^2 + \frac{2}{3} \\ X_2(s) &= -\frac{1}{2}s^3 + \frac{1}{2}s^2 + \frac{1}{2}s + \frac{1}{6} \\ X_3(s) &= \frac{1}{6}s^3 \end{aligned} \quad \dots(3.2)$$

Functions $X_k(s)$ make the slopes and curvature vectors continuous at the connection point between curve segments (see Figure 3-2). Thus the curve segment formula (3.3) can be expressed as:

$$g_i(s) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \quad \dots(3.3)$$

Thus

$$g_i(s) = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix}, \quad i = 1, 2, \dots, n-2 \quad \dots(3.4)$$

where:

$$M_R(s) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \quad \dots(3.5)$$

In Figure 3-1, the connection points between curve segments are called the knot points $P_i (i = 1, 2, \dots, n-1)$, where the B-spline bases are tied together, $P_i = g_i(0), i = 1, 2, \dots, n-1$.

Figure 3-2 Uniform cubic B-spline basic functions.

For the case of the closed B-spline curve, given a set of $n+1$ control points, $\{Q_0, Q_1, \dots, Q_n\}$, the closed uniform cubic B-spline curve (Figure 3-3) consists of $(n+1)$ connected curve segments $g_i(s) = (x_i(s), y_i(s))$, $i = 1, 2, \dots, (n+1)$:

$$g_i(s) = M_R(s) \begin{bmatrix} Q_{(i-1) \text{ Mod } (n+1)} \\ Q_{i \text{ Mod } (n+1)} \\ Q_{(i+1) \text{ Mod } (n+1)} \\ Q_{(i+2) \text{ Mod } (n+1)} \end{bmatrix} \quad \dots (3.6)$$

where $M_R(s)$ is defined in equation (3.5).

Figure 3-3 A closed cubic B-spline curve.

(B) Multiple Control Points

The B-spline curves of multiple control points are particularly interesting. Multiple control point means repeating the control point some number of times. Figure 3-4 shows the cases of double and triple control points. It can be observed that as long as in a triple control point, the control points immediately preceding and succeeding it fail to be collinear, the left and right derivative vectors at the triple control point also fail to be collinear; a corner is formed as shown in Figure 3-4(c).

It seems in the first impression that this violates the continuity and smoothness of the cubic B-spline curve. However, the first derivative vector at the knot point can be derived to be 0, and is still continuous there. As the derivative vectors just to the left and right of the knot point are in different directions, therefore a corner is formed. This B-spline’s property shows the B-spline has the capability to describe not only the smooth and round shape, but also sharp corners and concave shapes.

Figure 3-4 Forming a corner by tripling a control point. The double control point is not interpolated, while the triple control point is. (a) a single control point; (b) a double control point; (c) a triple control point.

(C) Shape Invariant under Affine Transformation

In many imaging systems, detected images are subject to geometric distortion introduced by perspective irregularities wherein the position of the camera(s) with respect to the object alters the apparent dimensions of the object geometry. Applying an affine transformation to a uniformly distorted image can correct for a range of perspective distortions by transforming the measurements from the ideal coordinates to those actually used. (For example, this is useful in satellite imaging where geometrically correct ground maps are desired.).

The general affine transformation is commonly written in homogeneous coordinates as shown below:

$$\begin{bmatrix} x'(t) & y'(t) \end{bmatrix} = \begin{bmatrix} x(t) & y(t) \end{bmatrix} \times A + B \qquad \dots(3.7)$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 \end{bmatrix} \text{ and } A \text{ is non-singular} \quad \dots(3.8)$$

For a B-spline curve that is

$$g_i(s) = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix}, \quad i = 1, 2, \dots, n-2 \quad \dots(3.9)$$

After an affine transformation on the B-spline curve, it becomes:

$$g'_i(s) = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \times A + B \quad \dots(3.10)$$

Since

$$\begin{aligned} M_R(s) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} &= \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & -1 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = 1 \quad \dots(3.11) \end{aligned}$$

Therefore the equation (3.10) can be rewritten as:

$$g'_i(s) = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \times A + M_R(s) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} B \quad \dots(3.12)$$

and then we can obtain

$$\begin{aligned} g'_i(s) &= M_R(s) \left(\begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix} \times A + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} B \right) \\ &= M_R(s) \begin{bmatrix} Q_{i-1} \times A + B \\ Q_i \times A + B \\ Q_{i+1} \times A + B \\ Q_{i+2} \times A + B \end{bmatrix} \quad \dots(3.13) \end{aligned}$$

Equation (3.14) shows that, the affine invariance property also holds for B-spline curves. If an affine transformation is applied to a B-spline curve, the result can be constructed from its affine transformed control points. When we want to apply a geometric or even affine transformation to a B-spline curve, this property states that we can simply apply the transformation to control points, and once the transformed control points are obtained, the transformed B-spline curve is the one defined by these new points. Therefore, we do not have to transform the curve.

(D) Relationship Between Knot Points and Control Points

Given the set of knot points $P = (P_1, P_2, \dots, P_{n-1})$ of a uniform cubic B-spline curve (Figure 3-1), we can uniquely determine the control points $Q = (Q_0, Q_1, \dots, Q_n)$, by substituting $s = 0$ into equation (3.3). The relationship between the control points and the knot points is given as [152]:

$$P_i = \frac{1}{6}Q_{i-1} + \frac{2}{3}Q_i + \frac{1}{6}Q_{i+1}, \quad i = 1, 2, \dots, n-1 \quad \dots(3.14)$$

The matrix form of equation (3.14) could be expressed as:

$$P = AQ \quad \dots(3.15)$$

where A is a $(n-2) \times n$ matrix:

$$A = \begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \dots & \dots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ 0 & \dots & \dots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix} \quad \dots(3.16)$$

From equation (3.15) we can also get:

$$Q = A^{-1}P \quad \dots(3.17)$$

To solve equation (3.17), two more equations have to be known. The following conditions are applied for open curves and closed curves, respectively:

(a) double the first and the last control points for open curve generation:

$$Q_0 = Q_1, \quad Q_n = Q_{n-1} \quad \dots(3.18)$$

- (b) overlap the first two control points to the last two control points, respectively, for closed curve generation:

$$Q_0 = Q_{n-1}, \quad Q_1 = Q_n \quad \dots(3.19)$$

By adding the above conditions, the number of equations is made equal to the number of unknowns, and the simultaneous equation (3.15) can be solved.

3.2.2 Insertion of Knot Points

A B-spline curve segment can be divided into two curve segments at an arbitrary point while maintaining its shape. This property is very convenient in curve design when it is necessary to apply fine shape control.

However, there are three extra control points that have to be added in for maintaining its shape [153]. Also, it causes the extra calculation and discontinuity of control points since the added three control points are only for the divided segment. In our B-snake model, the curve shape formed by a new set of control points is not necessary of the exactly the same as the old one, as the B-snake deformation will adjust the control points of the B-spline curve and form the desired shape afterward. Here we have developed a simple knot insertion algorithm and present below.

Suppose that we have a segment of cubic B-spline curve between the knots P_i and P_{i+1} , it is defined on a set of four control points $(Q_{i-1}, Q_i, Q_{i+1}, Q_{i+2})$. A new knot $\hat{P}_i$ is expected to be inserted between P_i and P_{i+1} at s_i . The new set of control points which has five control points is $(\hat{Q}_{i-1}, \hat{Q}_i, \hat{Q}_{i+1}, \hat{Q}_{i+2}, \hat{Q}_{i+3})$ and it is determined by:

$$\hat{Q}_j = (1 - \alpha_j) Q_{j-1} + \alpha_j Q_j \quad (i-1 \leq j \leq i+3) \quad \dots(3.20)$$

where

$$\alpha_j = \begin{cases} 1 & (j = i-1) \\ \frac{s_i + (j-i)}{3} & (i \leq j \leq i+2) \\ 0 & (j = i+3) \end{cases} \quad \dots(3.21)$$

Equations (3.20) and (3.21) show that the new control point only affects the location of four control points that form the curve segment in which the new knot point has been inserted. This control point insertion algorithm has been implemented in the B-spline shown in Figure 3-5, the B-spline with a new control point is shown in Figure 3-6.

Figure 3-5 B-spline needs to insert a new knot point.

Figure 3-6 B-spline after knot point insertion.

Figure 3-7 Comparison of two B-splines curves before and after control points insertion.

As shown in Figure 3-7, the B-spline after knot insertion is slightly different than the original one. However, we will show that it does not affect the result of B-snake deformation in next chapter.

3.2.3 Why B-Splines ?

In light of the above underlying structure of the B-splines, the following attractive properties make them suitable for curve representation and analysis. These are:

- (1) **Compact representation** - The number of variables to be estimated is reduced to the number of control points. This makes the B-spline stand as one of the most efficient curve representation and have been widely used in practical implementations, such as curve generation, data compression and transmission.
- (2) **Smoothness and continuity** - This allows any curve to consist of a concatenation of curve segments, yet be treated as a single unit. Smoothness is guaranteed by implicit constraints in the representation. It is particularly suitable for deformable

model in image processing, as the internal force for maintaining the smoothness of the model is not necessary.

- (3) Shape invariance under affine geometric transformation - This means that the B-spline representations of two sets of control points that are related through an affine transform (scaling, rotation and translation), are related by the same affine transform as well. As affine transformation is assumed frequently in image mapping [139][144][172][173], the B-spline is a very attractive curve model to adapt to this case.
- (4) Continuous formula expression - This provides a quick and simple way to perform the extraction of some features of B-spline curve such as curvature, affine length and inflection point. As it is based on the whole curve rather than the discrete points of point-based curve models, it thus avoids those errors such as the errors of sampling and digitalization, and improves accuracy.
- (5) Local controllability - This implies that local changes in curve are confined to the B-spline parameters local to that change. Local control makes it possible to modify part of a curve without affecting other portions that are already satisfactory. This property is particular useful in object tracking, as normally only part of the object is changed in this case.
- (6) Separate parametric representation - The decoupling of the x, y , and z coordinates, with each having its parametric representation, is treated separately. This means a B-spline curve can be easily extended from 2-D curve to 3-D by just simply adding in one portion for the third coordinate and it will not be affecting the representations in other coordinates.
- (7) Generative nature [171] – Due to continuous equation expression, the B-spline curve can be totally generated to any degree of detail desired (i.e. different sampling rates or scales) once the control points are known. This means if the control points of a B-spline curve are obtained, we could get any desired number of points that reside on the B-spline curve.

3.3 On Active Contour - Snake

Snakes, or active contours, are curves defined within an image domain which can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. Snakes are widely used in many applications, including edge detection [2], shape modeling [112]-[113], segmentation [114]-[115], and motion tracking [114][116].

A traditional snake [2] is a parametric contour $r(s) = (x(s), y(s))$, $s \in [0, 1]$, in the image plane (x, y) . It moves through the spatial domain of an image to minimize the energy functional

$$E = \int_0^1 \frac{1}{2} (\alpha |r_s(s)|^2 + \beta |r_{ss}(s)|^2) + E_{ext}(r(s)) ds \quad \dots(3.22)$$

where α and β are weighting parameters that control the snake tension and rigidity, respectively, and $r_s(s)$ and $r_{ss}(s)$ denote the first and second derivatives of $r(s)$ with respect to s . The external function E_{ext} is derived from the image so that it takes on its smaller values at the features of interest, such as boundaries. Given a gray-level image $I(x, y)$, viewed as a function of continuous position variables (x, y) , typical external energies designed to lead an active contour toward step edges are [2]:

$$E_{ext}^{(1)}(x, y) = -\|\nabla I(x, y)\|_2 \quad \dots(3.23)$$

$$E_{ext}^{(2)}(x, y) = -\|\nabla(G_\sigma(x, y) * I(x, y))\|_2 \quad \dots(3.24)$$

where $G_\sigma(x, y)$ is a two-dimensional Gaussian function with standard σ and ∇ is the gradient operator. If the image is a line drawing (black on white), then appropriate external energies include [117]:

$$E_{ext}^{(3)}(x, y) = -I(x, y) \quad \dots(3.25)$$

$$E_{ext}^{(4)}(x, y) = -G_{\sigma}(x, y) * I(x, y) \quad \dots(3.26)$$

It is easy to see from these definitions that larger σ 's will cause the boundaries to become blurry. Such large σ 's are often necessary, however, in order to increase the capture range of the active contour.

A snake that minimizes E must satisfy the Euler-Lagrange equations [2]:

$$\alpha r_{ss}(s) - \beta r_{ssss}(s) - \nabla E_{ext} = 0 \quad \dots(3.27)$$

This can be viewed as a force balance equation:

$$F_{int} + F_{ext}^{(p)} = 0 \quad \dots(3.28)$$

where $F_{int} = \alpha r_{ss}(s) - \beta r_{ssss}(s)$ and $F_{ext}^{(p)} = -\nabla E_{ext}$. The internal force F_{int} discourages stretching and bending while the external potential force $F_{ext}^{(p)}$ pulls the snake towards the desired image edges.

To find a solution to (3.27), the snake is made dynamic by treating r as function of time t as well as s i.e., $r(s, t)$. Then, the partial derivative of r with respect to t is then set equal to the left-hand side of (3.27) as follows:

$$r_t(s, t) = \alpha r_{ss}(s, t) - \beta r_{ssss}(s, t) - \nabla E_{ext} \quad \dots(3.29)$$

when the solution $r(s, t)$ stabilizes, the term $r_t(s, t)$ vanishes and we achieve a solution of (3.27).

To solve (3.27) numerically, first we separate (3.27) into a pair of independent Euler-Lagrange equations,

$$\alpha x_{ss}(s) - \beta x_{ssss}(s) - \frac{\partial E_{ext}}{\partial x} = 0 \quad \dots(3.30)$$

$$\alpha y_{ss}(s) - \beta y_{ssss}(s) - \frac{\partial E_{ext}}{\partial y} = 0 \quad \dots(3.31)$$

Then the Euler equations, with $f_x(i) = \frac{\partial E_{ext}}{\partial x_i}$ and $f_y(i) = \frac{\partial E_{ext}}{\partial y_i}$ discretized, yield

$$\begin{aligned} \alpha_i(r_i - r_{i-1}) - \alpha_{i+1}(r_{i+1} - r_i) + \beta_{i-1}(r_{i-2} - 2r_{i-1} + r_i) \\ - 2\beta_i(r_{i-1} - 2r_i + r_{i+1}) \\ + \beta_{i+1}(r_i - 2r_{i+1} + r_{i+2}) + (f_x(i), f_y(i)) = 0 \end{aligned} \quad \dots(3.32)$$

Writing the equation in matrix forms, one for x and another for y , yields

$$Ax + f_x(x, y) = 0 \quad \dots(3.33)$$

$$Ay + f_y(x, y) = 0 \quad \dots(3.34)$$

where B is a pentadiagonal banded matrix [2]. We can now position vectors iteratively,

$$x_t = (B + \gamma I)^{-1}(\gamma x_{t-1} - f_x(x_{t-1}, y_{t-1})) \quad \dots(3.35)$$

$$y_t = (B + \gamma I)^{-1}(\gamma y_{t-1} - f_y(x_{t-1}, y_{t-1})) \quad \dots(3.36)$$

where γ is a step size.

3.4 On the B-Snake Model

The traditional snake is a point-based deformable model, the curve is represented by a series of discrete points. It is deformed by the external and internal forces that are calculated from an energy function. From the original approach of snake, an alteration is using a parametric B-spline representation as curve descriptor. Based on the discussion

in section 3.2, compared to traditional point-based snake, the B-snake greatly reduces the number of state variables required for a snake.

B-snake can be used for any order. However, we are mainly concerned about implementing the cubic B-spline to B-snake in this thesis. In [127], we have presented an open B-snake model with a fixed number of control points for lane detection and tracking. Actually, the B-snake can be either open or closed and defined as follows:

$$r(s) = \sum_i g_i(s) \quad \dots(3.37)$$

For a cubic B-snake, $g_i(s)$ is defined by equation (3.4) or (3.6) for open or closed curve respectively. The number of control points can be flexible to adapt to the structure of object shape. The flexibility of the curve increases as more control points are added; each allows either one more inflection in curve or, when multiple knots are used, reduced continuity at one point. The control points are positioned so that the curve locates the desired image contour.

The external energy term on $r(s)$ is defined as $E(r(s))$. Therefore, the total energy function of the B-snake $E_{B\text{-snake}}$ can be defined by integrating $E(r(s))$ along the B-snake. That is,

$$E_{B\text{-snake}} = \int_0^1 E(r(s)) ds \quad \dots(3.38)$$

There are no internal forces required in B-snake since the B-spline representation maintains smoothness via hard constraints implicit in the representation. The external force for B-snake can be defined as:

$$F(x, y) = \nabla(G_\sigma(x, y) * I(x, y)) \quad \dots(3.39)$$

When the external forces along B-snake is vanishing, the B-snake will be stabilizing close to a contour of edges. There is an alternative to define the external force at each sample point proportional to the distance between the sample and the target points.

However, it is very time consuming as every time it has to search the closest points and calculate the distance.

In conclusion, B-snake has two advantages over traditional snake:

- (1) B-snake uses fewer parameters. As the cubic B-spline is proposed to be used, deformable curves are represented by four or more parameters (control points).
- (2) Internal forces are not required in deforming B-snake since the representation of B-spline has guaranteed smoothness by implicit constraints.

However, although B-snake has the above mentioned advantages, the disadvantages of the B-snake are from the B-spline. They are:

- (1) The number of control points is difficult to determine for describing an unknown or complex shape.
- (2) The prior knowledge of object shape is hard to be implemented in B-snake for guiding the deformation of B-snake.

One of the objectives of this research work is to develop a B-snake model for object segmentation by making use of the above advantages and overcome the disadvantages as well.

3.5 Application of B-Snake for Object Boundary Detection

As described in previous sections, B-spline has a good capability to represent any shape of object. Therefore, it is very suitable for object boundary detection and object segmentation. However, to apply the B-snake to real applications, some problems listed below have to be solved for B-snake:

- (1) The more control points are given, the more complex shape B-spline can be formed. However, how to automatically determine the number of control points for B-snake to adapt to the structures of desired object contour, rather than just using a fixed

number of control points for a variety of shapes, is a hard task. The solution should be faster and be able to use existing conditions to conduct this purpose.

- (2) The shape of B-spline curve is determined only by a set of control points. Therefore, in the deformation of B-snake, how to efficiently transfer the external forces to the control points and position the B-snake, remains a challenge. It should be able to co-operate with the solution of (1) in the deformation procedure.
- (3) In the case when the prior knowledge of object is available, how to extract and model the features of objects, and implement them to guide the B-snake deformation, is a challenging task.

In the next three chapters, the solutions for the above mentioned problems will be addressed. Figure 3-8 shows an example of B-snake in object segmentation. In this application, the B-snake is implemented to human ventricle detection in a real MR image. The B-snake model is first initialized by a minimum number of control points, which is four in the cubic B-snake. During the deformation procedure, an energy minimum method of MMSE and an adaptive control point insertion strategy proposed in the next chapter, is involved. The new control points are added to adapt the structure of the object and the MMSE deforms the B-snake to the object edges. As we can see in Figure 3-8(b), eventually the B-snake consists of a total of 17 control points and the closed curves formed by these control points are precisely fitted to the boundary of ventricle. More details will be introduced in chapter 3. Note this example does not involve any prior knowledge of object in detection. In chapter 5 and chapter 6, this issue will be addressed.

Figure 3-8 Using B-snake to detect the ventricle in MR image.

3.6 Concluding Remarks

In this chapter, the uniform cubic B-spline is introduced. A B-spline based deformable model, B-snake, is then presented. In B-snake, one set of control points is used to describe the object shape. Since B-spline has local controllability and it can form arbitrary shapes, this model can describe a much wider range of object shapes while still maintaining fewer variables than other point-base deformable models.

The B-snake model is also easy for object boundary tracking application since the parameters of B-snake for the current frame is usually similar to those in the previous frame, i.e. the movements of the control points are smaller. On the contrary, for other deformable models such as the second-order polynomial model, when the object shape has a small change, it may cause a big change in the parameters of the model. All the parameters of this B-snake model can be estimated in a single image by the proposed method which will be introduced in the next few chapters.

Chapter 4

Estimation of B-Snake Parameters from Image Data

4.1 Introduction

Based on the initial location of the control points that are determined either by manual or as a result of the previous frame, the B-snake would further deform to the desired contour accurately in the current frame. This chapter deals with the problems related to these processes.

There are some existing techniques for deforming a B-spline to desired object boundaries. The method presented in [84] for cubic B-spline model to approximate a planar curve is by employing the least-squares solution of the sum of squared distance between each sample point of B-spline and the nearest point of the shape curve for finding the best location of control points. However, for every iteration of deformation procedure, this method has to re-estimate the correspondence and recalculate the distance between the B-spline and the object shape. A similar solution is proposed in [85]-[86] for tracking image contours with a B-snake. Klein *et al.* presented an algorithm that is using Dynamic Programming to minimize the energy of cubic B-spline model in [101]. The iterative optimization algorithm begins with the first segment of B-spline, the locations of all four control points are varied within a limited window and energy corresponding to each conformation is calculated. These energies are minimized over

the first control point. The procedure is repeated for the next segment, the conformation of which is controlled by the last three control points of the previous segment and the next control point in the sequence. Energies corresponding to all of the possible locations of these four control points are calculated from the spline energy plus the minimum energies from the first segment. This scheme continues until all the control points have been varied, and the minima of all the remaining conformations can be sought. Stepping back through stored optimal control point locations reveals the lowest energy state conformation. This optimization process is repeated until the overall spline energy reaches a minimum. The same method has been used for B-snake to track the heart motion in [103]. The drawbacks of this method are: (1) when more control points are used, more steps have to be involved; (2) as all possible locations of every control point have to be explored, the computation time for performing dynamic programming is increased greatly. The computational complexity is $O(nm^3)$, where n is the number of control points and m is the size of the neighborhood in which a point can move during a single iteration.

In the method presented in [106]-[107] for B-spline Model to describe the unsupervised contour, the Minimum description Length (MDL) is used to estimate the order and the number of control points of B-spline model. A likelihood function which depends on the image data, having been split into inner and outer regions of the object of interest, is constructed to estimate the location of control points. This method is suitable for the case where the contour boundary is not significant, but the user has to indicate a point inside the object boundary and the algorithm is initialized with a small contour around that point (which is guaranteed to be completely inside the inner region). The number of control points is found by exhaustive search over a range of values.

In [127], I have presented an open B-snake lane model with a fixed number of control points and GVF has been used as its external force to detect the lane boundary by MMSE method. In this chapter, a new B-snake model is presented. Compared to the B-snake model in [127], it has a few novelties: (1) a control point insertion strategy is embedded to adapt the complex shape without prior fixing of the number of control points; (2) the normal portion of GVF is chosen to efficiently deform the B-snake; (3) the MMSE

method is implemented to both closed B-snake and open B-snake models. The experimental results and discussion of applications of the proposed B-snake model are presented in this chapter as well.

4.2 Shape Extraction Algorithm Using B-Snake

Currently the grey image is processed by using Canny edge detector to locate the position of pixels where significant edges exist as shown in Figure 4-1.

Figure 4-1 Canny edge detection.

Here we choose GVF [132] as the external force for B-snake to perform deformation since GVF has a larger capture range. In the next section, GVF will be introduced.

4.2.1 Gradient Vector Flow

As a new external force field for active contours, GVF was first introduced in [131]. It is computed as a diffusion of the gradient vector of a binary edge map $f(x,y)$ derived from the image.

GVF field is defined to be the vector field $w(x, y) = (u(x, y), v(x, y))$ that minimizes the energy functional [131]:

$$\varepsilon = \iint \mu (u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |w - \nabla f|^2 dx dy \quad \dots(4.1)$$

The parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand. Using calculus of variations [133], the GVF field can be found by solving the following Euler equations:

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \quad \dots(4.2)$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \quad \dots(4.3)$$

where ∇^2 is the Laplacian operator. Equations (4.2) and (4.3) can then be solved by treating u and v as a function of time and solving:

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y))(f_x^2(x, y) + f_y^2(x, y)) \quad \dots(4.4)$$

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - (v(x, y, t) - f_y(x, y))(f_x^2(x, y) + f_y^2(x, y)) \quad \dots(4.5)$$

The steady-state solution of these linear parabolic equations is the desired solution of the Euler equations (4.2) and (4.3). The equations in (4.4) and (4.5) are known as generalized diffusion equations, and are known to arise in such diverse fields as heat conduction, reactor physics and fluid flow [136].

We implement 2-D GVF by MATLAB codes to compute the detected edge image. For a 240x256-pixel image, it takes 0.5 seconds. The GVF force superimposes on edge image was shown in Figure 4-2.

Figure 4-2 External force using the GVF. The arrows represent the direction and magnitude of the GVF force. It can be observed that the GVF force points to the nearest edge point and the location far away from the edge has a strong force.

4.2.2 External Forces for B-Snake

Moreover, by using external forces, we can deform the B-snake. The B-snake external forces F_{ext} are calculated by the sum of external force on the sampled points of the B-spline curve model which are defined as $F_B(s)$:

$$F_{ext} = \sum (F_B(s)) \quad \dots(4.6)$$

However, as we found that, the projection of $F_B(s)$ at the tangent axis of each sampled point on the B-snake does not help to deform the B-snake to desired boundary. Hence,

in our practical implementation, $F_B(s)$ is replaced by $\hat{F}_B(s)$ which is the projection of $F_B(s)$ at the normal axis of that sampled point on the B-spline curve and shown in Figure 4-3.

$$\hat{F}_B(s) = |F_B(s)| \cdot \cos \theta \cdot V_{normal} \quad \dots(4.7)$$

where θ is the angle between the vector of external force and normal axis of sampled point on the B-snake curve, and V_{normal} is the unit vector of the normal axis. Therefore, equation (4.6) becomes:

$$F_{ext} = \sum (\hat{F}_B(s)) \quad \dots(4.8)$$

A real example of B-snake using the above external force is shown in Figure 4-5. Figure 4-4 shows the original GVF external force $F_B(s)$ while Figure 4-5 shows the modified external force $\hat{F}_B(s)$ for the sample points of B-snake. $\hat{F}_B(s)$ is assigned the portion of $F_B(s)$ at the normal axis of that sampled point on the B-snake, it would push the B-snake to the desired boundary more efficiently.

Figure 4-3 Demonstration of $\hat{F}_B(s)$.

Figure 4-4 External force $F_B(s)$.

Figure 4-5 External forces $\hat{F}_B(s)$.

4.2.3 Minimum Mean Square Error Approach

The advantage of using B-snake is that internal forces are not required since the B-snake representation maintains smoothness via hard constraints implicit in the representation.

B-snake should be updated to minimize: (1) the sum of the external forces from the B-spline for achieving accurate position of B-snake, and (2) the external forces should be transmitted to each control point when updating B-snake. The details are described as follows.

To deform the B-snake to the object boundaries, the energy function has to be minimized as:

$$E_{ext} = \min \left(\sum \left(\hat{E}_B(s) \right) \right) \quad \dots(4.9)$$

Once the energy function of B-snake is minimized, then no movement is driven for control points and hence no change in the shape of the B-snake. To iteratively minimize the external force, the following equation is defined to link the movement of control points to the energy function:

$$E_{ext} = \gamma M_R(s) \Delta Q(t) \quad \dots(4.10)$$

where γ is a step-size and $\Delta Q(t)$ is the adjustment of the control points Q in each iteration step.

$$Q(t) = Q(t-1) + \Delta Q(t) \quad \dots(4.11)$$

External force can be sampled along the B-spline of B-snake at a certain distance apart (we are using 2 ~ 3 pixels). Then equation (4.10) can be solved digitally. Here, the MMSE solution for the digital version of the equation (4.10) is given as a matrix form.

$$\Delta Q(t) = \gamma^{-1} \left[M^T M \right]^{-1} M^T E_{ext} \quad \dots(4.12)$$

where

(i) for the case of open cubic B-snake [127]:

$$M = \begin{bmatrix} M_{-1} & 0 & \dots & \dots & 0 \\ 0 & M_0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & M_{n-1} & 0 \\ 0 & \dots & \dots & 0 & M_n \end{bmatrix} \quad \dots(4.13)$$

and

$$M_i = \begin{bmatrix} s_{i,1}^3 & s_{i,1}^2 & s_{i,1} & 1 \\ s_{i,2}^3 & s_{i,2}^2 & s_{i,2} & 1 \\ \cdot & \cdot & \cdot & \cdot \\ s_{i,m}^3 & s_{i,m}^2 & s_{i,m} & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} & 0 \end{bmatrix} \quad \dots(4.14)$$

where m is the sampling points number in i th segment of the B-spline. E_{ext} is the force vector digitized on the B-spline.

(ii) for the case of closed cubic B-snake:

$$M = \begin{bmatrix} M_1 & 0 & \dots & \dots & 0 \\ 0 & M_2 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & M_{n-3} & 0 \\ 0 & \dots & \dots & 0 & M_{n-2} \\ M_{n-1} & 0 & \dots & 0 & M_{n-1} \\ M_n & 0 & \dots & 0 & M_n \\ M_{n+1} & 0 & \dots & 0 & M_{n+1} \end{bmatrix}, \quad \dots(4.15)$$

where

$$M_i = \begin{bmatrix} s_{i,1}^3 & s_{i,1}^2 & s_{i,1} & 1 \\ s_{i,2}^3 & s_{i,2}^2 & s_{i,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{i,m}^3 & s_{i,m}^2 & s_{i,m} & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix}, \quad i = 1, 2, \dots, n-2.$$

$$M_{n-1} = \begin{bmatrix} s_{n-1,1}^3 & s_{n-1,1}^2 & s_{n-1,1} & 1 \\ s_{n-1,2}^3 & s_{n-1,2}^2 & s_{n-1,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n-1,m}^3 & s_{n-1,m}^2 & s_{n-1,m} & 1 \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{6} & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & -1 & \frac{1}{2} \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix},$$

$$M'_{n-1} = \begin{bmatrix} s_{n-1,1}^3 & s_{n-1,1}^2 & s_{n-1,1} & 1 \\ s_{n-1,2}^3 & s_{n-1,2}^2 & s_{n-1,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n-1,m}^3 & s_{n-1,m}^2 & s_{n-1,m} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_n = \begin{bmatrix} s_{n,1}^3 & s_{n,1}^2 & s_{n,1} & 1 \\ s_{n,2}^3 & s_{n,2}^2 & s_{n,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n,m}^3 & s_{n,m}^2 & s_{n,m} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -\frac{1}{6} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -1 \\ 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix},$$

$$M'_n = \begin{bmatrix} s_{n,1}^3 & s_{n,1}^2 & s_{n,1} & 1 \\ s_{n,2}^3 & s_{n,2}^2 & s_{n,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n,m}^3 & s_{n,m}^2 & s_{n,m} & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{1}{6} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 & 0 \end{bmatrix},$$

$$M_{n+1} = \begin{bmatrix} s_{n+1,1}^3 & s_{n+1,1}^2 & s_{n+1,1} & 1 \\ s_{n+1,2}^3 & s_{n+1,2}^2 & s_{n+1,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n+1,m}^3 & s_{n+1,m}^2 & s_{n+1,m} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix},$$

$$M'_{n+1} = \begin{bmatrix} s_{n+1,1}^3 & s_{n+1,1}^2 & s_{n+1,1} & 1 \\ s_{n+1,2}^3 & s_{n+1,2}^2 & s_{n+1,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ s_{n+1,m}^3 & s_{n+1,m}^2 & s_{n+1,m} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{1}{6} & 0 \\ -1 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ \frac{2}{3} & \frac{1}{6} & 0 & 0 \end{bmatrix}.$$

and $s_{i,m}$ is the parameter s to form the sampling point m in i th segment.

4.2.4 Adaptive Knot Point Insertion Strategy

To adaptively approach a complex contour without prior fixing of the number of control points, a structure-adaptive knot point insertion strategy is developed [129]. The location of new knot point is determined by checking the external force on the B-snake when B-snake has converged in the deforming procedure. The new knot point is inserted into the location where it has maximal external force on the B-spline curve. In the GVF field, the magnitude of the force reflects the distance to the nearest significant edge. The stronger external force means the farther distance to the nearest object boundary. In this case, the knot point should be inserted to increase the flexibility of the B-spline curve to adapt the object shape. Compared to other algorithms [105]-[107] which have to exhaustive search over a range of value for determining the number of control points along the entire B-spline contour, this method is simple and works fast. Once we compute the location of new knot point to be inserted, the new sequence of control points and knot points can be generated by the algorithm presented in section 3.2.2.

4.2.5 The Complete Object Contour Detection Algorithm

In order to arrive at the solutions in equations (4.11), an iterative procedure is adopted. The steps contained in this iterative minimization process are as follows:

- (1) Calculate the edge image of original image by using Canny edge detector [182]. Various edge detectors can be implemented here. Canny has been chosen, because it produces superior result compared to other edge detectors.
- (2) Calculate the GVF of the edge image as the external force of B-snake.
- (3) Initialize the control point parameters.
- (4) Compute the MMSE in equation (4.12) for obtaining $\Delta Q(t)$.
- (5) If $\|\Delta Q(t)\| > threshold_1$, set $Q(t)$ to $Q(t-1)$, then go to step 4; Otherwise, go to step 6.
- (6) If $\max(\|E_B(t)\|) > threshold_2$, a new knot point is inserted in that location which has maximum external force, then go to step 4; Otherwise, go to step 7.
- (7) Stop. The last estimates of $Q(t)$ are regarded as the final results.

Canny edge detector [182] is employed, as it produces the superior result compared to other edge algorithms. This iterative minimization is guaranteed to converge to at least a local minimum of E_{ext} in equations (4.10).

4.3 Experimental Results and Discussion

This B-snake model has been simulated by MATLAB codes and tested on real images. The goal in this experiment is to extract the object boundaries without a prior knowledge of the studied object. Two experimental results are presented here. One is for the lane detection, which is the case of open curve; another is for the human brain ventricle detection in MR images, which is the case of closed curve.

4.3.1 Experimental Results on Complex Lane Detection - The Case of Open B-Snake Model

The goal of this experiment is to test the proposed B-snake Model in the case of open shape detection: Lane detection. We have previously proposed a B-snake based lane model [125]-[128], which describes the perspective effect of parallel line with dual external forces for generic lane boundary or marking. This model is able to describe a wider range of lane structures than other lane models such as straight and parabolic models. However, the B-snake lane model is fixed to three different location's control points, Q_0 , Q_1 and Q_2 (Q_0 and Q_2 are triple, so we actually compute a curve of 2 segments from the sequence of 7 control points $Q_0, Q_0, Q_0, Q_1, Q_2, Q_2, Q_2$). Although it is sufficient for most roads, it lacks the flexibility to model the complex shape of some roads. The proposed algorithm in section 4.2 is applied to the B-snake lane model to perform the complex road detection without fixing the number of control points [130]. Below, the B-snake lane model is briefly introduced and then the results are presented.

(A) B-Snake Lane Model

The full derivation of lane model can be found in [127], here we only present the lane model. Define the mid-line of the road in the image plane as

$$L_{mid} = (c_m, r_m) \quad \dots(4.16)$$

Thus the left side of the modeled road is

$$L_{left} = (c_l, r_l) \quad \dots(4.17)$$

where

$$c_l = c_m - \frac{1}{2}d = c_m - \frac{1}{2}k(r_l - hz) \text{ and } r_l = r_m. \quad \dots(4.18)$$

and k is a variable to reflect the width of road, hz is the position of vanishing line in the image plane. Similarly, the right side of the modeled road is

$$L_{right} = (c_r, r_r) \quad \dots(4.19)$$

where

$$c_r = c_m + \frac{1}{2}d = c_m + \frac{1}{2}k(r_r - hz) \text{ and } r_r = r_m. \quad \dots(4.20)$$

We use a set of control points to describe the mid-line of the road by B-spline and an additional parameter k to determine the left and the right sides of road model. In order to make B-splines pass through the first and the last control points, the first control point and the last control points are tripled. The mid-line of road model can be expressed by a B-spline as

$$L_{mid} = (c_m, r_m) \\ = M_R(s) \begin{bmatrix} Q_{i-1} \\ Q_i \\ Q_{i+1} \\ Q_{i+2} \end{bmatrix}, \quad i = -1, 0, 1, 2, \dots, n. \quad \dots(4.21)$$

The mid-line of lane model can be deformed by the external forces $E_{M_sum}(s)$, which is the sum of the *dual external forces* calculated from both the left and the right sides of lane model, $E_L(s)$ and $E_R(s)$.

$$E_{ext} = E_{M_sum}(s) = E_L(s) + E_R(s) \quad \dots(4.22)$$

Also, the difference of horizontal components of $E_L(s)$ and $E_R(s)$, denoted as $E_{M_dif}^c(s)$, would lead to adjustment of the parameter k .

$$E_k = E_{M_dif}^c = (E_L^c(s) - E_R^c(s)) \quad \dots(4.23)$$

(B) Lane Detection Algorithm with Knot Point Insertion Strategy

The shape extraction algorithm presented in section 4.2.5 has to be modified to adapt the case of lane model. The steps contained in this iterative minimization process are as follows [130]:

- (1) Initialization Step. Initialize the control point parameters by CHEVP algorithm [127]. The number of control point is set to three at this moment.
- (2) Calculate the GVF of the edge road image as the external force of the B-snake.
- (3) Calculate MMSE for obtaining $\Delta Q(t)$ and $\Delta k(t)$, respectively.
- (4) If $\|\Delta Q(t)\| > threshold_1$ and $\|\Delta k(t)\| > threshold_2$, set $Q(t)$ to $Q(t-1)$ and $k(t)$ to $k(t-1)$, then go to step 3; Otherwise, go to step 5.
- (5) If $\max(\|E_{ext}(t)\|) > threshold_3$, a new knot is inserted in that location which has maximum external force, then go to step 3; Otherwise, go to step 6.
- (6) Stop. The last estimations of $Q(t)$ and $k(t)$ are regarded as the final results.

(C) Lane Detection Results and Discussion

The lane detection algorithm presented in last section has been tested on real road images by using Matlab codes in PIII 660Mhz system. The threshold values in this algorithm, $threshold_1$, $threshold_2$ and $threshold_3$ are set to 0.05, 0.01 and 0.4, respectively. These values are chosen in such way that optimum object boundary is obtained based on our experience while computation time to achieve this is acceptable. This is independent of complexity in the object shape. The thresholds can not be set too strict, because the camera focus is fixed, the lane on the ground projected on the image is not all the way

clear. See Figure 4-6, the lane closed to camera (bottom portion of image) is clear in boundary, but it is blur on the top portion of image, which is far away from the camera. Too tight thresholds would cause the processing time increased greatly. Moreover, this lane model is designed to approximate the boundary of lane based on some assumptions (parallel line and constant width), in some cases the lane may not satisfy the assumptions perfectly, too strict thresholds would cause the iteration never meet the thresholds. The initialization for proposed B-snake lane model is obtained from CHEVP (Canny/Hough Estimation of Vanishing Points) in [127]. In the beginning, the lane model is initialized by three control points, Q_0 , Q_1 and Q_2 .

A “S” shape road with painted line is shown in Figure 4-6 for detection. The B-snake lane model is initialized by three control points. In the deforming procedure that is shown in Figure 4-7, we can observe that the B-snake is deformed to the road boundaries mainly on the bottom part of image. This is because this part of road is closed to the camera and is relatively straight with clear edges. Due to the lack of flexibility of 3 control points of B-snake for describing the complex shape, the B-snake lane model cannot approach the top part of the road which is far from the camera and quite curved (Figure 4-8).

Figure 4-6 An initialization of B-snake lane model for “S” shape road with painted lane.

Figure 4-7 Deformation of B-snake with three control points.

Figure 4-8 Result of B-snake with three control points.

A new knot point is then inserted on the location where has maximum external force on the B-snake curve, shown in Figure 4-9. The B-snake lane model with new knot point inserted is shown in Figure 4-10.

Figure 4-9 The magnitude of external forces on the B-snake with three control points.

Figure 4-10 One new knot point (or control point) has been inserted.

The B-snake with four different locations of control points has much more flexibility in deforming the model to real road boundary as shown by the deformation of B-snake in the top part of Figure 4-11. The final result is shown Figure 4-12. We could find that the lane boundary detection in the top part of image, where the road is far from the

camera, is less accurate compared to the result in the bottom part of image due to the blurred road edges far away, but the result is still acceptable. The external forces of B-snake are shown in Figure 4-13, all are below the $threshold_3$ which is set to 0.4. The whole lane detection procedure takes around 1.2 second. It takes 3.3 second if the $threshold_3$ is set to 0.3, and it will never converge if the $threshold_3$ is set to 0.2.

Figure 4-11 B-snake deformation with four control points.

Figure 4-12 Final result of B-snake with four control points.

Figure 4-13 The magnitude of external forces on the B-snake with four control points.

Figure 4-14 presented another lane detection result on wet lane with painted line. This image was taken just after raining. Therefore, the light reflection can be seen on the road surface. Again, the lane model is initialized by three control points shown in Figure 4-14(a). One more knot point is inserted. The final result is shown in Figure 4-14(b). It is quite good although some part may not be exactly mapping to the road boundary. This is due to the limitation of 2-D lane model since the proposed lane model in [127] is designed only for the lane on planar ground. In this image, the lane can be observed that it is on a slight hill. To construct a 3-D lane model is beyond the scope of this thesis and will be our research work in the future.

One example of applying this algorithm to an unpainted road is shown in Figure 4-16. It is a path in a park. The detection result shows that the B-snake model matches well to the lane boundaries even though the boundaries are not smooth due to disturbance by the grass.

The experiments presented in this section show that the B-snake model with a control point insertion strategy is working well for the open curve. However, it can be

implemented to closed curve as well. In the next section, the experiments on closed B-snake model will be presented.

Figure 4-14 Lane detection for wet and painted road.

(a) Initialization of B-Snake lane model

(b) Lane detection result for unpainted lane

Figure 4-15 Lane detection for unpainted lane.