

NEUROANATOMICAL SEGMENTATION IN MRI EXPLOITING A PRIORI
KNOWLEDGE

by

KAILI

A DISSERTATION

Presented to the Department of Computer
and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2007

“Neuroanatomical Segmentation In MRI Exploiting A Priori Knowledge,” a dissertation prepared by Kai Li in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science. This dissertation has been approved and accepted by:

Dr. Allen D. Malony, Chair of the Examining Committee

Date

Committee in charge: Dr. Allen D. Malony, Chair
 Dr. Dejing Dou
 Dr. Kent Stevens
 Dr. Don M. Tucker

Accepted by:

Dean of the Graduate School

© 2007 Kai Li

An Abstract of the Dissertation of
Kai Li for the degree of Doctor of Philosophy
in the Department of Computer and Information Science
to be taken March 2007
Title: NEUROANATOMICAL SEGMENTATION IN MRI EXPLOITING A
PRIORI KNOWLEDGE

Approved: _____
Dr. Allen D. Malony

Neuroanatomical segmentation is a problem of extraction of a description of particular neuroanatomical structures of interest that reflects the morphometry (shape measurements) of the subject's neuroanatomy from any image rendering the neuroanatomical structures of the subject. This dissertation presents a set of algorithms for automatic extraction of cerebral white matter (WM) and gray matter (GM) as well as reconstruction of cortical surfaces from T1-weighted MR images.

Neuroanatomical segmentation presented in this dissertation is performed by an image analysis pipeline that steps through five major procedures: 1) the original MR image is processed by a new *relative thresholding* procedure and a new *terrain analysis* procedure such that all voxels are classified into one of the three types: WM, GM, and background; 2) the topology defects of the WM are eliminated by a new *multiscale morphological topology correction* algorithm; 3) cerebral WM is extracted from its superset with a new

procedure called *cell-complex-based morphometric analysis*; 4) cerebral GM is extracted based on the prior cerebral WM extraction with a set of morphological image analysis procedures; and 5) cortical surfaces are finally reconstructed preserving correct topology with an existing marching cube isosurface algorithm.

In this dissertation, we evaluated our neuroanatomical segmentation tool both quantitatively and qualitatively on a set of MR images with groundtruth or manual segmentation, compared the results of our tool with those of four other tools, and demonstrated that the performance of our tool is highly accurate, robust, automatic and computationally efficient.

The advantages of our tool are mainly attributed to extensive exploration of various structural, geometrical, morphological, and radiological *a priori* knowledge, which persists despite of image artifacts and inter-subject anatomical variations. By exploiting *a priori* knowledge, we also demonstrated that performing voxel classification prior to brain extraction is a promising research direction, contrary to the traditional procedure of brain extraction followed by voxel classification. Finally, it's worth noting that the algorithms of voxel classification and morphological image analysis presented in this dissertation for neuroanatomical segmentation can be potentially applied in wider areas in computer vision.

CURRICULUM VITAE

NAME OF AUTHOR: Kai Li

PLACE OF BIRTH: Jiangsu Province, P. R. China

DATE OF BIRTH: May 6, 1976

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, USA, 2001-2007

Nanjing University, Nanjing, China, 1993-2000

DEGREES AWARDED:

Doctor of Philosophy, 2007, University of Oregon

Master of Science, 2004, University of Oregon

Master of Science, 2000, Nanjing University

Bachelor of Science, 1997, Nanjing University

AREAS OF SPECIAL INTEREST:

Medical image analysis

PROFESSIONAL EXPERIENCE:

Research Assistant, Department of Computer and Information Science, University of Oregon, 2001-2006

Software Engineer, VHSOFT Technology Co. Ltd., Nanjing, P. R. China, 2000-2001

Research Assistant, Nanjing University, P.R. China, 1997-2000

PUBLICATIONS:

K. Li, A. D. Malony, and D. M. Tucker, "A Multiscale Morphological Approach to Topology Correction of Cortical Surfaces," in *Proceedings of the Third International Workshop on Medical Imaging and Augmented Reality - MIAR 2006*, pp. 52-59, August, 2006, Shanghai, China.

K. Li, A. D. Malony, and D. M. Tucker, Poster, "Automatic Brain MR Image Segmentation by Relative Thresholding and Morphological Image Analysis," in *12th Annual Meeting of Human Brain Mapping - HBM 2006*, June 11-15, 2006, Florence, Italy.

K. Li, A. D. Malony, and D. M. Tucker, "Automatic Brain MR Image Segmentation by Relative Thresholding and Morphological Image Analysis," in *Proceedings of the First International Conference on Computer Vision Theory and Applications - VISAPP 2006*, pp. 354-364, February, 2006, Setubal, Portugal.

K. Li, A. D. Malony, R. Bell, and S. Shende, "A Framework for On-line Performance Analysis and Visualization of Large-Scale Parallel Applications," in *Proceedings of the Fifth International Conference on Parallel Processing and Applied Mathematics - PPAM 2003*, September 7-10, 2003, Czestochowa, Poland.

A. D. Malony, S. Shende, R. Bell, K. Li, L. Li, and N. Trebon, "Advances in the TAU Performance System," Chapter, "Performance Analysis and Grid Computing," (Eds. V. Getov, M. Gerndt, A. Hoisie, A. Malony, B. Miller), Kluwer, Norwell, MA, pp. 129-144, 2003.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Dr. Allen Malony, whose enlightening mentoring and critical eye were instrumental and inspiring. I wish to acknowledge Dr. Don Tucker, whose knowledge on the needs and ideas of the neuroscience community was really helpful during my Ph.D. research. I also thank the members of my dissertation committee Dr. Kent Stevens and Dr. Dejing Dou for their invaluable suggestions and feedback.

I thank my wife Li Li. Without here, I can fulfill nothing.

To my wife Li Li.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
1.1 Problem Statement	1
1.2 Overview Of The Methodology	3
1.3 Overview Of The Dissertation	5
1.4 Contributions Of The Dissertation	6
1.5 Organization Of The Dissertation	8
II. NEUROANATOMICAL SEGMENTATION IN MRI	9
2.1 Clustering	9
2.1.1 K-means Clustering And Fuzzy C-means Clustering	10
2.1.2 Finite Mixture Resolving	11
2.1.3 Clustering With Artificial Neural Networks	12
2.2 Markov Random Field	13
2.2.1 Markov Random Field Theory	13
2.2.2 Hidden Markov Random Field Model	14
2.2.3 Image Segmentation With Hidden Markov Random Field Model	15
2.3 Atlas-Based Segmentation	16
2.4 Edge Detection Based Segmentation	18
2.4.1 Edge Detection	19
2.4.2 Edge Relaxation	20
2.4.3 Heuristic Graph Search	21
2.4.4 3D Boundary Following	21
2.4.5 MR Brain Segmentation By Edge Detection	22
2.4.6 Integrating Edge Detection And Region Growing	23
2.4.7 Pros And Cons	25
2.5 Deformable Models	25
2.5.1 Snakes	26
2.5.2 Level set based deformable models	27
2.5.3 Image Influences	29
2.6 Structural And Geometric <i>A Priori</i> Knowledge	30
2.7 Intensity Inhomogeneity Correction	32

Chapter	Page
2.7.1 Adaptive Fuzzy C-Means	33
2.7.2 Adaptive Expectation-Maximization	34
2.8 Brain Extraction	34
III. RELATIVE THRESHOLDING	37
3.1 Structure Modeling	37
3.2 Image Modeling	38
3.3 Applying Relative Thresholds	42
3.4 Finding Optimal Thresholds	44
3.5 Results	45
IV. DIGITAL POINT CLASSIFICATION	48
4.1 Digital Topology	48
4.2 Point Classification	51
4.3 Related Work	54
V. 3D SKELETONIZATION	56
5.1 Thick-surface Skeletonization	57
5.2 Thin-surface Skeletonization	58
5.3 Curve Skeletonization	61
5.4 Related Work	64
VI. CELL COMPLEX BASED MORPHOMETRIC ANALYSIS	67
6.1 Motivation	67
6.2 Cell Complex	68
6.3 Cell Complex Construction	69
6.4 Generating 2-complex	70
6.5 Generating 1-complex	71
VII. TOPOLOGY CORRECTION	74

Chapter	Page
7.1 Topology Correction Preserving Surface-likeness	74
7.2 Three-fold Topology Correction	75
7.3 Multiscale Topology Correction	76
7.4 Morphological Topology Correction (MTC) Filter	77
7.5 Geodesic Dilation With Topology Control	78
7.6 Related Work	81
VIII. SEGMENTATION PIPELINE	85
8.1 Relative Thresholding	85
8.2 Terrain Analysis	87
8.3 Topology Correction Of White Matter	89
8.4 Cerebral White Matter Segmentation	91
8.4.1 Locating Cerebral White Matter	93
8.4.2 Reconstruction Of Thin-surface Skeleton Of Cerebral White Matter	97
8.4.3 Reconstruction Of Cerebral White Matter	99
8.5 Cerebral Gray Matter Segmentation	100
8.6 Cortical Surface Extraction	103
IX. EVALUATION	106
9.1 Comparative Packages	106
9.1.1 Freesurfer	106
9.1.2 SPM5	107
9.1.3 FSL	107
9.1.4 BrainVisa	108
9.2 Data Sets	108
9.2.1 BrainWeb Data Sets	108
9.2.2 IBSR Data Sets	109
9.2.3 Pathological Data Sets	110
9.3 Quantitative Evaluation	110
9.3.1 Dice Metric	111
9.3.2 Experiments	112
9.3.3 Comparison	116
9.4 Qualitative Evaluation	126
9.4.1 Segmentation Functionalities	127

Chapter	Page
9.4.2 Segmentation Automaticity And User Intervention	127
9.4.3 Segmentation Abnormalities	129
X. CONCLUSION	137
10.1 Summary Of The Segmentation Pipeline	137
10.2 Segmentation Performance	137
10.3 Contributions	138
10.4 Future Work	140
BIBLIOGRAPHY	142

LIST OF FIGURES

Figure	Page
3.1 Effect of Gaussian smoothing of a 1D signal	42
3.2 Analysis of relative thresholding with different relative thresholds applied on a phantom image.	46
3.3 Relative thresholding results on real MR scans	47
4.1 3D digital point types.	54
4.2 3D digital point classification into a tree structure.	55
5.1 Surface skeletonization results	62
5.2 Curve skeletonization results	65
5.3 Partial curve skeletonization results.	66
7.1 Surface-likeness of cerebral white matter	75
7.2 Topology correction adapted to the surface-likeness of objects	75
7.3 An example of residual points.	79
7.4 Illustration of all stages of topology correction	80
8.1 Relative thresholding result	87
8.2 Terrain analysis result	88
8.3 Tunnel filling in topology correction	91
8.4 Brainstem breaking in topology correction	91
8.5 Handle elimination in topology correction	92
8.6 Cerebral WM segmentation	94
8.7 Maximum hemisphere paths and corpus callosum path	95
8.8 Cerebral GM segmentation.	102
8.9 Reconstruction of outer cortical surface from cerebral GM segmentation.	105
9.1 Cerebral cortex Dice metrics of SPM5 and TAS on the IBSR datasets.	118
9.2 Cerebral WM Dice metrics of SPM5 and TAS on the IBSR datasets.	119
9.3 Cerebral cortex Dice metrics of Freesurfer and TAS on the IBSR datasets.	119
9.4 Cerebral WM metrics of Freesurfer and TAS on the IBSR datasets.	119
9.5 Cerebral cortex Dice metrics of FSL and TAS on the IBSR datasets.	120
9.6 Cerebral WM Dice metrics of FSL and TAS on the IBSR datasets.	120
9.7 Cerebral cortex Dice metrics of BrainVisa and TAS on the IBSR datasets.	120
9.8 Cerebral WM metrics of BrainVisa and TAS on the IBSR datasets.	121
9.9 Mean Dice metrics of five tools on the IBSR datasets	121
9.10 Mean Dice metrics of five tools on the 13 good IBSR datasets	121
9.11 Standard deviations of Dice metrics of five tools on the IBSR datasets	123
9.12 Standard deviations of of Dice metrics of five tools on the 13 good IBSR datasets	123

Figure	Page
9.13 Cerebral GM Dice metrics of five tools on the BrainWeb datasets.	125
9.14 Cerebral WM Dice metrics of five tools on the BrainWeb datasets.	126
9.15 GM-shrinking phenomenon.	131
9.16 Segmentation differences between segmentation of TAS and other four tools . .	132
9.17 Poor brain masks	133
9.18 Freesurfer abnormalities in the pathological data sets and comparison with TAS	134
9.19 Freesurfer segmentation on the BrainWeb MRI with comparison to TAS and FSL	135
9.20 TAS segmentation missing part of amigdala on the BrainWeb MRI with com- parison to Freesurfer and FSL	136

LIST OF TABLES

Table	Page
9.1 Noise levels and IHH levels of the BrainWeb data set	109
9.2 Dice metrics collected for Freesurfer over IBSR data sets	114
9.3 Dice metrics collected for FSL using different brain masks	114
9.4 SPM5 experiments with different bias regularization	115
9.5 Dice metrics of five tools with respect to cerebral cortex over the IBDR data sets	117
9.6 Dice metrics of five tools with respect to cerebral WM over the IBDR data sets	118
9.7 Standard deviation of Dice metrics of five tools over the IBSR data sets	122
9.8 Dice metrics of five tools with respect to cerebral GM over the BrainWeb data sets	124
9.9 Dice metrics of five tools with respect to cerebral WM over the BrainWeb data sets	124
9.10 Computation times of five packages on the IBSR and the BrainWeb datasets	126

CHAPTER I

Introduction

1.1 Problem Statement

Image segmentation is the problem of extracting the figure (object of interest) from an arbitrary image [1]. It can also be defined as the subdivision of an image into its constituent parts or objects [2]. By *neuroanatomical segmentation*, we mean the extraction of a description of particular neuroanatomical structures of interest that reflects the morphometry (shape measurements) of the subjects's neuroanatomy [3]. Neuroanatomical segmentation is usually conducted on T1-weighted magnetic resonance images (MRI) due to their relatively good image quality in tissue contrast and signal-to-noise ratio. Neuroanatomical structures of interest include cerebral cortex, cerebral white matter, brainstem, ventricular systems, cerebellum, and so on. The morphometric description usually takes the form of a surface representation. For example, we can represent the cerebral cortex by the inner cortical surface (the interface between the cortex and the white matter), the outer cortical surface (the interface between the cortex and the cerebrospinal fluid), or the middle cortical surface running midway through the cortical thickness. Morphometric analysis of neuroanatomical structures in conjunction with neuropsychological, neurological, and psychiatric observations, and coupled with functional neuroimaging, has found broad applications such as precisely locating active brain regions in functional neuroimaging studies, planning treatments for brain damage, and various neuropathological studies [3].

Magnetic resonance imaging (MRI) is a non-invasive method for rendering images of the inside of an object mainly based on the relaxation properties of excited hydrogen nuclei in water and lipids. Each image element (i.e. *voxel* in 3D image volumes) has a brightness, commonly referred to as intensity, that corresponds to a measurement of the tissue weighted by certain MR parameters averaged over a small 3D region. In T1-weighted MRI, spin-lattice relaxation time is selected as the image acquisition parameter. Other MRI techniques include T2-weighted and Proton-density(PD), where spin-spin relaxation time and no relaxation time is used respectively in image acquisition.

Neuroanatomical segmentation should be precise, robust, efficient and automatic in order to be practically applied for morphometric analysis [3]. Most existing neuroanatomical segmentation methods classify image voxels as belonging to three types: gray matter, white matter, and cerebrospinal fluid. *Precise* neuroanatomical segmentation requires high geometrical accuracy and topology correctness of neuroanatomical structures. It also requires morphometric description at a finer level. For example, the precise morphometric description of cerebral cortex requires further segmentation of the gray matter into the cerebral cortex and the telencephalic nuclei. The cortical surfaces should be topologically equivalent to a sphere if the opening at the brainstem is artificially closed. By *robust*, we mean that the neuroanatomical segmentation should be able to produce acceptable results for a variety of subjects and for an appropriate range of the quality of available data . Furthermore, neuroanatomical segmentation should be *efficient* and *automatic* such that the computation can be accomplished in a reasonable amount of time and requires no or limited user intervention.

Despite vigorous research for many years, precise, robust, efficient and automatic neuroanatomical segmentation remains as an unsolved problem. There are several challenges that make the problem difficult.

1. The exact intensity a given location is determined not only by the tissue type at the location but also by the neighboring tissues. This may produce an effect of blurring borders of different tissues.
2. Spatial inhomogeneities in the radio frequency (RF) gain in the RF coil [4] lead to *intensity homogeneities* (IIH), or *bias field*, in the single tissue. The presence of IIH

as a shading effect over the image causes significant overlaps between histograms of different tissues.

3. Mainly due to insufficient resolution, the *partial volume* effect [4] when the volume sampled by a single voxel contains more than one kind of tissue type blurs the tissue border and very thin structures.
4. There are significant variations in the intrinsic tissue parameters. For example, frontal cortex has been found to have an average T_1 that is 20% longer than that found in motor and somatosensory cortex [5]. It has been reported that different regions of white matter also have significantly different T_1 properties [6].
5. There are normal anatomical variations among different subjects [7]. This inter-subject variation plus the highly convoluted shape of the cerebral cortex brings further difficulties, particularly to those model and template based methods
6. Noise is inevitable in MRI, as in almost all image acquisition methods.
7. There may be other image artifacts such as motion, blood flow, echo, and so on.

As a summary, the main difficulty in neuroanatomical segmentation come from intensity variations in a single tissue, the complex anatomical structure, and the inter-subject anatomical variations.

1.2 Overview Of The Methodology

There are three key components in the work flow of most existing neuroanatomical MR image segmentation tools: 1) a brain extraction component that generates a brain mask for subsequent brain tissue classification, 2) a bias field correction component for intensity inhomogeneity elimination such that the subsequent tissue classification is simplified, and 3) a brain tissue classification component that recognizes the tissue type for each voxel in the brain. Brain tissue classification is commonly performed with statistical modeling on

the image intensities and is sometimes combined with certain *a priori* knowledge such as tissue probability maps where each location in a standard brain space is given a probability for each tissue being present. In different segmentation methods, these components may be conducted sequentially or in an iterative loop that alternates running the three components until a conversion point is reached.

The present work on neuroanatomical MR image segmentation differs from the traditional workflow in several aspects and attempts to overcome the relevant segmentation problems or limitations. First, bias field correction, no matter if it is performed prior to or simultaneously with tissue classification, is a procedure that attempts to explicitly reconstruct the image without bias field. However, there has been no guarantee for ideal correction under all circumstances [8]. The present method, on the other hand, performs robust segmentation against IHH without explicit bias field correction. This is due to a new tissue classification algorithm referred to as *relative thresholding*, which regards IHH as transparent in the segmentation.

Second, good brain extraction is a prerequisite for brain tissue classification in most segmentation work flows. However, brain extraction itself is a difficult problem and poor brain extraction usually leads to poor brain tissue classification. In the present work, relative thresholding for brain tissue classification is independent of prior brain extraction. Brain extraction follows as a procedure that eliminates false positives of the relative thresholding result. This new perspective of brain extraction based on tissue classification exploits morphology properties of the brain structures and is inherently more accurate and more robust than traditional brain extraction approach.

Third, one of the major difficulties in neuroanatomical segmentation involves the significant amount of intensity variations within a single tissue. Our initial tissue classification algorithm, relative thresholding, is based on an image modeling that is formulated as spatial constraints on intensities of different voxels instead of traditional statistical distributions such that it allows reasonable yet high extent of intensity variations for a given tissue.

Exploiting various *a priori* knowledge is the essential methodology in our approach to the neuroanatomical MR image segmentation problem. We have exploited structural, geometrical, and morphological *a priori* knowledge with respect to neuroanatomy as well as radiological properties with respect to MR imaging. These *a priori* knowledge are invariant

across different subjects and robust against various MR imaging parameters; Interestingly, in contrast, the *a priori* knowledge used in traditional methods such as the tissue probability maps, may cause an over-regularization problem where the segmentation may not fully adapt to inter-subject variations.

1.3 Overview Of The Dissertation

Neuroanatomical segmentation presented in this dissertation is performed by an image analysis pipeline that steps through five major procedures as follows.

1. The original MR image is processed by an original *relative thresholding* procedure and an original *terrain analysis* procedure such that all voxels are classified into one of the three types: white matter (WM), gray matter (GM), and background.
2. The topology defects of the WM are eliminated by an original *multiscale morphological topology correction* algorithm.
3. Cerebral WM is extracted from its superset with an original procedure called *cell-complex-based morphometric analysis*.
4. Cerebral GM is extracted based on the prior cerebral WM extraction with a set of morphological image analysis procedures.
5. Cortical surfaces are finally reconstructed preserving correct topology with an existing marching cube isosurface algorithm.

Note that step 2 through 4 can be seen as a series segmentation error correction procedures after initial brain tissue classification with relative thresholding and terrain analysis. Step 3 and 4 together can be taken as a cerebrum extraction procedure following brain tissue classification and white matter topology correction.

In this dissertation, we will evaluate our neuroanatomical segmentation tool both quantitatively and qualitatively on simulated and real MR images with groundtruth and manual

segmentation respectively, and compare the results of our tool with leading brain segmentation tools (Freesurfer, SPM5, FSL and BrainVisa) using metrics of accuracy, automation, robustness, and computational efficiency.

1.4 Contributions Of The Dissertation

In this dissertation, we proposed a new work flow for neuroanatomical MR image segmentation in which brain tissue classification is conducted prior to brain extraction and is independent of explicit bias field correction, designed and implemented a set of original algorithms that were applied in different stages in the work flow, and demonstrated by comparative evaluation that our method is highly accurate, robust, automatic and computationally efficient.

The major original algorithms presented in this dissertation include a *relative thresholding* algorithm for initial brain tissue classification, a *multiscale morphological topology correction* algorithm for topology correction of white matter, a *cell-complex-based morphometric analysis* algorithm and a *3D curve skeletonization* algorithm.

The relative thresholding algorithm is based on a new structure modeling of neuroanatomy and a new image modeling of the T1-weighted MR images exploiting various structural, geometrical and radiological *a priori* knowledge. Brain tissue classification with relative thresholding is free from three typical problems that occur in traditional intensity based segmentation methods. First, it is independent of prior brain extraction and thus avoids performance instabilities caused by poor brain extraction in many traditional methods. Second, relative thresholding is robust against intensity inhomogeneities without explicit bias field correction. Third, relative thresholding is also able to adapt to large intensity variations within a given brain tissue and thus tends to produce more accurate segmentation. On the other hand, relative thresholding can be seen as a special edge (or intensity difference) based segmentation method that overcomes several critical disadvantages of edge based segmentation approaches. First, it produces coherent regions labeled with brain tissue types. Second, it is able to recognize blurred edges and tissue boundaries where

intensities vary smoothly. Third, it is able to suppress spurious edges between voxels of same tissue types. In these respects, we see relative thresholding as a fusion of intensity based segmentation and edge based segmentation. The idea can be effective in other image segmentation problems, particularly where there are intensity inhomogeneities and blurred edges.

The cell complex based morphometric analysis simplifies a 3D object into a 1D structure and gives a quantitative measurement on the wideness and connectivity on every location in the 3D object. This is a significant advancement over the fact that traditionally only a “thickness” metric (i.e. the distance to the boundary) can be calculated for each point in the 3D object. This new 3D morphometric instrument will potentially promote more applications of morphological analysis for various problems in computer vision and image understanding. By applying this new morphometric analysis on the white matter generated, we are able to eliminate non-brain tissues and divide the entire white matter at the brain stem based on the *a priori* knowledge of strong connectivity of cerebral white matter. Cerebrum extraction using cell-complex-based morphometric analysis provides higher robustness than other brain extraction such as traditional morphological image analysis, deformable model based methods, and atlas based methods.

The white matter topology correction algorithm is based on the *a priori* observation that human white matter, particularly cerebral white matter, is a surface-like object. Preservation of this morphological property is taken as the major criterion for eliminating topology defects. In addition, our topology correction algorithm involves WM, GM and background in the procedure, in contrast to the traditional procedure where only the foreground and background are involved. Our three-fold procedure exploiting the surface-likeness morphological *a priori* knowledge tends to more robustly produce reasonable solutions to topology defect elimination than other methods.

The 3D curve skeletonization algorithm is performed directly on a 3D object in contrast to the traditional methods that depend on prior surface skeletonization and tend to generate skeletons with better “medialness”. A variant of our curve skeletonization is referred to as “shape and topology preserving erosion” in which the skeletonization procedure is conducted in certain iterations instead of until conversion. This variant algorithm is used as an important component in the white matter topology correction algorithm as well as for

generating topology correct gray matter in our neuroanatomical pipeline. A similar procedure can also be used in 3D object smoothing to eliminate noisy protrusions on the 3D object. This framework of 3D curve skeletonization is based on a systematic point classification of discrete 3D objects. In this classification approach, we proposed the central notion of a *thick-simple point*. This notion enables deeper and wider topology and geometry characterization of any points in a 3D digital object.

1.5 Organization Of The Dissertation

The segmentation pipeline is described step by step in chapter VIII, but the key algorithms are separately presented in previous chapters. Chapter III describes the relative thresholding algorithm (summarized in section 8.1). Terrain analysis as a complementary technique to relative thresholding for brain tissue classification is described in section 8.2. The multiscale morphological topology correction of WM is described in chapter VII (summarized in section 8.3). Cell complex based morphometric analysis is presented in chapter 6 and its application for cerebral WM extraction is discussed in section 8.4. Cerebral GM extraction is described in section 8.5 and cortical surface reconstruction is presented in section 8.6. The segmentation pipeline also depends on several 3D skeletonization routines as described in chapter V. Chapter IV is a set of definitions on characterization of different points in 3D discrete object, which forms the basis for the 3D skeletonization and the topology correction algorithms. The comparative evaluation of the segmentation pipeline is presented in chapter IX. Chapter II of the dissertation contains a survey on the existing neuroanatomical MR image segmentation methods.

CHAPTER II

Neuroanatomical Segmentation In MRI

In this chapter we give a survey on the existing neuroanatomical segmentation methods. There are mainly two types of segmentation methods: intensity-based methods and edge-based methods. Intensity-based methods classify image voxels based on the voxel intensities while edge-based methods extract anatomical contours based on a gradient or edge map of the original image. A variety of widely applied intensity-based automatic segmentation methods are also often referred to as clustering methods. These methods, such as k-means clustering and finite mixture resolving assume there is no intensity inhomogeneities and are supposed to be performed on the brain region extracted in a prior procedure. Basic clustering methods are often extended with atlas or Markov random field to improve performance. Intensity inhomogeneity correction has been researched by extending the exiting segmentation methods or by proposing a separate preprocessing procedure. There are mainly two types of edge-based segmentation methods: those use plain edge detection or fuse edge detection with other segmentation techniques such as morphological operations and region growing, and those based on deformable models, in which a surface model is deformed such that it is attached to the salient edges while maintaining the smoothness of the model.

In the first section of this chapter, we present several basic clustering segmentation methods. In section 2 and section 3, we describe segmentation methods using Markov random field and brain atlas respectively that extend the basic intensity-based segmentation methods. Edge-detection based segmentation methods are reviewed in section 4 and segmentation based on deformable models is described in section 5. Intensity inhomogeneity correction and brain extraction methods are reviewed in section 6 and section 7 respectively.

2.1 Clustering

Clustering can be loosely defined as the unsupervised classification that groups similar objects from a collection of unlabeled data, in which each object is associated with a vector of feature values. The similarity is determined based on the feature vectors, which could be simply the pixel intensities in image segmentation. There are two basic clustering methods for image segmentation: partitional clustering and hierarchical clustering. Partitional clustering generates a flat group structure in that all data groups form disjoint cluster sets. Hierarchical clustering generates pattern groups in a hierarchical structure (i.e. dendrogram) which can be cut at a dissimilarity level forming a partition. In neuroanatomical MR image segmentation, many methods either extend traditional partitional clustering and use them to generate initial results for further processing. This section describes some common partitional clustering methods used in neuroanatomical MR image segmentation.

2.1.1 K-means Clustering And Fuzzy C-means Clustering

k -means clustering and fuzzy c -means clustering are two widely used clustering methods. k -means clustering partitions samples with the sum-of-squared-error criterion. Starting from an initial partition, it iteratively classifies all samples into k clusters according to nearest mean and then recompute the means in each iteration. The iteration stops when there is no change in the means.

Fuzzy c -means clustering is an extension of the k -means clustering with fuzzy set theory. We assume that each sample y_j has some graded or “fuzzy” membership in a cluster. At root, the “membership” μ_{ij} is equivalent to the posterior probability $P(\omega_i|y_j)$. The fuzzy c -means clustering seeks a minimum of a heuristic global criterion function $J_{fuz} = \sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^b \|y_j - m_i\|^2$, where b is a parameter chosen to adjust the “blending” of different clusters. It can be shown that when J_{fuz} arrives at a minimum, $m_i = \frac{\sum_{j=1}^n (\mu_{ij})^b y_j}{\sum_{j=1}^n (\mu_{ij})^b}$ and $\mu_{ij} = \frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^c (1/d_{rj})^{1/(b-1)}}$, where $d_{ij} = \|y_j - m_i\|^2$. The fuzzy c -means clustering al-

gorithm proceeds iteratively recomputing m_j and μ_{ij} according to the above two equations until they reach stable values.

K-means [9] and fuzzy c-means [10, 11, 12] clustering are commonly applied for image segmentation. The main issue of the two methods is to determine the feature composition of each pattern. Generally speaking, the measurements/features could be point multispectral values, point color components and derived color components, or derived statistics such as mean, standard deviation, and modes, in a certain neighborhood of the pixel [13].

It has been shown that the k -means algorithm converges to a locally optimal solution. Generally the fuzzy c -means algorithm is better than the hard k -means algorithm, but it may still converge to local minima of the squared error criterion [14]. Note that little or no spatial information is considered in the feature space based clustering methods. Usually only the spatial coordinates of a pixel are considered as additional features. This tends to result in poor segmentation, particularly when the desired regions do not form simple regions with similar coordinates.

2.1.2 Finite Mixture Resolving

This is a parametric clustering method in that the intensity levels of pixels in the image are assumed to be a mixture of finite number of certain probability distributions, usually Gaussian, with a parameter vector θ . Formally, let $\omega = \{\omega_i, 1 \leq i \leq N\}$ be the pixel types and $P(y_j|\omega_i, \theta_{\omega_i})$ be the probability of the intensity level y_j of pixel j conditional on the pixel class ω_i and the given parameters θ_{ω_i} . Then the marginal probability of the intensity level y_j over all labels is $P(y_j|\theta) = \sum_i P(y_j|\omega_i, \theta_{\omega_i})P(\omega_i)$. After the parameters are estimated, usually by the Expectation-maximization (EM) method [15], the region type of a pixel is determined by maximization of the posterior probability $P(\omega_i|y_j)$. A commonly used parameter estimation method is to take the labels as hidden or missing data and to use the expectation-maximization method to maximize the likelihood $P(y, \omega|\theta)$.

The expectation-maximization (EM) algorithm is often used to estimate the parameters of the probability distribution $P(y, \omega|\theta)$ that models some incomplete data, where y denotes the observed variables and ω denotes the missing or hidden variables. Note that here y is a vector representing all samples $(y_1, y_2, \dots, y_n)^T$. Then the log-likelihood of the observed

variables $l(\theta) = \log P(y|\theta) = \log \sum_{\omega} P(y, \omega|\theta)$. The EM algorithm attempts to find the values of θ such that the likelihood $l(\theta)$ is maximized. The EM algorithm proceeds in rounds starting from an initial guess of the parameters θ . In each round, let the θ^t be the current parameter setting, we want to find a new parameter setting, θ^{t+1} that increase the log-likelihood. When $l(\theta)$ converges, we obtained an local maximum value of the log-likelihood $l(\theta)$.

In each round, the EM algorithm finds a lower bound $B(\theta|\theta^t)$ of $l(\theta)$ that touches $l(\theta)$ at θ^t (i.e. $\forall \theta, B(\theta|\theta^t) \leq l(\theta)$, and $L(\theta^t|\theta^t) = l(\theta^t)$). Intuitively, when we locally maximize the bound with respect to θ in each iteration, it will guarantee that we obtain an improved estimate θ^{t+1} . The bound can be derived by Jensen's inequality as follows:

$$l(\theta) = \log \sum_{\omega} P(y, \omega|\theta) = \log \sum_{\omega} P(\omega|y, \theta^t) \frac{P(y, \omega|\theta)}{P(\omega|y, \theta^t)} \geq \sum_{\omega} P(\omega|y, \theta^t) \log \frac{P(y, \omega|\theta)}{P(\omega|y, \theta^t)}$$

$$= B(\theta|\theta^t). \text{ Since } P(\omega|y, \theta^t) = \frac{P(y, \omega|\theta^t)}{\sum_{\omega'} P(y, \omega'|\theta^t)} = \frac{P(y, \omega|\theta^t)}{P(y|\theta^t)}, \text{ we can show that } B(\theta^t|\theta^t)$$

$$= \sum_{\omega} P(\omega|y, \theta^t) \log \frac{P(y, \omega|\theta^t)}{P(y, \omega|\theta^t)/P(y|\theta^t)} = \log P(y|\theta^t) \sum_{\omega} P(\omega|y, \theta^t) = \log P(y|\theta^t) = l(\theta^t).$$
 Note that maximizing $B(\theta|\theta^t)$ with respect to θ does not involve the nominator of the log term. Therefore maximizing $B(\theta|\theta^t)$ is equivalent to maximizing

$$Q(\theta|\theta^t) = \sum_{\omega} P(\omega|y, \theta^t) \log P(y, \omega|\theta).$$

An interpretation of Q is that it calculates the expectation $E_{\omega}[P(y, \omega|\theta)]$ of the likelihood of the complete data over the hidden data space (hence the name of the algorithm).

Note that only statistical information are considered in the segmentation with mixture resolving and this often results in poor segmentation. The finite mixture model can be extended with the Markov random field model and probability tissue maps to incorporate spatial information into the segmentation and even bias field correction, as described later in this chapter.

2.1.3 Clustering With Artificial Neural Networks

Clustering methods based on artificial neural networks have also been applied in medical image segmentation. Many of these methods attempt to resolve the standard finite

normal mixture model based on competitive learning [16, 17]. Some others incorporate spatial dependence of labels with statistical neural networks [18, 19]. One of the critical drawbacks of these methods is that most of them can only be applied to small data sets because obtaining suitable learning/control parameters for the network is difficult and their execution times are very high for large data sets [14]. For brain MR image segmentation, only results for individual slices were shown. The only network that have been applied to large data set is the Kohonen net and the results are equivalent to the k -means algorithm [20].

2.2 Markov Random Field

2.2.1 Markov Random Field Theory

In probability theory, a *stochastic process* is a random function defined over a time interval or a region of space. In the former case, the stochastic process is called a *time series*; in the latter case, it is called a *random field*. Mathematically, a stochastic process is usually defined as an indexed collection of random variables X_i with index i running over an index set $S = \{1, 2, \dots, N\}$ and with the values ω_i of the variables chosen from range R . For random fields, S represents a set of sites in space and particularly the coordinates of pixels for an image. A configuration $\omega = \{\omega_i \in R, i \in S\}$ of a random field X is the set of realization values of the random variables in X . Let Ω be the all possible configurations so that $\Omega = \{\omega = (\omega_1, \dots, \omega_N) | \omega_i \in R, i \in S\}$.

For a random field x defined on the sites S , define a *neighborhood system* $\mathcal{N} = \{\mathcal{N}_i, i \in S\}$, where \mathcal{N}_i is the set of sites neighboring i , $i \notin \mathcal{N}_i$ and $i \in \mathcal{N}_j \Leftrightarrow j \in \mathcal{N}_i$. A random field X is said to be a *Markov Random Field* (MRF) on S with respect to a neighborhood system \mathcal{N} if and only if for $\forall \omega \in \Omega$, $P(\omega) > 0$ and $P(\omega_i | \omega_{S-\{i\}}) = P(\omega_i | \omega_{\mathcal{N}_i})$. The last condition is referred to as the local characteristic of MRF. From the local characteristic of MRF, we see that it is a natural facility to model the spatial dependence of region/tissue types in an image.

A MRF is usually constructed with a *Gibbs distribution*. Before the definition of the Gibbs distribution, we remark that the site set S together with the neighborhood system \mathcal{N} define a graph in the usual way. Let C be all the cliques in the graph represented by $G = \{S, \mathcal{N}\}$. A clique in a graph G is subgraph of G and is a complete graph by itself. A Gibbs distribution with respect to $G = \{S, \mathcal{N}\}$ is a probability measure π on Ω with the form $\pi(\omega) = \frac{1}{Z} e^{-U(\omega)/T}$, where Z and T are constants and the *energy function* U is of the form $U(\omega) = \sum_{c \in C} V_c(\omega)$. Each V_c is a potential function on Ω with the property that $V_c(\omega)$ depends only on those variables on the sites $s \in c$. Z is the normalizing constant: $Z = \sum_{\omega} e^{-U(\omega)/T}$ and is called the *partition function*. The equivalence between MRF and Gibbs distribution states that given a site set S and a neighborhood system \mathcal{N} , X is an MRF with respect to \mathcal{N} if and only if $\pi(\omega) = P(X = \omega)$ is a Gibbs distribution with respect to \mathcal{N} . For image segmentation, a typical form of V_c is as follows. If c is a clique with two neighboring sites $\{r, s\}$, then $V_c(\omega) = \beta$ in the case of $\omega_s = \omega_r$, or $V_c(\omega) = -\beta$ in the case of $\omega_s \neq \omega_r$, where β is a parameter of the model. If c is a clique with only one site s , then $V_c(\omega) = -\alpha_{\omega_s}$, where α_{ω_s} is another tissue dependent parameter. For all the other cliques in the graph, $V_c(\omega) = 0$. This model is the well-known *Ising model*.

2.2.2 Hidden Markov Random Field Model

The *hidden Markov random field* (HMRF) model for image segmentation is derived from *hidden Markov models* (HMM), in which a stochastic process is generated by a Markov chain, which can be taken as a 1D Markov random field, with hidden state sequence. In HMM, each observation is assumed to be a stochastic function of the state sequence. The underlying Markov chain changes its state according to a $l \times l$ transition probability matrix, where l is the number of states. HMMs have been successfully applied in applications such as speech recognition and handwritten script recognition.

The hidden Markov random field model consists of a hidden Markov random field $X = \{X_i, i \in S\}$ with its values in a finite state space R with probability $P(X = \omega) = \pi(\omega)$ and an observable random field $Y = \{Y_i, i \in S\}$ with its value in a finite state space I . For image segmentation, each hidden state represents a region type and each observable state represent an intensity level. Given any particular configuration $\omega \in \Omega$

of the random field X , every observed state y_i follows a certain conditional probability distribution $P(y_i|\omega_i)$. This distribution is called the *emission probability function* and Y is also called the *emitted random field*. In HMRF, the observations y are conditionally independent given ω : $P(y|\omega) = \prod_i P(y_i|\omega_i)$.

Hidden Markov random field model have been applied for both supervised [21] and unsupervised [22, 23, 24, 25] image segmentation. For supervised segmentation, the emission probability function can be estimated with nonparametric methods such as the Parzen-window method given a training set. For unsupervised segmentation, they usually assume the same functional form $f(y_i; \theta_{\omega_i})$, where θ_{ω_i} is the involved parameters. Gaussian distribution is a typical choice for the emission probability function. We use θ to denote all the parameters involved in all emission probability functions and those involved in the prior distribution $P(\omega)$ (the latter are usually assumed known in prior).

2.2.3 Image Segmentation With Hidden Markov Random Field Model

With the hidden Markov random field model defined above, supervised image segmentation can be posed as an optimization problem of finding the optimal estimates of ω^* by maximization of the posterior probability $P(\omega|y) \propto P(y|\omega)P(\omega)$; for unsupervised image segmentation, the involved parameters θ^* as well as the optimal ω^* are estimated concurrently by maximization of the posterior probability $P(\omega, \theta|y) \propto P(y|\omega, \theta)P(\omega)$ given the observed intensity y . However, direct solution of the problem is both analytically and computationally intractable due to the exponential complexity of Ω and the multimodal (i.e. multiple local minima) nature of the posterior distribution. Most practical solutions are performed by iterations of segmentation steps, each which only updates the configuration of ω at local sites. For unsupervised segmentation, the segmentation is interrupted by a step of parameter estimation at regular intervals. The segmentation step finds the optimal solution of ω given the current estimation of the parameters θ , while the parameter estimation step finds the optimal estimation of the parameters θ given the current segmentation ω .

Simulated annealing (SA) [26] and iterated conditional mode (ICM) [27] are two commonly used methods in the segmentation step. The SA algorithm scans all sites (pixels), randomly drawing a tentative region type ω_s for each site s . If the selection of ω_s increases

the posterior probability conditional on the labels of s ' neighbors $\omega_{\mathcal{N}_s}$, $P(\omega_s|\omega_{\mathcal{N}_s}, y)$ in supervised case or $P(\omega_s|\omega_{\mathcal{N}_s}, y, \theta)$ in unsupervised case, then ω_s is chosen for site s , otherwise, it is chosen according to a certain probability based on a temperature parameter T , which is decreased according to a certain schedule as the algorithm proceeds. The ICM algorithm is deterministic and can be taken as an extreme case of the SA algorithm with the temperature parameter always being zero so that in each site s , ω_s is chosen by local maximization of the conditional posterior probability. The SA algorithm provides better approximation of the optimal segmentation, but it is very slow. The ICA algorithm is fast, but may be trapped into a local minimum. A typical example application of HMRF model in neuroanatomical segmentation is in FSL [28] in which HMDF model is used to enforce spatial regularization in order to improve the segmentation robustness against image noise. It is also demonstrated in FSL that implementation of HMRF using ICM can be integrated with the finite mixture resolving and bias field correction in the iterative Expectation-Maximization method. Other work on using MRF for brain MR image segmentation are [29, 30, 31, 21, 22, 23].

The commonly used Ising-like model was criticized for that it tends to minimize the boundary length between tissues [32], which discourages classifications from accurately following the highly convoluted shape of the complex human cortex [33]. This effect is particularly amplified in brain images where the presence of large uniform regions of single tissue types results in high estimates of the transition parameter β and strong favor for smooth boundaries. As a possible solution and research direction, it was suggested in [33] to use a nonstationary Ising model with different parameters in uniform regions of pure tissue from those used at places where tissues mix.

2.3 Atlas-Based Segmentation

The main idea here is to use a template of the target object to find an ideal match between the template and the image. The template in the case of neuroimaging is usually referred to as a brain atlas and the relevant segmentation methods are called atlas-guided seg-

mentation methods. A brain atlas is a detailed representation of a single subject's anatomy (i.e. anatomical atlas) in a standardized three-dimensional coordinate system [34] or probability tissue maps (i.e. probabilistic atlas) learned from a training set. The most commonly used coordinate system is the Talairach reference system [35], in which the anterior commissure is the origin and the plane containing the line connecting the anterior commissure and the posterior commissure, perpendicular to the sagittal midplane, is taken as the horizontal plane. While a probabilistic atlas is usually used together with traditional clustering methods, an anatomical atlas is matched to a new scan in a procedure called brain warping so that any information in the atlas including the tissue types are transferred into the new scan. Because of the complex structural variability of brains between individuals, it is generally impossible to obtain an exact matching with rigid (translation, rotation) or linear (translation, rotation, scaling, shearing) transformations and research on brain warping has been focusing on deformable atlases, which can be adapted to the anatomy of new subjects with nonlinear transformation.

There are two brain warping approaches based on deformable atlases: volumetric warping and nonvolumetric (or model-driven) warping. Model-driven warping is an image registration method which requires a segmentation preprocessing step to obtain good performance. Key surfaces in the brain are first extracted with deformable model based methods and matched to the surfaces in the atlas [36, 37]. Volumetric warping [38, 39, 40, 41] aims to match the atlas and the target scan according to a regularized criterion. The most commonly used criterion is the sum of cross-correlation [38, 39, 40, 41] locally calculated on the intensity or/and the edge maps between the atlas and the target scan. Other criteria include the sum of squared differences [40], and mutual information which is more effective for matching images with different modalities. The brain warping is thus to find the optimal deformation field such that the criterion is maximized (or minimized), which is an ill-posed problem in that there are many possible solutions. Usually some constraints are used to regularize the solution. These constraints range from the simple maximum deformation limit [41] to the widely-used physically-based elastic model [38, 39] and the viscous model [40] that enforces topological properties on the deformation. To save time and to obtain better performance, volumetric warping usually follow a preliminary global linear transformation and a multi-resolution scheme is used to in the implementation.

In SPM5[42], probability maps for different tissues are used as spatial priors in the finite mixture model and atlas registration is performed together with finite mixture resolving and bias field correction in a circular procedure. The atlas registration is implemented with a linear combination of about a thousand cosine transform bases. In Freesurfer[43], the atlas includes not only prior probabilities for each tissue class at each atlas location, but also a Gaussian distribution of the intensities for each class at each atlas location and a neighborhood function representing the probability that a given point belongs to a label given the classification of its neighboring points based on an anisotropic nonstationary Markov random field. The atlas is first registered with the image in a linear affine transformation and then voxels are labeled with the maximum a posteriori (MAP) method and the segmentation is then sequentially updated using the iterated conditional mode (ICM) algorithm in which the a posteriori probability of a class at each point is computed as the probability that the given class appeared at that location in the training set times the likelihood of getting the subject-specific measured value from that class. The latter is computed from the PDF for that label as estimated from the training set. The probability of each class at each point is computed. An initial segmentation is generated by assigning each point to the class for which the probability is greatest. Given this segmentation, the neighborhood function is used to recompute the class probabilities in the second step. The new class probabilities are then used for resegmentation in the next round.

2.4 Edge Detection Based Segmentation

Edge detection algorithms produce a map of edge points with associated magnitudes and directions. The edge detection results are in the form of edge segments, which are usually short and disconnected. The edge-based segmentation algorithms mainly involve aggregating these short edge segments into extended edges that correspond to object boundaries, a procedure often referred to as *edge linking* or *edge following*. Edge-based segmentation methods are effective when there are good contrast between regions in the image. The most common problems are due to the presence of edges in locations where there is no

object boundary, as well as the absence of edges where a real boundary exists. In this section, we describe two well-known edge-based 2D image segmentation methods: edge relaxation as a local edge linking method and heuristic graph search as a global edge linking method. Another well-known edge linking method, usually for line segmentation, is the Hough transform [44, 45, 13], which is not presented here. Note that deformable model based segmentation methods (see section 2.5) that utilize edge formation are also generally referred to as edge-based segmentation methods. They work for 2D images as well as 3D images, and provide additional robustness against noise and spurious edges.

This section first gives an overview of edge detection methods followed by two 2D edge linking methods: edge relaxation and heuristic graph search. Next, a 3D boundary following method using 2D graph search is described. Cortical surface reconstruction based on edge detection are then described in the next two subsections followed by integration of edge detection and region-growing for improved performance. Finally, the pros and cons of the methods based on edge detection are summarized in the last subsection.

2.4.1 Edge Detection

In computer vision, edge detection is a process that attempts to capture the discontinuities in the photometrical, geometrical and physical characteristics of objects [46]. The basic method of edge detection is to first calculate the gradient at each image pixel and then threshold the gradient threshold to label edges. Gradient calculation masks, as shown below for Δ_x and Δ_y , are used for this purpose. Well-known gradient calculation masks are Prewitt's masks [47] and Sobel's masks [48],

$$\Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -a & 0 & a \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \Delta_y = \begin{bmatrix} -1 & -a & -1 \\ 0 & 0 & 0 \\ 1 & a & 1 \end{bmatrix}$$

where a is a positive real number (1 in the case of Prewitt's masks and 2 in the case of Sobel's masks). The performance of these operators deteriorates when the image is noisy. Rosenfeld and Thurston [49] proposed a smoothing operation to reduce the noise image by replacing the value of a pixel by the average computed on a squared window.

Two commonly-used edge detection algorithms are the Canny edge detection algorithm [50] and the zero-crossing edge detection algorithm [51]. In the zero-crossing method, the image is convolved with the Laplacian of the Gaussian (LOG) and the zero-crossings are labeled. For efficient computation, LOG can be approximated by the Difference of Gaussians (DOG) that subtracts a wide Gaussian from a narrow Gaussian. In the Canny's method, the image is first convolved with the first-order derivatives of the Gaussian, and then the edges are located at the maxima of the gradient modulus taken in the direction of the gradient. Canny's scheme of edge detection inspired significant research in the field. A survey of edge detection methods is given in [46].

2.4.2 Edge Relaxation

Edge relaxation is a procedure performed on the crack edges. If one thinks of a 2D image as a city map with each pixel corresponding to a block, then a crack edge is a street segment between two blocks. The edge detection algorithm provides an initial confidence for each crack edge with normalized values ranging from 0 to 1. The relaxation procedure then iteratively updates the confidence of each crack edge considering the edge properties in the context of their mutual neighbors until the confidence converges to either 0 or 1. Eventually crack edges with confidence 1 are taken as object boundaries and others are discarded. Edge relaxation can effectively improve segmentation results when region contrast is good at boundaries, but may be corrupted by noises.

A typical context of a crack edge consists of six other crack edges with three on each side of the central edge. A central edge is then classified with a pair of numbers $a-b$, where a and b represent the number of edges having greater confidence than a threshold. More sophisticated classification methods are also possible. The meaning of the types and the related rules to update the confidence are listed below:

- 0 – 0 isolated edge: negative influence on the edge confidence
- 0 – 2, 0 – 3 dead end: negative influence on the edge confidence
- 0 – 1 uncertain: weak positive, or no influence on edge confidence
- 1 – 1 continuation: strong positive influence on edge confidence

- 1 – 2, 1 – 3 continuation to boundary intersection: medium positive influence on edge confidence
- 2 – 2, 2 – 3, 3 – 3 bridge between boundaries: not necessary for segmentation, no influence on edge confidence

2.4.3 Heuristic Graph Search

In heuristic graph search based methods, a directed (oriented) weighted-graph is first constructed. In this graph, each vertex corresponds to an edge point in the edge map and they are linked with directed arcs (the term “arc” is used here to avoid the abuse of the term “edge”) according to certain heuristics. Commonly used heuristics are based on the following assumptions: 1) the edge magnitudes along the object boundaries are approximately constant; 2) the object boundaries are smooth; and 3) the edge magnitudes at object boundaries are high. With such assumptions, two vertices in the graph are linked with an arc only if their magnitude is greater than a threshold, if their magnitude difference is smaller than a second threshold, and if their edge direction difference is smaller than a third threshold.

The problem that the heuristic graph search algorithm addresses is how to determine the optimal path between two given pixels p_A and p_B such that a cost function is minimized. A typical cost function is $C = -D + \alpha E + \beta F$, where D is the sum of the magnitudes of edge points along the path, E is the sum of the difference of magnitude of adjacent edge points along the path, F is the sum of the difference of directions of adjacent edge points along the path, and α and β are two weighting parameters. Dynamic programming is usually used to implement the algorithm to search for the optimal path based on the observation that the optimal path from p_A to p_B can be split into two optimal sub-paths: from p_A to p_i and from p_i to p_B . The details of the implementation are omitted here.

2.4.4 3D Boundary Following

3D boundary following algorithms use prior edge detection results [52, 53, 54, 55]. Here, we will present a typical algorithm proposed by Cappelletti and Rosenfeld [55].

This algorithm generates a series of 2D boundaries which, when stacked, provided a 3D boundary of the object of interest. An assumption is that the object has smooth surfaces and its sections are all connected and approximately circular. A graph search based method is proposed for following 2D boundaries in each of the cross section.

The graph search algorithm finds an optimal path with lowest cost in the graph corresponding to each cross section. The cost function is a combination of the mean gradient magnitude, the circularity of the 2D boundary and the closeness of the neighboring 2D boundaries. The three cost function components are weighted with scaling factors defined as program parameters. In this work, the start node in the path is allowed to be placed slightly off the intended boundary. The path found by the algorithm contains a closed subpath that constitutes the detected boundary.

The initial cross section for 2D boundary following is chosen to pass near the center of the object. The 3D boundary following consists of a series of 3D passes to generate a 3D boundary in which 2D boundaries in adjacent cross sections are consistent such that they are “aligned” with one another. In the first 3D pass, a series of 2D boundaries are extracted independently of one another. The cost function applied in the first pass is based only on the mean gradient magnitude and boundary circularity. In subsequent passes, the cost function for 2D boundary detection in a given cross section is extended to include constraints from 2D boundaries in adjacent cross-sections.

The Cappelletti and Rosenfeld algorithm was tested on some synthetic images for extraction of object with simple and compact shape. However, cortical surface extraction with 3D boundary following is challenging for various reasons. First, the shape of the cortical surface is convoluted such that the circularity and compactness is not satisfied for many graph search algorithms for 2D edge linking. Second, in each 2D cross section, cortical surface is not necessarily connected. Third, various image artifacts may degrade the results of edge detection. For example, we consistently observed blurring edges between WM and GM in regions such as superior gyri, border between cerebrum and cerebellum, and border between temporal lobes and flesh. 3D boundary following for cortical surface reconstruction may also be disturbed by spurious edges within WM due to noise and undesired edges between tissues such as blood vessel, dura mater, fat, and flesh.

2.4.5 MR Brain Segmentation By Edge Detection

Edge detection algorithms have been used for MR brain segmentation. In [56], a two-stage edge detection scheme is used to segment brain structures in 2D MR images. First, a location within the desired region, say, white matter, is indicated. A differential intensity map is then created by calculating the absolute value of the difference of the image intensity at each pixel with respect to that at the reference location. By picking a differential value, the initial contour between the desired region and the bordering region is created. This initial contour is improved in the second edge detection step considering the edge detected by the Sobel operator. For 3D MR brain segmentation, a 2D slice is first segmented and the contour is projected into adjacent slices as an initial contour to be improved. The success of this process requires that the image slices be relatively thin and that the user evaluates the resulting 2D contour and corrects, when necessary, any errors that occur before they propagate through the data set.

MR brain extraction with edge detection described in [57] uses DOG for edge detection followed by region binarization into brain and non-brain regions. Region classification is conducted on four slices at a time instead of the whole 3D image in order to prevent local errors from corrupting the entire data set. The classification assumes that there is a large brain region in every 4 slices, whose mean intensity together with those of other regions such as fat and CSF are used to compute thresholds for classification of smaller regions. Heuristic rules as well as user interaction are applied for correction of segmentation errors. DOG edge detection followed by morphological operations is applied in [58] for segmentation of more anatomical contours in the head such as skin, bone, brain and the ventricular systems, but the labeling of these structures is done interactively. DOG edge detection together with pixel classification is also used in [59] for brain tissue classification. Here, significant amount of user interaction is also required for accurate performance of the segmentation.

2.4.6 Integrating Edge Detection And Region Growing

Region-based segmentation methods directly find coherent regions assuming the regions have homogeneous intensities [44, 45]. Unlike edge-based segmentation methods,

region-based methods are guaranteed to produce coherent regions without linking edges. However, decisions about region membership are usually more difficult than applying edge detectors. A commonly used region-based method, region growing, is described below, followed by presentation of techniques integrating region growing and edge detection for image segmentation.

The basic idea of region growing is to start from a seed point as the initial region and grow the region by iteratively adding neighboring pixels if some similarity measure between the region and the neighboring pixel is high enough (i.e. greater than a certain threshold). So region growing mainly involves the selection of the similarity measure. Two possible similarity measures compare the candidate pixel with the original seed or its neighbors in the region. However, the former is sensitive to the selection of the seed pixel and the latter causes significant drift as the region grows farther away from the original seed pixel. A compromise is to compare the candidate pixel with certain region statistics, usually the mean. By initializing the region with multiple seeds, the candidate pixel can also be compared with the mean with respect to the variance of the region. Another approach is to use the cumulative difference as one follows a path from the seed to the candidate pixel. Yet another approach is to provide not only the seeds that should be in the region but also the seeds that should not be in the region.

Three types of errors may occur in the region boundaries produced by any region growing process : a) false positive boundaries: a region boundary is not an edge and there are no edges nearby; b) false negative boundaries: there exist edges with no boundaries near them; and c) false localization: a region boundary corresponds to an edge but it does not coincide with it. By boundary we refer to border of regions and by edge we refer to low-level image feature produced by edge detection. Usually the false negative boundaries can be significantly reduced by proper selection of parameters in region growing, which results in an over-segmented image and increase of the false positive boundary errors. A boundary elimination technique and a boundary modification technique are proposed in [60] for correction of the false positive boundaries and false boundary localization integrating edge detection results. Following the method in [60], boundary elimination is performed considering the contrast along the boundary and the length of the boundary penalizing for long boundaries with low contrast in [61] for MR brain image segmentation. Boundary modifi-

cation is performed using deformable model based methods to bring boundaries to nearby edges with locally maximum contrast. The integration of region growing and edge detection may improve the results of plain region growing and plain edge detection. The method proposed in [60] and applied in [61] is conducted in 2D images. In 3D, the criterion for boundary elimination is more difficult to define.

2.4.7 Pros And Cons

The main advantage of edge detection based techniques is that it may generate accurate results when the contrast between two regions is high. Edge detection is also more robust against intensity inhomogeneity than intensity based methods and the computation is usually efficient. The weaknesses of segmentation based on edge detection are as follows[62]. First, edge gaps often occur due to variation in the gradient strengths of the tissue characteristics. For example, we consistently observed no local maximum of gradient strength between WM and GM in regions such as superior gyri, the border between cerebrum and cerebellum, and the border between temporal lobes and flesh. Second, variation in edge strength can bring discontinuities in the boundaries. For example, the strength of edge between GM and WM at many gyri areas are significantly less than those in some sulci areas. Third, spurious edge may occur due to noise and texture. Fourth, cortical surface reconstruction by stacking contours in 2D slices is highly dependent on the accuracy of the segmentation process in the 2D slices and problematic in both geometry and topology due to the highly convoluted shape of the brain, limited image resolution, and various image artifacts. In conclusion, these methods based on plain edge detection are not reliable or robust and require significant amount of user intervention for acceptable results.

2.5 Deformable Models

Medical image segmentation methods based on deformable models attempt to track anatomic structures in the image by exploiting (bottom-up) constraints derived from the image data together with (top-down) *a priori* knowledge about the location, geometry, and

shape of these structures [63]. This is a vigorously researched area and numerous methods have been proposed that vary in the representation of the model, the constraints imposed on the model, the optimal approximation methods that fit the models to the measured data, and the degree of freedom of the model. Deformable models are also called active contours by some researchers. Those models with preferably fewer degrees of freedom are usually called deformable templates or active shapes (still regarded as deformable models) will also be discussed briefly in this section.

2.5.1 Snakes

The groundbreaking work on deformable models is the concept of snakes [64]. A snake is a 2D parametric curve $\mathbf{c}(s) = (x(s), y(s))^T$ embedded in the image plane $(x, y) \in \mathbb{R}^2$, where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain. The curve is usually closed such that $x(0) = x(1)$ and $y(0) = y(1)$. An optimal snake is the one that is attached to salient image features, typically edges, and maintains internal smoothness. This is expressed by minimization of the functional $\mathcal{E}(\mathbf{c}) = \mathcal{S}(\mathbf{c}) + \mathcal{P}(\mathbf{c})$, where $\mathcal{S}(\mathbf{c}) = \int_0^1 \alpha(s) \left| \frac{\partial \mathbf{c}}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 \mathbf{c}}{\partial s^2} \right|^2 ds$, and $\mathcal{P}(\mathbf{c}) = \int_0^1 P(\mathbf{c}(s)) ds$. In physics terminology, \mathcal{E} is referred to as an energy functional which consists of an internal energy \mathcal{S} of the snake that characterizes its smoothness and an image energy \mathcal{P} which is derived from an external image constraint that pushes the snake toward salient image features. The first-order term in \mathcal{S} makes the snake act like a membrane and the second-order term makes it act like a thin plate. $P(\mathbf{c})$ denotes a scalar potential function on the image plane and is typically defined as the scaled magnitude of the gradient of the Gaussian smoothed image: $P(x, y) = -|\nabla[G * I(x, y)]|$. The weights $\alpha(s)$ and $\beta(s)$ control the relative importance of the first-order smoothness and the second-order smoothness. Generally, they can vary both along the length of the snake and over time. In practice, α is usually a positive constant, and β is usually zero. (We'll see later on that the second-order smoothness is not necessary.)

According to the calculus of variations, the snake that minimizes the energy functional \mathcal{E} must satisfy the Euler-Lagrange equation: $-\frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{c}}{\partial s}) + \frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{c}}{\partial s^2}) + \nabla P(\mathbf{c}) = 0$. Taking the snake as a dynamic system in terms of Lagrangian mechanics, the minima of the energy functional can be computed by solving the Lagrangian equation $\frac{\partial \mathbf{c}}{\partial t} = \frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{c}}{\partial s}) -$

$\frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{c}}{\partial s^2}) - \nabla P(\mathbf{c})$. An intuitive interpretation is that the shape of the snake changes over time driven by the internal stretching and bending forces (the first two terms in the right side of the above equation) and external image forces (the third term in the right side) starting from an initial shape. The term in the left side is referred to as the damping force and when it stabilizes (i.e. when it tends to zero), the energy functional goes to a local minimum. Finite difference methods [64] and finite element methods [65] have been used to discretize the snake and numerical time integration methods are applied to iteratively update the coordinates of the snakes over time steps until the difference of the snake between two successive iterations is sufficiently small. In the same spirit, snakes have been extended for 3D surfaces. The formulation is omitted here.

The main advantage of image segmentation using snakes is that it provides robustness to noise and spurious edges since the shapes of the curves or surfaces are regulated by their smoothness [66]. The limitations of the traditional deformable models including snakes are: 1) The snake must be initialized close to the structure of interest to guarantee good performance [63]; 2) The performance of the deformable model is also sensitive to the weighting parameters α and β ; 3) The snakes cannot extrude through any significant protrusions that a shape may possess without resorting to cumbersome resampling techniques [67]; 4) The topology of the object to be segmented must be known beforehand, that is, the snakes cannot adapt to the topology of the objects in the image without additional machinery; and 5) The *a priori* knowledge is limited to the smoothness of the contour. Various methods have been developed to address these disadvantages, as described below.

2.5.2 Level set based deformable models

Another type of deformable models that saves the parameter resampling and has the advantage of topology adaptability is the level set based models [68, 67]. Let $\phi(\mathbf{x})$ be a function from \mathbb{R}^n to \mathbb{R} . Then the deformable model \mathbf{c} in \mathbb{R}^n is defined as a specific level set of ϕ , typically zero level set $\phi(\mathbf{c}) = 0$. The model is one or more closed curves in \mathbb{R}^2 or one or more closed surfaces in \mathbb{R}^3 . A typical way to define the level set function is $\phi(\mathbf{x}) = d$, where $|d|$ is the shortest distance from \mathbf{x} to the zero level set with the sign chosen depending on if \mathbf{x} is outside or inside of the zero level set. The level set function

was originally proposed to solve front propagations problems such as flame propagation and crystal growth in [69], where it was shown that the motion equation of the front (i.e. the zero level set) $\frac{\partial \mathbf{c}}{\partial t} = F(\mathbf{c}(t))$ can be solved by solving instead the evolution equation of the level set function $\frac{\partial \phi}{\partial t} = -F|\nabla \phi|$. Here $F(\mathbf{c}(t))$ is the speed function of the front and t is the time domain $[0, \infty)$.

The local optimal front can be found by numerically updating ϕ over iterations until the changes of ϕ in zero level set are sufficiently small. When the level set function is defined over an 2D or 3D image, it should be updated for each pixel in the image and the speed function F has to be extended to have values over all pixels. For a pixel p that is not in the zero level set, its speed can be set to the speed of the point q in the zero level set which is of the shortest distance from p . One way to improve the efficiency is to only update ϕ within a narrow band around the zero level set while keeping all the others stationary until the zero level set collides with the bounds of the band, when the narrow band is reconstructed.

For image segmentation, a basic criterion of setting the speed function F is that it should be closer to zero in regions of high image gradient and closer to unity in regions with relatively homogeneous intensity. For example, [68, 67] shows that $F(\mathbf{x}) = g(|\nabla G * I(\mathbf{x})|)(c + \kappa)\vec{n}$, where c is a constant, κ is the curvature at point \mathbf{x} , \vec{n} is the unity normal vector at \mathbf{x} , and $g(|\nabla G * I(\mathbf{x})|)$ is a decreasing function of the gradient of the Gaussian smoothed image such that $g(|\nabla G * I(\mathbf{x})|) \rightarrow 0$ as $|\nabla G * I(\mathbf{x})| \rightarrow \infty$. Two examples of a reasonable g are $g(|\nabla G * I(\mathbf{x})|) = \frac{1}{1 + |\nabla G * I(\mathbf{x})|}$ and $g(|\nabla G * I(\mathbf{x})|) = e^{-|\nabla G * I(\mathbf{x})|}$. Here $\kappa\vec{n}$ acts as a smoothing force. The greater the curvature at \mathbf{x} , the greater the speed at \mathbf{x} ; positive curvature of a point p in the zero level set makes it deform inward and negative curvature make it go outward. The constant term c is referred to as the advection speed term and acts similarly as the pressure force or the weight force introduced in the parametric balloon model [70, 65], which is an extension of the traditional snake model. The pressure force inflates the model and the weight force deflates the model independent of the object geometry so that the boundary of the object can pass spurious edges when the initial model is not near the target model. The advection speed also helps form concave shapes if the model is not initialized properly.

In the geodesic active contours proposed in [71], the selection of the speed function F is formulated in terms of minimization of the energy functional \mathcal{E} in the traditional snakes. It is shown first that the smoothness of the contour can be sufficiently regularized by the first order term in \mathcal{E} . Generalizing the image potential term $P(c) = -|\nabla G * I(\mathbf{x})|$ with a decreasing function $g(|\nabla G * I(\mathbf{x})|)$, it is then proved that the minimization of the rewritten energy functional is equivalent to finding the geodesic contour in the Riemannian space with the Riemannian metric $g_{ij} = g(|\nabla G * I|)^2 \delta_{ij}$. This can be intuitively interpreted as the minimization of the length of the contour weighted by the measure of the edge salience ($g(|\nabla G * I|)$) of each point in the contour so that the Riemannian length of the contour is minimal when it is attached to salient edges. In order to minimize the Riemannian length of the contour, the gradient descent method (steepest descent method) is used to evolve the contour according to the equation $\frac{\partial \mathbf{c}}{\partial t} = g\kappa \vec{n} - (\nabla g \cdot \vec{n})\vec{n}$. Compared to the speed term in the previous paragraph, here the additional speed term $-(\nabla g \cdot \vec{n})\vec{n}$ increases the attraction of the deformation contour toward the boundary. It works like a doublet in that when the contour is approaching the boundary, the speed toward the boundary is increased and when the contour is leaving the boundary, the speed outof the boundary is decreased. The advection term $c\vec{n}$ can also be added to make the performance less dependent on the initialization of the contour.

All the discussions on the selection of the speed F can be extended easily for surfaces in \mathbb{R}^3 . The changes mainly involve the computation of the 3D gradients of Gaussian smoothed image and the curvature speed term can be derived from the mean surface curvature or the minimum surface curvature.

2.5.3 Image Influences

To address the initialization problem of traditional deformable models, much research have been done to impose global image influences on the contour in addition to or replacing the traditional image influence based on local gradients. Deformable models using global image influences include the balloon model [70, 65] that uses the pressure force and the weight force as described above, the gradient vector flow model [72, 73], and many models that incorporate region information [74, 75, 76, 77, 78].

The initialization problem of traditional deformable models with local gradient influences is mainly due to the fact that in homogeneous regions the gradients are nearly zero and the contour is mainly driven by the internal smoothness influences. Let f be the image intensity which may be Gaussian smoothed and ∇f be its gradient vector field. The main idea of the gradient vector flow is to construct a new gradient vector field \mathbf{u} in the image domain such that the vectors vary slowly in homogeneous regions and keep nearly equal values to ∇f at salient edges. This is achieved by solving diffusion equation $\frac{\partial \mathbf{u}}{\partial t} = \lambda(|\nabla f|)\nabla^2 \mathbf{u} - \mu(|\nabla f|)(\mathbf{u} - \nabla f)$ starting from the initial field $\mathbf{u} = \nabla f$. The first term on the right side of the equation is referred to as the smoothing term since it diffuses the gradient field ∇f . The second term encourages the vector field \mathbf{u} to be close to ∇f . λ and μ are two weighting functions that control the relative importance of these two terms. When the equilibrium solution is computed, the traditional image potential term ∇f is replaced with \mathbf{u} in the traditional deformable model. The gradient vector field \mathbf{u} has a larger capture range than the original gradient field ∇f and also helps move contours into boundary concavities.

2.6 Structural And Geometric A Priori Knowledge

A significant characteristic of the brain anatomy is that key anatomic surfaces in the human head are organized in a layered manner. These surfaces include the skin surface, the outer cortical surface, the inner cortical surface, and the ventricle surface. In particular, the thickness of the cortical layer (the shortest distances between points in the outer cortical surface and points in the inner cortical surface) is nearly constant. This characteristic has been used as structural and geometric *a priori* knowledge in both deformable models and statistical segmentation methods.

In [79], distances of each white matter voxel x_i to the skin surface and the ventricle surface are taken as a vector of two random variables (ds_i, dv_i) . The skin surface and the ventricle surface are believed to be much easier to segment and their segmentation is performed before that of the white matter. From segmented images, the joint probability

density of a random distance pair $P(ds_i, dv_i|x_i \in WM)$ is estimated with non-parametric methods such as histogramming or Parzen Windowing. The optimal segmentation is obtained by maximization of the posterior probability $P(x_i \in WM|ds_i, dv_i, I_i) \propto P(I_i|x_i \in WM)P(ds_i, dv_i|x_i \in WM)P(x_i \in WM)$, where I_i denotes the intensity of the voxel x_i . The first term at the rightside is a Gaussian intensity model for the white matter voxels and the third term is the prior probability that a pixel belongs to the white matter. The geometric prior knowledge can be used together with the MRF spatial prior model or by itself with stationary spatial prior.

In [80], the outer and inner cortical surface are represented as two polyhedral meshes. The deformation was formulated as a cost function minimization problem. The cost function is a weighted summation of several types of terms, including the image terms that push the deforming surface to tissue boundaries, internal stretching and bending terms that impose surface smoothness, and three additional proximity terms that prevent deformation from forming a self-intersecting surface and impose constraint on the thickness of the cortical layer. The self-proximity term $T_{self-proximity}$ is defined for every pair of polygons in the mesh. If the minimum distance between two polygons P_i and P_j is smaller than a threshold d_{ij} , $T_{self-proximity}(P_i, P_j) = (d(P_i, P_j) - d_{ij})^2$. Otherwise, $T_{self-proximity}$ is zero. The intersurface proximity term is defined in a similar manner. The weighting parameters of these two terms are set such that as the distance $d(P_i, P_j)$ approaches zero, the weighting parameters approach infinity. In this way, both self-intersection and intersurface intersection are prevented. The third term governs the thickness of the cortical layer and is defined for each pair of corresponding vertices in the two surfaces $T_{vertex-vertex} = (d(x_i, x_o) - d_B)^2$, where x_i and x_o are corresponding vertices in the inner cortical surface and outer cortical surface respectively and d_B is the preferred distance between the two vertices. The main disadvantage of this method is that the deformation involves the calculation of a huge number of distances between pairs of polygons and hence the algorithm is extremely slow.

Another coupled-surface deformation is proposed in [81] within the framework of level set formulation. Both the inner cortical surface and the outer cortical surface are embedded as zero-level sets in their level set function ϕ_{in} and ϕ_{out} respectively. The two evolution equations are given as $\frac{\partial \phi_{in}}{\partial t} + F_{in}|\nabla \phi_{in}| = 0$ and $\frac{\partial \phi_{out}}{\partial t} + F_{out}|\nabla \phi_{out}| = 0$ respectively. Since the value of the level set function of a front at any point is simply the distance from

this point to the current front, there is a natural way to establish a correspondence between the points on the two evolving surfaces through distance without adding much computational expense. For any point on the inner moving surface, the distance to the outer moving surface is the value ϕ_{out} at this point and vice versa for the point on the outer moving surface. The constraint on the thickness of the cortical layer is imposed by formulating the speeds as $F_{in} = \bar{F}_{in}h(\phi_{out})$ and $F_{out} = \bar{F}_{out}h(\phi_{in})$, where \bar{F}_{in} and \bar{F}_{out} are the speeds formulated without consideration of the thickness constraint of the cortical layer and $h(x)$ regularizes the speeds such that when the distance is too small or too large, the speed is reduced to zero.

2.7 Intensity Inhomogeneity Correction

There are two general ideas regarding how to overcome the intensity inhomogeneity problem in MR image segmentation. One is to correct the intensity inhomogeneity prior to brain tissue segmentation and the other is perform intensity inhomogeneity correction simultaneously with brain tissue segmentation. There are numerous methods to correct intensity inhomogeneity as a preprocessing step prior to image segmentation [82, 83, 84]. These methods is generally based on the assumption that intensity inhomogeneity is a low-frequency spatial variation that can be distinguished from higher-frequency components representing anatomic information [85]. It is believed by some researchers that the latter approach has the advantage of being able to use intermediate information from the segmentation while performing the correction.

There are two prevailing approaches for modeling inhomogeneities in methods that perform simultaneous segmentation [66]. The first approach assumes that each tissue class spatially varies independently; the second approach models the inhomogeneities as multiplicative gain field or additive bias field of the image logarithm. As a typical example of the first approach, [23] extends the standard mixture model resolving method with the MRF spatial prior model and the mean and variance of each tissue is allowed to vary over the spatial domain instead of using spatially invariant parameters. The main difference in

the implementation from the standard mixture resolving is that the spatially varying parameters in each voxel are estimated within a certain neighborhood of the voxel. The second approach is described in more detail in the following two subsections.

2.7.1 Adaptive Fuzzy C-Means

The standard fuzzy c-means objective function for partitioning an image represented as an voxel set $\{y_j | 1 \leq j \leq N\}$ into c clusters is $J = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^p \|y_j - m_i\|^2$, where $\{m_i | 1 \leq i \leq c\}$ are the prototypes (means) of the clusters, u_{ij} represents the fuzzy membership of the voxel y_j in the i -th cluster, and p is a weighting exponent and determines the amount of fuzziness of the resulting classification. Here, y_j represents the feature vector of each voxel and may simply contain the intensity of the voxel. The objective function is minimized when high membership values are assigned to voxels whose intensities are close to the centroid (means) of its particular class and low membership values are assigned when the voxel data is far from the centroid.

In [86], the observed intensities of the image $\{y_j | 1 \leq j \leq N\}$ are assumed to be multiplications of the anatomy field and the gain field $\{g_j | 1 \leq j \leq N\}$ and the objective function is modified to allow smooth intensity variation in a tissue: $J = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^p \|y_j - m_i g_j\|^2 + \lambda_1 R_1 + \lambda_2 R_2$, where R_1 and R_2 are two regularization terms that enforce the smoothness of the gain field and λ_1 and λ_2 are their controlling parameters. Here R_1 is the sum of the first-order finite differences at each pixel along rows and columns; R_2 is the sum of the second-order finite differences. One of the problems of this method is that it is sensitive to noise in the image. In [87], the intensities y_j are logarithmic converted to z_j and the multiplicative gain field is hence converted to an additive bias field β_j . A spatial regularizer is used to segment images corrupted by salt-and-pepper noise. The modified objective function is given by $J = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^p \|z_j - \beta_j - m_i\|^2 + \frac{\alpha}{N_R} \sum_{i=1}^c \sum_{j=1}^N u_{ij}^p (\sum_{z_r \in \mathcal{N}_j} \|z_r - \beta_r - m_i\|^2)$, where \mathcal{N}_j represents the neighbors of y_j and N_R is the cardinality of \mathcal{N}_j and α is a parameter that controls the importance of the regularization and depends on the signal-to-noise ratio of the image. The key reason why this objective function works is that the

regularization encourages the smoothness of both the membership values and the bias field at each voxel.

2.7.2 Adaptive Expectation-Maximization

In [88], intensity data is logarithmically transformed so that the multiplicative gain artifact is converted to an additive bias field. Observed log-intensity y_i at i th voxel is modeled as a normal distribution, independent of other voxels: $P(y_i|\omega_i = k, \beta_i) = G_{\sigma_k^2}(y_i - \mu_k - \beta_i)$, where $G_{\sigma_k^2}(\mu_k)$ is the normal distribution with mean μ_k and variance σ_k^2 , $\omega_i = k$ represents the tissue type of the voxel, and β_i denotes the bias field at i th voxel. In terms of the independence of probability between voxels, the probability density for the entire image is $P(y|\beta) = \prod_i P(y_i|\beta_i)$. The bias field β is modeled with a N -dimensional zero mean Gaussian prior probability density, where N is the total number of the voxels in the image. The posterior probability of the bias field, given observed intensity data, is $P(\beta|y) \propto P(y|\beta)P(\beta)$. Use the *maximum-a-posteriori* principle, the optimal estimate of β is determined by maximization of the posterior probability: $\hat{\beta} = \arg \max_{\beta} P(\beta|y)$. The equation to calculate the optimal β is derived using the zero gradient condition and is solved with the EM method in [88]. The model proposed in [88] is found to be problematic when there are tissues in an image that do not follow a Gaussian distribution. In [89], these tissues are unified into an outlier class with uniform distribution. Another extension is to use the MRF to incorporate spatial dependences of tissue types [21, 79, 90]. In the EM method that considers both bias field estimation and MRF spatial model, each iteration consists of following main steps: 1) estimate the bias field by maximization of its posterior probability, given the current estimation of the tissue types and the likelihood parameters; 2) update the likelihood distribution with the new estimation of the bias field; 3) estimate the tissue types by maximization of the posterior probability of the MRF; and 4) estimate the likelihood parameters by maximization of the expectation of the complete data log likelihood.

2.8 Brain Extraction

There are mainly four basic type of methods that have been proposed for brain extraction on MR images: the thresholding-with-morphology method, the watershed method, the deformable-model based method, and the atlas registration based method in which the image is normalized to a standard space. There are also hybrid methods that combine these basic methods for more accurate and robust results.

The thresholding-with-morphology [91] method essentially is conducted as following procedures. First, lower and upper thresholds are determined in order to separate the entire image into three parts: very bright parts such as eyeballs and fats, bright parts representing brain tissues, and dark parts including air and skull. The bright part, however, usually contains non-brain tissue and a morphology filtering is used to remove the non-brain bright part. An erosion operation is first performed on the bright part so that the “bridges” between brain and non-brain tissue are eliminated, the isolated component representing brain tissue is then determined, and this is then dilated back by the same extent as the erosion resulting in the final brain mask. There are some variants of this method in the thresholding part with more sophisticated methods for threshold selection using Gaussian mixture model [92] or histogram scale-space analysis [93]. Carefully tuned morphological filtering was also research in [93], but basically the metric “thickness” (i.e. the distance to boundary) was used as the measurement on the connectivity between brain and non-brain tissue.

The watershed method [94] obtained its name as the metaphor to the water flowing from hills to basins. For brain extraction in MRI, the gray level is first inverted so that white matter has lower intensity than gray matter and CSF and the intensity at each voxel is regarded as the “height” in the landscape. Voxels are then connected into “basins” in a way similar to how a watershed separates two adjacent river systems. This transform often leads to an “over-segmentation” problem where there are more basins in the result than are desired. This problem is often alleviated by a procedure called “preflooding” [94] to merge over-segmented basins. In brain extraction, a macro basin is finally determined as the result of brain extraction[94] or as intermediate result for further processing to get a more accurate result [95].

A typical deformable-model based brain extraction method is the one used in BET [96] (a brain extraction tool in the FSL package). In BET, the intensity histogram is processed to get a rough brain/non-brain threshold. Then the center-of-gravity of the head image is found and the rough size of the head in the image is determined. This information is then used to establish an initial triangular tessellation of a sphere's surface, which is deformed towards the brain's edge while maintaining reasonable size and smoothness.

A hybrid brain extraction method for more robust performance is proposed in the Freesurfer package [95]. An initial brain extraction is performed with the same watershed method presented in [94]. Then a surface model is established on the intermediate result and deformed to determine the brain's edge. In the deformation procedure, an atlas-based term is integrated so that the model is regularized with respect to both the smoothness and deviation from the atlas.

CHAPTER III

Relative Thresholding

Relative thresholding (RT) is characterized as differentiating the labels of near voxels by comparing their intensities with respect to a relative threshold [97]. RT is based on a structural model on the human brain anatomy and a model of T1-weighted human brain MR images. The modeling exploits various structural, geometrical and radiological *a priori* knowledge and is formulated as constraints in terms of first-order logic. This chapter starts with the structure modeling and the image modeling, and then presents the relative thresholding algorithm.

3.1 Structure Modeling

Let $\vec{g} = \nabla g(\sigma_{\nabla})$ be the gradient vector image of $g(\sigma_{\nabla})$. Throughout this dissertation, we use $g(\sigma)$ to denote the resultant image of performing Gaussian filtering with standard deviation σ on the input image y . We construct a directed graph $G = (V, E)$ from \vec{g} such that each vertex $v_i \in V$ corresponds to the voxel x_i in a region of interest R and each directed edge $e_i \in E$ emanates from v_i to v_j , where v_j is one of v_i 's 26-neighbors that is in the direction of the gradient vector \vec{g}_i . When v_j is outside R , e_i is forced to be a loop from v_i to itself.

The structural, geometrical and radiological *a priori* knowledge that we use in RT is:

- K_1 : skull, CSF, GM, and WM are organized as a layered structure from outside to inside;

- K_2 : The average intensities of skull, CSF, GM, and WM in local regions are in ascending order in T1-weighted MR images;
- K_3 : The cortex thickness is nearly uniform.

Based on this *a priori* knowledge we use a gradient graph to model the anatomical structure of the human brain as the following first-order logic. Let τ be the maximum cortex thickness of the subject and p is a value slightly greater than τ , then we can construct a gradient graph G with a suitable parameter σ_{∇} such that:

- For each GM voxel v_i , there is a path in G of length p from v_i to a WM voxel;
- For each CSF voxel v_i adjacent to GM, there is a path in G of length p from v_i to a WM voxel;
- For each CSF voxel v_i adjacent to GM, there is a path in G of length $\leq p$ from v_i to a GM voxel;
- There is no path from a WM voxel to a non-WM voxel in G ; and
- There is no path from any non-brain voxels to WM in G without passing CSF.

3.2 Image Modeling

A common approach to image segmentation is based on the image modeling in which image intensities are modeled as statistical distributions. While the intensities of WM voxels in the T1-weighted MR image can be safely modeled with common statistical distributions (e.g. a normal distribution) once the intensity inhomogeneity has been corrected, the intensities of GM voxels hardly meet any common statistical distributions in practice even if intensity inhomogeneity has already been corrected. Based on this observation, we model images in terms of the spatial relationships between voxels instead of as statistical distributions on the absolute voxel intensities. The basic intuition is that if the segmentation task is not beyond the human recognition capability, near voxels of the same type

should possess less difference in intensity than near voxels of different types. With this type of image modeling, we attempt to avoid the limitation imposed by the form of statistical distributions and provide a framework for introducing various *a priori* knowledge into the segmentation task.

Suppose that there are K voxel types among a total of N voxels in the space domain Ω , which represents either the whole image or a region of interest in the image. In brain MR image segmentation, we assume that K equals to 3 and the three tissues of interest are WM, GM, and background and denoted as with the number 3, 2 and 1 respectively. When the domain Ω contains exactly the brain volume, the background tissue type corresponds to CSF; otherwise, it refers to anything excluding WM and GM. In this thesis, we set Ω to be whole image space and do not depend on any prior skull stripping or brain extraction procedure. Let the coordinates of voxels be x_i , $1 \leq i \leq N$, and the variable and true label of each voxel respectively be ω_i (or $\omega(x_i)$) $\in [1, K]$ and $\bar{\omega}_i \in [1, K]$, $1 \leq i \leq N$. Incorporating a multiplicative bias field b_i and an additive noise ρ , the image intensity y_i (or $y(x_i)$), $1 \leq i \leq N$, is modeled as:

$$y_i = b_i \sum_{k=1}^K \delta_i^k y_i^k + \rho, \text{ where } \delta_i^k = \begin{cases} 0 & \bar{\omega}_i \neq k \\ 1 & \bar{\omega}_i = k \end{cases} \quad (3.1)$$

In equation 3.1, $\delta_i^k y_i^k$ represents the component given by tissue k in the ideal image without influence from noise and IHH and we refer their sum $\sum_{k=1}^K \delta_i^k y_i^k$ as the *ideal image*. Here we do not assume any particular statistical form on the noise term. Equation 3.1 is our initial image model and will be gradually transformed to facilitate image segmentation.

The term y_i^k in equation 3.1 can be seen as an arbitrary function over the space domain governed by the constraints on the spatial relationship between near voxels. Generally, we think the constraints should consider *a priori* knowledge about the structure and geometry of the objects in the image as well as the inherent image properties related to the image acquisition process. In brain T1-weighted MR images, we consider *a priori* knowledge K_1 and K_2 and use the following first-order logic to describe a spatial constraint:

$$\begin{aligned}
& \forall x_i, x_j \in \Omega \forall k \in [1, K] \exists T^k \in [0, 1) d(x_i, x_j) \leq p \Rightarrow \\
& (\bar{\omega}_i = k \wedge \bar{\omega}_j = k + 1 \Rightarrow r(y_i^k, y_j^{k+1}) < T^k) \wedge \\
& (\bar{\omega}_i = \bar{\omega}_j = (k + 1) \Rightarrow r(y_i^k, y_j^k) \geq T^k), \\
& \text{where } r(a, b) = \begin{cases} a/b & a < b \\ b/a & a \geq b \end{cases} \quad (3.2)
\end{aligned}$$

In equation 3.2, T^0 and T^K are forced to be 0, $d(x_i, x_j)$ represents the distance between voxel x_i and x_j , and p is the distance threshold (a voxel cube is of unit dimension) and is in the same value as the one used in the structure modeling in section 3.1. Theoretically, any form of distance, including Euclidean distance, can be used. However, D^6 , D^{18} or D^{26} distance is preferable because of the computational efficiency.

Spatial constraints expressed in equation 3.2 can also be described informally as follows. Let r_{gw} represent the ratio between a GM voxel x_i and a WM voxel x_j near x_i (we use a distance threshold to express the nearness between two voxels). Let r_{bg} represent the ratio between a background voxel x_m and a GM voxel x_n near x_m . Let r_{ww} , r_{gg} and r_{bb} , respectively, represent the ratio between two near WM voxels, two near GM voxel and two near background voxels. Then we have the following constraints on the four ratios:

$$r_{gw} > r_{ww} \wedge r_{bg} > r_{gg} \quad (3.3)$$

Note that inequation 3.3 leaves a great deal of freedom for r_{gg} and r_{bb} so that: 1) the intensities of two near GM voxels can differ even more than any pair of near voxels of GM and WM; and 2) the intensities of two near background voxels can differ even more than those of any pair of near voxels of CSF and GM. In this way, we allow a greater extent of variations among GM voxels and permit the background to include various type of tissue types such as CSF, skull, air and so on.

A reasonable assumption about the bias field is that it varies slowly across the space with respect to the intensity variation between different tissues in the ideal image. We use a first-order logic to describe this assumption in equation 3.4 without any constraints on the variation patterns.

$$\begin{aligned} & \forall x_i, x_j \in \Omega \exists \epsilon \in (0, 1) \\ (1 - \epsilon \ll 1 - \max(T^1, \dots, T^{K-1})) \wedge (d(x_i, x_j) \leq p \Rightarrow r(b_i, b_j) > \epsilon) \end{aligned} \quad (3.4)$$

Based on the low frequency property of the bias field, we can safely let y_i^k absorb the bias field term and the latter can thus be dropped from equation 3.1 while validity of the constraint in equation 3.2 is maintained. Therefore, the image artifact of IHH is made transparent in our image model.

Next, we apply Gaussian filtering on the original gray level image to counteract the noise and drop the noise term from equation 3.1. Let $z = g(\sigma_z)$ be a specific blurred image. The new image model on z is:

$$z_i = \sum_{k=1}^K \delta_i^k z_i^k \quad (3.5)$$

Here, z^k corresponds to the contribution of tissue k to the smoothed image. After Gaussian filtering, we want to maintain the spatial relationships between voxels, as described below:

$$\begin{aligned} & \forall x_i, x_j \in \Omega \forall k \in [1, K] \exists T^k \in [0, 1] d(x_i, x_j) \leq p \Rightarrow \\ & (\bar{\omega}_i = k \wedge \bar{\omega}_j = k + 1 \Rightarrow r(z_i^k, z_j^{k+1}) < T^k) \wedge \\ & (\bar{\omega}_i = \bar{\omega}_j = k + 1 \Rightarrow r(z_i^k, z_j^k) \geq T^{k+1}) \end{aligned} \quad (3.6)$$

In terms of the definition of the function δ_i^k , we have $z_i^k = z_i$ when $\omega_i = k$ and hence the equation 3.6 can be rewritten as:

$$\begin{aligned} & \forall x_i, x_j \in \Omega \forall k \in [1, K] \exists T^k \in [0, 1] d(x_i, x_j) \leq p \Rightarrow \\ & (\bar{\omega}_i = k \wedge \bar{\omega}_j = k + 1 \Rightarrow r(z_i, z_j) < T^k) \wedge \\ & (\bar{\omega}_i = \bar{\omega}_j = k + 1 \Rightarrow r(z_i, z_j) \geq T^{k+1}) \end{aligned} \quad (3.7)$$

It is well-known that Gaussian filtering blurs both homogeneous regions and edges. This might lead to main two types of violation to the constraint. First, for a voxel pair

(x_i, x_j) of different types on the opposite sides of an edge, if they are too close to each other, $r(z_i, z_j)$ may be significantly increased such that they may be identified as the same type. We think this adverse effect can be minimized by increasing the distance between voxel pairs for comparison in the relative thresholding procedure. This can be demonstrated in figure 3.1, where the spatial constraint is maintained in the Gaussian blurred 1D signal with $\sigma_z = 2$, $p = 10$, and $T^1 = 0.45$. The second type of violation may occur when the dimension of some parts of the structure of interest is too narrow compared to the Gaussian filter aperture (σ_z). We found that for current MRI techniques, the usual resolution (around $1mm^3$) is high enough so that this violation brings very little negative influence.

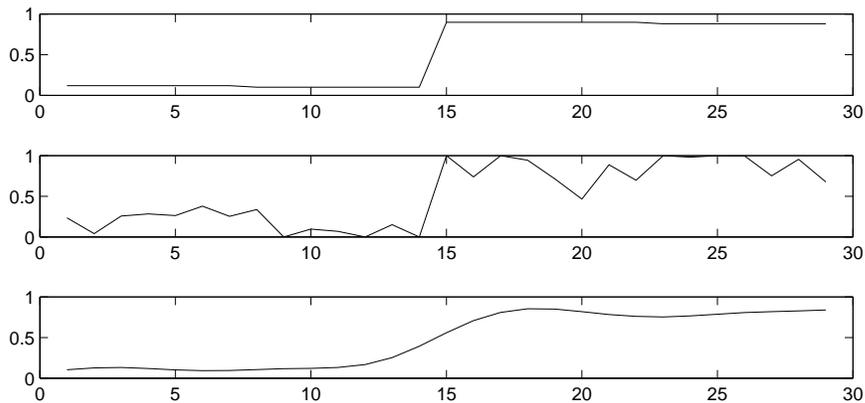


FIGURE 3.1: Effect of Gaussian smoothing of a 1D signal. Top: a 1D signal; Middle: noise added; Bottom: smoothed signal.

3.3 Applying Relative Thresholds

Suppose we are given the two relative thresholds T_{gw} (between GM and WM) and T_{bg} (between background and GM) that govern the constraints in equation 3.7. We can combine the structure model and the image model and easily obtain the following rule to differentiate WM, GM and background:

1. All voxels are initialized as WM.
2. For any voxel pair (x_i, x_j) both labeled as WM, if there is a path from x_i to x_j of length p in the gradient graph G and their intensity ratio $(x_i/x_j < T_{gw})$, then x_i is relabeled as GM. This rule sets all true GM voxels as GM and will not flip the label of true WM voxels. In other words, this rule detects borders between WM and GM.
3. For any voxel pair (x_i, x_j) both labeled as GM, if there is a path from x_i to x_j of length $\leq p$ in G and their intensity ratio $z_i/z_j < T_{bg}$, then x_i is relabeled as background. This rule detects borders between background (including CSF) and GM.

We designed two algorithms, algorithm 1 and algorithm 2, to implement the above rules for GM/WM segmentation and background/GM segmentation respectively. The inputs to both include a comparing image z from which voxels are compared, a relative threshold, and a gradient graph G constructed on the entire image domain. Both algorithms mainly consist of a sequence of voxel comparisons. Each comparison involves an objective voxel and a reference voxel. The objective voxel is the voxel whose tissue type is to be determined at the present comparison. The reference voxel is determined as follows.

- In algorithm 1, the reference voxel $ref_{gw}(v_i)$ for the object voxel v_i is the WM voxel with the maximum intensity in the path of length p that emanates from v_i in G ;
- In algorithm 2, the reference voxel $ref_{bg}(v_i)$ for the object voxel v_i is the GM voxel with the maximum intensity in the path of length p that emanates from v_i in G .

In practice, we found that it gave better results to temporarily substitute the intensity $z(ref_{bg}(v_i))$ with $z(ref_{gw}(ref_{bg}(v_i)))(1 - (1 - T_{gw}) * 2)$ for each comparison in algorithm 2. This is based on our observation that 1 minus the optimal threshold T_{gw}^* is roughly half of 1 minus the ratio between average GM and WM intensities.

The distance threshold p is an empirical value that we chose based on the average cortex thickness. In some areas of the brain, such as amygdala and caudate, the gray matter may be thicker than the average cortex and some gray matter voxels may be recognized as WM because p is too short. To work around this issue, we add an additional rule in algorithm

Algorithm 1: GW-Thresholding

Data: z, G, T_{gw}
 initialization: $\forall x_i \in R, \omega_i \leftarrow \text{WM};$
foreach voxel x_i in the image do
 if $\omega(\text{ref}_{gw}(x_i)) = \text{GM}$ then
 $\omega_i \leftarrow \text{GM};$
 else if $r(z_i, z(\text{ref}_{gw}(x_i))) < T_{gw}$ then
 $\omega_i \leftarrow \text{GM};$
foreach voxel x_i in the image do
 if $\omega(\text{ref}_{gw}(x_i)) = \text{GM}$ then
 $\omega_i \leftarrow \text{GM};$

Algorithm 2: BG-Thresholding

Data: z, G, T_{bg}
foreach voxel x_i in the image do
 if $\omega_i = \text{GM}$ and $r(z_i, z(\text{ref}_{bg}(x_i))) < T_{bg}$ then
 $\omega_i \leftarrow \text{background};$

1: if the reference voxel of the object voxel x_i is relabeled as GM, then x_i should also be relabeled as GM.

3.4 Finding Optimal Thresholds

Optimal thresholds can be found by user intervention in a trial-and-evaluation scheme. The user can first try different candidate thresholds between GM and WM, visually evaluate the result at the same time and finally select the threshold that gives the best result. The same procedure can be performed to select the best threshold between background and GM. It is under investigation whether the same two optimal thresholds can be used as *a priori* knowledge across different MR images acquired with the same or similar parameters.

Figure 3.2 gives a demonstration on the effects of choosing different relative thresholds: figure 3.2(b) shows that over-low threshold T_{gw} globally makes the white matter too fat while figure 3.2(c) shows that over-high T_{gw} globally makes it too thin. In either case,

the GM/WM boundary drifts away the correct situation in different directions while the optimal relative threshold makes the boundary fit at the right location, as shown in figure 3.2(d).

3.5 Results

Figure 3.3 shows segmentation results of applying relative thresholding on some real MRI scans. One of the advantages of relative thresholding is that it is robust to intensity inhomogeneity without additional correction processing. Another advantage is that it can adapt to high level intensity variations within a given tissue.

Relative thresholding mainly serves as a initial voxel classification processing step. Since relative thresholding is performed on the entire image, non-brain voxels can be labeled as brain voxels which may nor may not be connected to the true brain voxels. This type of false positive will be eliminated with morphometric analysis presented in chapter VI and chapter VIII. Another typical type of error misses certain fine portions of the WM structure. This type of false negative is mainly due to limited sampling resolution and the blurring effect of Gaussian filter. A procedure called *terrain analysis* will be presented in chapter VIII to largely recover these missing portions of the structure. Topology defects are yet another type of segmentation error, although little geometrical deviation may be involved. Topology correction methods will be presented in chapter VII and chapter VIII.

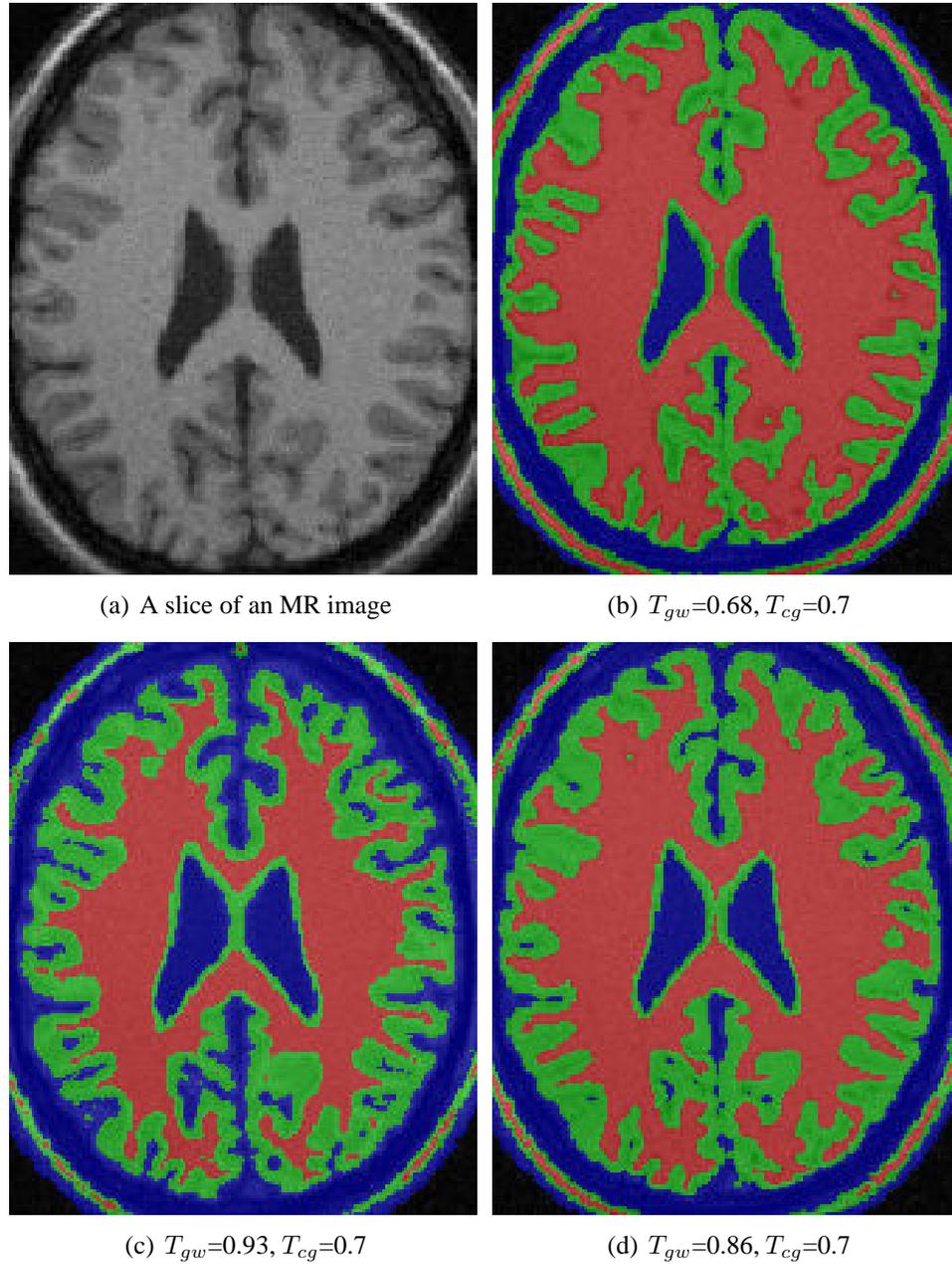
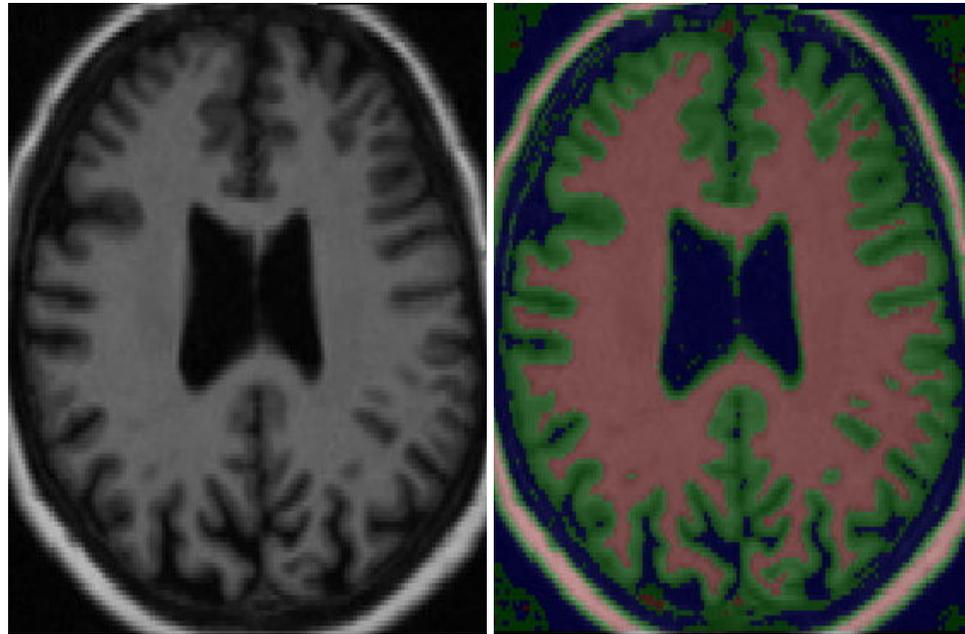
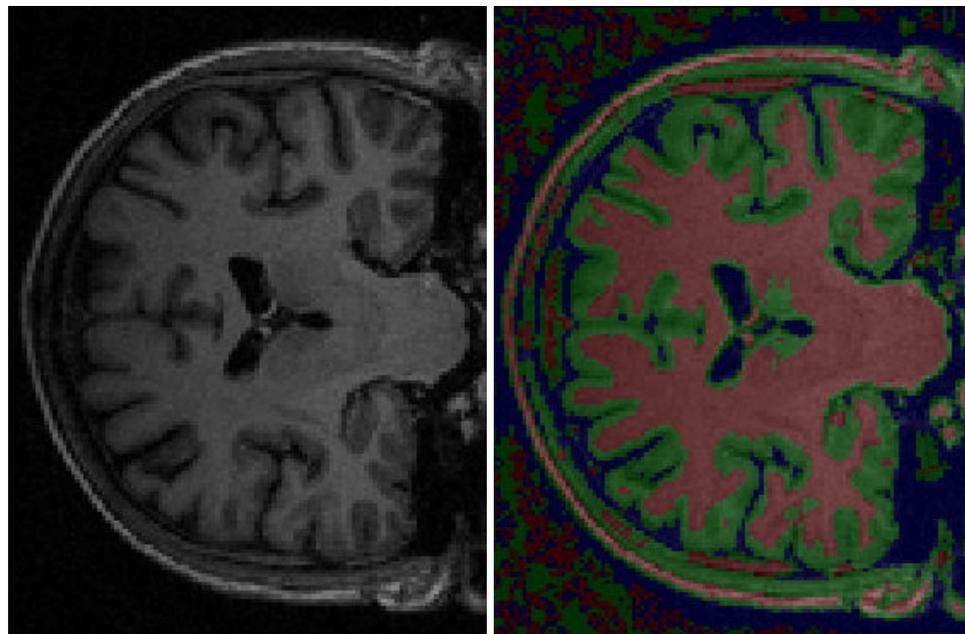


FIGURE 3.2: Analysis of relative thresholding with different relative thresholds applied on a phantom image. Optimal $T_{gw}^* = 0.86$ and optimal $T_{cg}^* = 0.7$. The relative threshold T_{gw} in (b) is over-low. The one in (c) is over-high. The one in (d) is optimal.



(a) A slice of a MR image

(b) RT result



(c) A slice of another MR image

(d) RT result

FIGURE 3.3: Relative thresholding results on real MR scans

CHAPTER IV

Digital Point Classification

Like most other pixel classification methods, relative thresholding will inevitably introduce classification errors in both geometry and topology. To correct classification errors, we rely on a set of morphological image analysis tools, which are described in the following chapters. In this chapter, we give a set of definitions on the classification of 3D digital points, which forms the basis for the morphological image analysis tools. We start with the introduction of fundamental digital topology theory.

4.1 Digital Topology

A 3D binary image is defined as the quadruple $(\mathcal{V}, n, \bar{n}, \mathcal{F})$ [98]. $\mathcal{V} \subset \mathcal{Z}^3$ is the 3D cubic grid representing all elements in the image. Each element in the 3D image is a cubic grid point and called a voxel. $\mathcal{F} \subset \mathcal{V}$ represents the set of foreground voxels and $\bar{\mathcal{F}}$ represents the complement of \mathcal{F} . n and \bar{n} respectively represent the adjacency in \mathcal{F} and $\bar{\mathcal{F}}$ defined below.

The topology of a digital image depends on a pair of digital adjacencies, one for the foreground and one for the background. Three types of adjacency are commonly used in 3D: 6-, 18-, and 26-adjacency. Two voxels are 6-adjacent if they share a face, 18-adjacent if they share a face or an edge, and 26-adjacent if they share a face, an edge, or a corner. In the rest of this thesis, we use “voxel” and “point” without distinction if not particularly noted. An n -neighbor of a point p is a point that is n -adjacent to p . The set of n -neighbors

of a point p is denoted as $\mathcal{N}_n(p)$. Note that here $\mathcal{N}_n(p)$ includes p since a voxel p can be seen as sharing six faces with itself. We use $\mathcal{N}_n^*(p)$ to denote $\mathcal{N}_n(p) \setminus \{p\}$ and $\mathcal{N}(p)$ to denote $\mathcal{N}_{26}(p)$. Topologically compatible adjacencies/connectivities of \mathcal{F} and $\overline{\mathcal{F}}$ are $(6, 26)$, $(6, 18)$, $(18, 6)$ and $(26, 6)$. For any adjacency $n \in \{6, 18, 26\}$, we use \overline{n} to denote its compatible adjacency.

An n -path of $l > 0$ from point p to q in $\mathcal{X} \subset Z^3$ represents a sequence of distinct points $p = p_0, p_1, \dots, p_l = q$ in \mathcal{X} such that p_i is n -adjacent to p_{i+1} , for $i = 0, 1, \dots, l-1$. An n -path is *closed* if and only if p_0 is n -adjacent to p_l . Two points $p, q \in \mathcal{X}$ are n -connected with respect to \mathcal{X} if and only if there exists an n -path from p to q in \mathcal{X} . The set \mathcal{X} is n -connected if every two points in \mathcal{X} are n -connected with respect to \mathcal{X} . An n -connected component of \mathcal{X} is a non-empty n -connected subset of \mathcal{X} that is not n -adjacent to any other point in \mathcal{X} . The set of all n -connected components of \mathcal{X} is denoted by $\mathcal{C}_n(\mathcal{X})$.

For any set $\mathcal{X} \subset \mathcal{V}$, we use $\overline{\mathcal{X}}$ to denote the complement of \mathcal{X} in the image. The point set \mathcal{X} is also referred to as an object since it corresponds to a binary object in the image. An object \mathcal{X} in n -adjacency has a *cavity* if and only if there exists a connected component in $\overline{\mathcal{X}}$ in \overline{n} -adjacency that is \overline{n} -connected to only \mathcal{X} .

An object \mathcal{X} has a *handle* whenever there is a closed path in \mathcal{X} that cannot be deformed through connected deformations in \mathcal{X} to a single point. A solid torus is an example of an object that has exactly one handle. The number of handles in an object is the maximum number of cutting along embedded disks without rendering the resultant object disconnected. A handle in the object \mathcal{X} is referred to as a *tunnel* in its complement $\overline{\mathcal{X}}$. The number of handles in a digital object is also called the *genus* of the object.

A central concept in digital topology is the definition of *simple point* [99], which is characterized by its *geodesic neighborhood* and *topological numbers*.

Definition 4.1.1. Simple points A point in a binary image $(\mathcal{V}, n, \overline{n}, \mathcal{F})$ is *simple* if it can be added to or removed from \mathcal{F} without changing the topology of both \mathcal{F} and $\overline{\mathcal{F}}$, i.e. without changing the number of connected components, cavities and handles of both \mathcal{F} and $\overline{\mathcal{F}}$.

Definition 4.1.2. Geodesic neighborhood The geodesic neighborhood of a point $p \in \mathcal{V}$ with respect to $\mathcal{X} \subset \mathcal{V}$ of order k is the set $\mathcal{N}_n^k(p, \mathcal{X})$ defined recursively by :

$$\begin{aligned}\mathcal{N}_n^1(p, \mathcal{X}) &= \mathcal{N}_n^*(p) \cap \mathcal{X} \\ \mathcal{N}_n^k(p, \mathcal{X}) &= \cup\{\mathcal{N}_n(q) \cap \mathcal{N}_{26}^*(p) \cap \mathcal{X}, q \in \mathcal{N}_n^{k-1}(p, \mathcal{X})\}\end{aligned}$$

Definition 4.1.3. Topological numbers The topological numbers of the point p relative to the set \mathcal{X} are:

$$\begin{aligned}T_6(p, \mathcal{X}) &= \#\mathcal{C}_6(\mathcal{N}_6^2(p, \mathcal{X})) \\ T_{6^+}(p, \mathcal{X}) &= \#\mathcal{C}_6(\mathcal{N}_6^3(p, \mathcal{X})) \\ T_{18}(p, \mathcal{X}) &= \#\mathcal{C}_{18}(\mathcal{N}_{18}^2(p, \mathcal{X})) \\ T_{26}(p, \mathcal{X}) &= \#\mathcal{C}_{26}(\mathcal{N}_{26}^1(p, \mathcal{X})),\end{aligned}$$

where $\#$ denotes set cardinality and 6^+ denotes 6-adjacency whose dual adjacency is 18 while 6 denotes 6-adjacency whose dual adjacency is 26.

It is proven in [99] that a simple point can be characterized by local computation of its topological numbers within the $3 \times 3 \times 3$ neighborhood of the point:

Theorem 4.1.4. *A point p in a 3D binary image $(\mathcal{V}, n, \bar{n}, \mathcal{F})$ is simple if and only if $T_n(p, \mathcal{F}) = 1$ and $T_{\bar{n}}(p, \bar{\mathcal{F}}) = 1$.*

Given a set $\mathcal{X} \subseteq \mathcal{F}$, a point p is simple relative to \mathcal{X} if and only if $T_n(p, \mathcal{X}) = 1$ and $T_{\bar{n}}(p, \bar{\mathcal{X}}) = 1$.

Another concept critical to our topology correction method presented in chapter VII is the definition of *multisimple point* [100].

Definition 4.1.5. Multisimple points A point p is *multisimple* relative to the set \mathcal{X} if and only if it can be added to or removed from \mathcal{X} without changing the number of handles and cavities of \mathcal{X} while splitting and merging connected components in \mathcal{X} are allowed.

Characterization of multisimple points is given as the following theorem [100].

Theorem 4.1.6. *Let $T_n^+(p, \mathcal{F})$ and $T_{\bar{n}}^+(p, \bar{\mathcal{F}})$ respectively denote the number of foreground and background components in the $\mathcal{V} \setminus p$ that are adjacent to a point p , then p is multisimple relative to \mathcal{F} iff $T_{\bar{n}}(p, \bar{\mathcal{F}}) = 1$ and $T_n^+(p, \mathcal{F}) = T_n(p, \mathcal{F})$; p is multisimple relative to $\bar{\mathcal{F}}$ iff $T_n(p, \mathcal{F}) = 1$ and $T_{\bar{n}}^+(p, \bar{\mathcal{F}}) = T_{\bar{n}}(p, \bar{\mathcal{F}})$.*

4.2 Point Classification

We can classify all points in the set \mathcal{X} into *border points* and *interior points*.

Definition 4.2.1. Border points A point $p \in \mathcal{X}$ is a border point relative to \mathcal{X} if $T_{\bar{n}}(p, \overline{\mathcal{X}}) > 0$, where n is the adjacency of X .

Definition 4.2.2. Interior points A point $p \in \mathcal{X}$ is an interior point relative to \mathcal{X} if $T_{\bar{n}}(p, \overline{\mathcal{X}}) = 0$, where n is the adjacency of X .

By the definition of simple points and border points, it is obvious that a simple point relative to \mathcal{X} must be a border point relative to \mathcal{X} . Therefore, the set of all border points \mathcal{B} relative to \mathcal{X} can be classified into the set of simple points and the set of non-simple points.

Corollary 4.2.3. *Let $\mathcal{B} \in \mathcal{X}$ be the set of all border points relative to \mathcal{X} and \mathcal{S} be the set of all simple points relative to \mathcal{X} , we have $\mathcal{S} \subseteq \mathcal{B}$.*

Simple points can be further classified into *thin-simple points* and *thick-simple points*.

Definition 4.2.4. Thick-simple points A point $p \in \mathcal{X}$ is a thick-simple point relative to \mathcal{X} if it is simple relative to \mathcal{X} and its removal from \mathcal{X} does not increase the number of tunnels and number of connected components in $\mathcal{N}_n(q) \cap \mathcal{X} \setminus \{p\}$, for all q in $\mathcal{N}_{26}^*(p) \cap \mathcal{X}$, i.e.:

$$\begin{aligned} & p \text{ is a simple point relative to } \mathcal{X}, \text{ and} \\ & \forall q \in \mathcal{N}_{26}^*(p) \cap \mathcal{X}, T_n(q, \mathcal{X} \setminus \{p\}) \leq T_n(q, \mathcal{X}), \text{ and} \\ & \forall q \in \mathcal{N}_{26}^*(p) \cap \mathcal{X}, T_{\bar{n}}(q, \overline{\mathcal{X}} \cup \{p\}) \leq T_{\bar{n}}(q, \overline{\mathcal{X}}) \vee T_{\bar{n}}(q, \overline{\mathcal{X}}) = 0, \end{aligned}$$

where n denotes the adjacency of \mathcal{X} .

Definition 4.2.5. Thin-simple points A point $p \in \mathcal{X}$ is a thin-simple point relative to \mathcal{X} if p is a simple point relative to \mathcal{X} , but p is not a thick-simple point relative to \mathcal{X} .

Now, let's turn to non-simple points and perform further classification on them.

Definition 4.2.6. Thin-surface points A point $p \in \mathcal{X}$ is a thin-surface point relative to \mathcal{X} if $T_{\bar{n}}(p, \overline{\mathcal{X}}) > 1$, where n denotes the adjacency of \mathcal{X} .

If a point $p \in \mathcal{X}$ is a thin-surface point relative to \mathcal{X} , it implies that the number of tunnels in the neighborhood $\mathcal{N}_n^*(p) \cap \mathcal{X}$ is greater than 0.

Definition 4.2.7. Thin-curve points A point $p \in \mathcal{X}$ is a thin-curve point relative to \mathcal{X} if $T_{\bar{n}}(p, \overline{\mathcal{X}}) = 1$ and $T_n(p, \mathcal{X}) > 1$.

If a point $p \in \mathcal{X}$ is a thin-curve point relative to \mathcal{X} , it implies that the number of components in the neighborhood $\mathcal{N}_n^*(p) \cap \mathcal{X}$ is greater than 1.

Definition 4.2.8. Isolated points A point $p \in \mathcal{X}$ is an isolated point relative to \mathcal{X} if $T_{\bar{n}}(p, \overline{\mathcal{X}}) = 1$ and $T_n(p, \mathcal{X}) = 0$.

In terms of the definition of multisimple points, we can further classify thin-curve points into finger points and handle points, which are involved in our topology correction method.

Definition 4.2.9. Finger points A point $p \in \mathcal{X}$ is a finger point relative to \mathcal{X} if p is a thin-curve point relative to \mathcal{X} and a multisimple point relative to \mathcal{X} .

Definition 4.2.10. Handle points A point $p \in \mathcal{X}$ is a handle point relative to \mathcal{X} if p is a thin-curve point relative to \mathcal{X} but not a multisimple point relative to \mathcal{X} .

Next, we further classify thin-simple points into thick-surface points and thick-curve points.

Definition 4.2.11. Thick-surface points A point $p \in \mathcal{X}$ is a thick-surface point relative to \mathcal{X} if p is a simple point relative to \mathcal{X} and there exists a point $q \in \mathcal{N}_{26}^*(p) \cap \mathcal{X}$ such that $T_{\bar{n}}(q, \overline{\mathcal{X}} \cup \{p\}) > T_{\bar{n}}(q, \overline{\mathcal{X}})$ and $T_{\bar{n}}(q, \overline{\mathcal{X}}) > 0$.

If a point $p \in \mathcal{X}$ is a thick-surface point relative to \mathcal{X} implies that removal of p from \mathcal{X} increases the number of tunnels in the neighborhood $\mathcal{N}_n^*(q) \cap (\mathcal{X} \setminus \{p\})$.

Definition 4.2.12. Thick-curve points A point $p \in \mathcal{X}$ is a thick-curve point relative to \mathcal{X} if p is a simple point relative to \mathcal{X} , is not a thick-surface point relative to \mathcal{X} , and there exists a point $q \in \mathcal{N}_{26}^*(p) \cap \mathcal{X}$ such that $T_n(q, \mathcal{X} \setminus \{p\}) > T_n(q, \mathcal{X})$.

If a point $p \in \mathcal{X}$ is a thick-curve point relative to \mathcal{X} , it implies that removal of p from \mathcal{X} increases the number of components in the neighborhood $\mathcal{N}_n^*(q) \cap (\mathcal{X} \setminus \{p\})$.

Thick-simple points can also be further classified into several types according to their geometrical characterization.

Definition 4.2.13. Volume-boundary points A point $p \in \mathcal{X}$ is a volume-boundary point relative to \mathcal{X} if it is a thick-simple point and is n -adjacent to an interior point relative to \mathcal{X} , where n is the adjacency of \mathcal{X} .

Definition 4.2.14. Surface-edge points A point $p \in \mathcal{X}$ is a surface-edge point relative to \mathcal{X} if it is a thick-simple point, is not a volume-boundary point, and is n -adjacent to a thin-surface point or a thick-surface point relative to \mathcal{X} , where n is the adjacency of \mathcal{X} .

Definition 4.2.15. Curve-end points A point $p \in \mathcal{X}$ is a curve-end point relative to \mathcal{X} if it is a thick-simple point, is neither a volume-boundary point nor a surface-edge point, and is n -adjacent to a thin-curve point or thick-curve point relative to \mathcal{X} , where n is the adjacency of \mathcal{X} .

Definition 4.2.16. Very-thick-curve points A point $p \in \mathcal{X}$ is a very-thick-curve point relative to \mathcal{X} if it is a thick-simple point, not a volume-boundary point, a surface-edge point, nor a curve-end point.

In some situations, what form the end of a curve may include more than one point. We define *thick curveend points* and *thin curveend points* to differentiate two types of curve ends.

Definition 4.2.17. Thick curve-end points A point $p \in \mathcal{X}$ is a thick curve-end point relative to \mathcal{X} if it is a curve-end point relative to \mathcal{X} and n -adjacent to another curve-end point relative to \mathcal{X} , where n is the adjacency of \mathcal{X} .

Definition 4.2.18. Thin curve-end points A point $p \in \mathcal{X}$ is a thin curve-end point relative to \mathcal{X} if it is a curve-end point relative to \mathcal{X} but not a thick curve-end point relative to \mathcal{X} .

We can also differentiate surface-edge points into types: *thick surface-edge points* and *thin surface-edge points*. For simplicity, we define these two types of surface-edge points only for surface-edge points that are adjacent to thin-surface points.

Definition 4.2.19. Thick surface-edge points of thin surface A point $p \in \mathcal{X}$ is a thick surface-edge point of thin surface relative to \mathcal{X} if it is a surface-edge point adjacent to at least one thin-surface point but no thick-surface points and for all thinsurface point $q \in \mathcal{N}_{26}^*(p) \cap \mathcal{X}$, $T_n(q, \overline{\mathcal{X}}) = T_n(q, \overline{\mathcal{X}} \cup \{p\})$, where n is the adjacency of \mathcal{X} .

If a point $p \in \mathcal{X}$ is a thick surface-edge point of thin surface relative to \mathcal{X} , it implies that its removal from \mathcal{X} does not change the number of tunnels in the neighborhood of any thinsurface point adjacent to p .

Definition 4.2.20. Thin surface-edge points of thin surface A point $p \in \mathcal{X}$ is a thin surface-edge point of thin surface relative to \mathcal{X} if it is a surface-edge point adjacent to at least one thin-surface point but no thick-surface points and is not a thick surface-edge points of thin surface relative to \mathcal{X} .

Examples of some of the major definitions given above are illustrated in figure 4.1. The classification of 3D digital points forms a tree structure, as shown in figure 4.2.

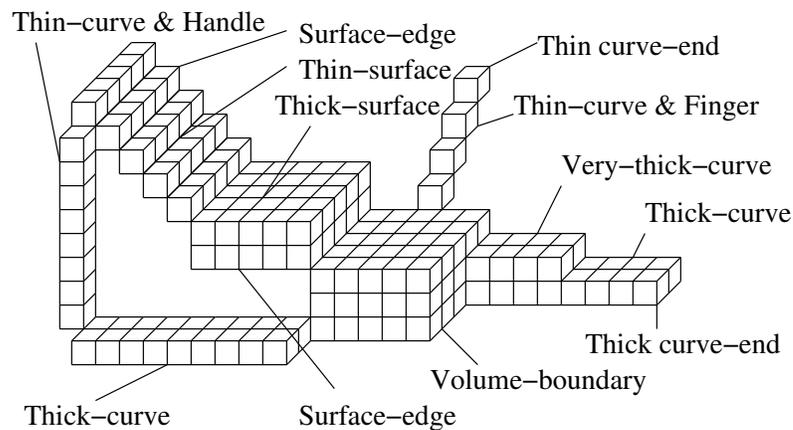


FIGURE 4.1: 3D digital point types. The object is in 26-adjacency.

4.3 Related Work

In [101], 3D digital points are classified into the following types: interior point, isolated point, border point, curve point, curves junctions, surface point, surface-curve(s) junction, surface junction, and surfaces-curve(s) junction. In [102], 3D digital points in a surface skeleton are classified into the following types: edge point of surface, inner point of surface, junction point of surfaces, junction point of surface and curves, curve end point, inner

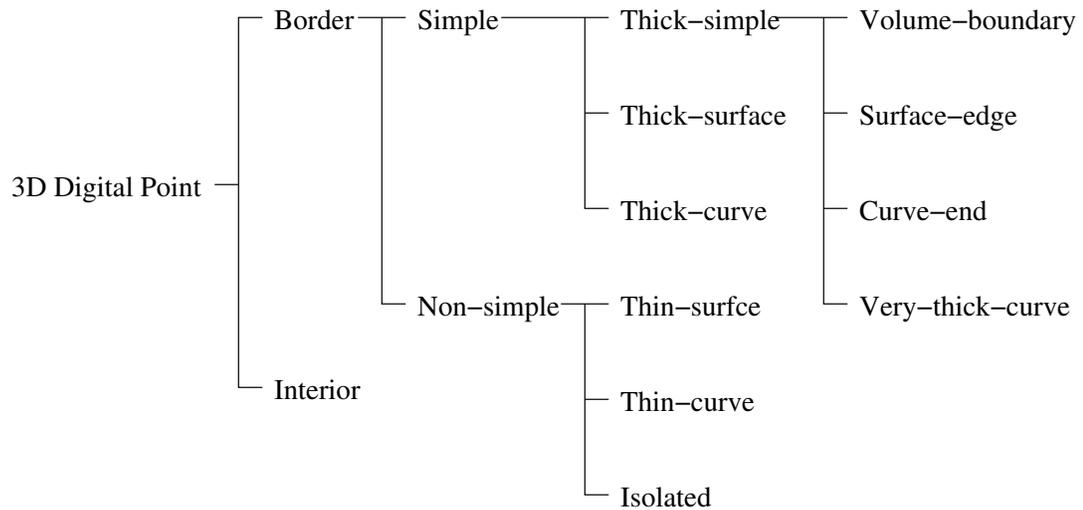


FIGURE 4.2: 3D digital point classification into a tree structure.

point of curve, junction point of curves, isolated point. In both works, surface and curve correspond to the thin surface and thin curve in our work. The point classification in [101] and [102] assumes the object is either originally a discrete surface or the surface skeleton of a 3D object. Based on the point classification, the discrete surface or the surface skeleton can be segmented into meaningful parts.

The major distinction between our work and the research above is that we proposed the notion of thick-simple point. Based on this notion, we can classify discrete surfaces into thin surface and thick surfaces, and curves into thin curves and thick curves. Here thick surface and thick curves refer to surfaces and curves that are at most two-point thick. The notion of thick-simple point is relevant to the fact that the very central discrete surface skeleton of a 3D object may be two-point thick at some places. It also helps to identify other geometrical features such as volume boundary, surface edges and curve ends. Based on the notion of thick-simple points, we can conveniently design surface and curve skeletonization algorithms that not only guarantee homotopy and thinness, but also ensure medialness and high level of shape preservation. In addition, the notion of thick-simple point also plays a critical role in our topology correction algorithm.

CHAPTER V

3D Skeletonization

Skeletonization is a process that reduces foreground regions in a binary image to a skeletal remnant while eliminating significant amount of foreground pixels (voxels). The skeleton of a 2D binary object is a set of 2D discrete curves. For a 3D binary image, *surface skeletonization* results in a discrete surface and *curve skeletonization* results in discrete curves. Skeletons have been widely used in computer vision, pattern recognition, image segmentation, computer graphics and image compression.

Although there is not a standard and precise definition of skeletonization, the following properties of skeletonization are commonly desired:

- *homotopy*: the skeleton should be homotopic (i.e. topologically equivalent) to the original object;
- *thinness*: the surface skeleton should be one point thick and the curve skeleton should be one point wide; and
- *medialness*: the skeleton should be centrally located within the object.

In this chapter, we propose a 3D surface skeletonization algorithm and a 3D curve skeletonization algorithm. In our neuroanatomical segmentation work, surface skeletonization results in data that plays a critical role in white matter localization and extraction. Variants of curve skeletonization are used for topology correction of white matter and other morphological image analysis tasks. Both surface skeletonization and curve skeletonization are based extensively upon the 3D digital point classification presented in the previous chapter.

There are two steps for surface skeletonization: thick-surface skeletonization and thin-surface skeletonization. The former results in a discrete surface of at most two voxels thick and the latter results a final thin surface skeleton of one voxel thick. We will present skeletonization algorithm in the order of thick-surface skeletonization, thin-surface skeletonization, and curve skeletonization.

5.1 Thick-surface Skeletonization

Thick-surface skeletonization of the set \mathcal{X} is a process that iteratively eliminates volume-boundary points relative to \mathcal{X} until no additional points can be eliminated. In each iteration, border points are first identified. Then simple points are extracted from border points and non-simple points are marked to prohibit elimination. Next, simple points are classified into thick-simple points and thin-simple points. Like non-simple points, thin-simple points are also prohibited from elimination. Among thick-simple points, volume-boundary points are recognized and are eliminated if they are still simple relative to \mathcal{X} at the moment of elimination. Whenever a point is eliminated, new border points in its neighborhood are identified as candidates for erosion in the next iteration.

Thick-surface skeletonization described above erodes voxels in the order of the \bar{n} -distance, where n is the adjacency of the set \mathcal{X} . In n -distance, the distance between any n -adjacent voxel is taken as unit distance. A variant of this algorithm, Chamfer-thick-surface skeletonization, erodes voxels in the order of chamfer distance which sets distances between adjacent voxels as follows. For any two voxels sharing a face, their chamfer distance is 3; for any two voxels sharing an edge, their chamfer distance is 4; and for any two voxels sharing only a point, their chamfer distance is 5. Chamfer distance provides closer approximation to Euclidean distance. For thick-surface skeletonization that erodes voxels in Chamfer distance, a map of distance to $\bar{\mathcal{X}}$ is first calculated for each point in \mathcal{X} . The main distinction of the two algorithms is the definition of the border points and interior points. We say a point $p \in \mathcal{X}' \subset \mathcal{X}$ is a border point relative to (rt) \mathcal{X}' with respect to (wrt) \mathcal{X} at distance d if p is a border point rt \mathcal{X}' and p 's distance from $\bar{\mathcal{X}}$ is d . If the distance of

$p \in \mathcal{X}'$ from $\overline{\mathcal{X}}$ is greater than d , then p is an interior point of \mathcal{X}' wrt \mathcal{X} at distance d . In short, we say that p is a border point at distance d or an interior point at distance d . The rest of the algorithm for Chamfer-thick-surface skeletonization is the same with regular thick-surface skeletonization. These two algorithms are presented in algorithm 3 and algorithm 4 respectively.

Both thick surface skeletonization algorithms remove a point from the object only when the point is a simple point, therefore the algorithms preserve topology after skeletonization. Meanwhile, points are removed in the order of either \bar{n} -distance or Chamfer distance from the complement of the object, therefore the algorithms also guarantee the medialness of the skeleton. Since the algorithms keep the thick-surface points and very-thick-curve points, the resultant skeleton may be of two voxels thick at some places. The resulted surface skeleton is hence referred to as *thick-surface skeleton* and can be further thinned into the thin-surface skeletonization algorithm described in the next section. Figure 5.1(b) shows the thick surface skeleton of a cerebral white matter object in terms of Chamfer distance.

5.2 Thin-surface Skeletonization

Thin-surface skeletonization further thins thick-surface skeleton by sequentially eroding thick-surface points, thick surface-edge points and thick curve-end points. First, it classifies all points in the thick-surface skeleton set \mathcal{X} into border points and interior points. Second, border points are classified into simple points and non-simple points. Third, simple points are classified into various types of thick-simple points and various types of thin-simple points. Thick-simple points include volume-boundary points (should be an empty set), surface-edge points, and very-thick-curve points. Thin-simple points include thick-surface points and thick-curve points.

After the point classification, thick-surface points and very-thick-curve points are first removed if they are still so at the moment of removal. After a thick-surface point is removed, points in its neighbor should be reclassified. The order in which thick-surface points are removed has a significant influence on the resultant shape of the surface skeleton.

Algorithm 3: Thick-surface skeletonization

Data: A set \mathcal{X}^0 in n -adjacency
 $\mathcal{X} \leftarrow \mathcal{X}^0$; /* \mathcal{X} denotes the eroded set in the process */
 Classify all points in \mathcal{X} into border points and interior points;
 $terminate \leftarrow false$;
repeat
 $terminate \leftarrow true$;
 Classify border points into simple points and non-simple points;
 Classify simple points into thick-simple points and thin-simple points;
 Classify thick-simple points into volume-boundary points and
 non-volume-boundary points;
 Let \mathcal{V} be the set of all volume-boundary points;
 $continue \leftarrow true$;
 repeat
 $continue \leftarrow false$;
 foreach point p in the set \mathcal{V} **do**
 if p is simple relative to \mathcal{X} **then**
 Remove p from \mathcal{X} ;
 Classify interior points in p 's neighbor $\mathcal{N}(p)$ into border points and
 interior points;
 $continue \leftarrow true$;
 $terminate \leftarrow false$
 until $continue = false$;
 Reset all non-interior points as border points;
until $terminate = true$;

Algorithm 4: Chamfer-thick-surface skeletonization

Data: A set \mathcal{X}^0 in n -adjacency
 $\mathcal{X} \leftarrow \mathcal{X}^0$; /* \mathcal{X} denotes the eroded set in the process */
 Calculate the Chamfer distance map;
 $d \leftarrow 3$;
 Classify all points in \mathcal{X} into border points and interior points at distance d ;
 $terminate \leftarrow false$;
repeat
 $terminate \leftarrow true$;
 Classify border points at distance d into simple points and non-simple points;
 Classify simple points into thick-simple points and thin-simple points;
 Classify thick-simple points into volume-boundary points and
 non-volume-boundary points;
 Let \mathcal{V} be the set of all volume-boundary points;
 $continue \leftarrow true$;
 repeat
 $continue \leftarrow false$;
 foreach point p in the set \mathcal{V} **do**
 if p is simple relative to \mathcal{X} **then**
 Remove p from \mathcal{X} ;
 Classify interior points at distance d in p 's neighbor $\mathcal{N}(p)$ into border
 points at distance $d + 1$ and interior points at distance $d + 1$;
 $continue \leftarrow true$;
 $terminate \leftarrow false$
 until $continue = false$;
 $d \leftarrow d + 1$;
 Reset all non-interior points as border points;
until $terminate = true$;

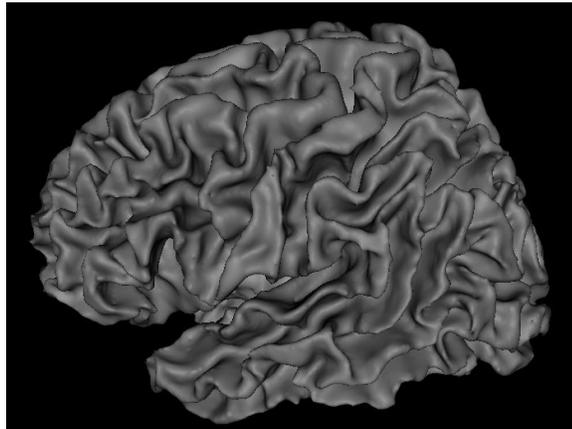
Whenever a thick-surface point is removed, we put its neighbors that are still thick-surface points in the highest priority for removal. Such a breath-first-search order of removal has a benefit in that it alleviates the jitter effect of the resulted surface skeleton. After removal of thick-surface points, thick surface-edge points of thin surfaces are thinned in the similar breath-first-search order. Finally, thick curve-end points are thinned. Thin-surface skeletonization is presented in algorithm 5. Figure 5.1(c) shows the thin surface skeleton of a cerebral white matter object.

Algorithm 5: Thin-surface skeletonization

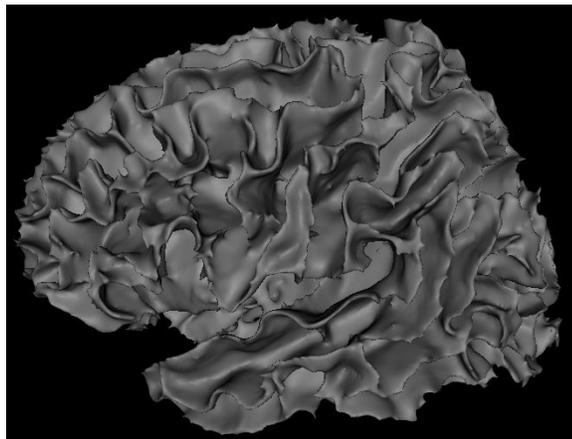
```

Data: A thick surface skeleton  $\mathcal{X}^0$ 
 $\mathcal{X} \leftarrow \mathcal{X}^0$ ;      /*  $\mathcal{X}$  is the eroded skeleton in the process */
/* Point classification                                     */
Classify all points in  $\mathcal{X}$  into border points and interior points;
Classify all border points into simple points and non-simple points;
Classify all simple points into volume-boundary points ( should be empty),
thick-surface points, surface-edge points, very-thick-curve points, thick-curve points,
and curve-end points. Let  $\mathcal{T}$  be the set of thick-surface points and very-thick-curve
points,  $\mathcal{E}$  be the set of surface-edge points, and  $\mathcal{C}$  be the set of the curve-end points;
/* Thin thick surfaces and very thick curves                */
foreach point  $p$  in the set  $\mathcal{T}$  do
  Put  $p$  in an empty queue  $Q$ ;
  while  $Q$  is not empty do
    Pop up a point  $q$  from the queue;
    if  $q$  is still a thick-surface point or a very-thick-curve point then
      Remove  $q$  from  $\mathcal{X}$ . Reclassify  $q$ 's neighbors that are in  $\mathcal{X}$  and enqueue
      new thick-surface points and very-thick-curve points;
/* Thin thick surface edge points of thin surfaces        */
foreach point  $p$  in the set  $\mathcal{E}$  do
  Put  $p$  in an empty queue  $Q$ ;
  while  $Q$  is not empty do
    Pop up a point  $q$  from the queue;
    if  $q$  is now actually a thick surface-edge points of thin surfaces then
      Remove  $q$  from  $\mathcal{X}$ ;
/* Thin thickcurveend points                             */
foreach point  $p$  in the set  $\mathcal{C}$  do
  if  $p$  is now actually a thick curve-end point then
    Remove  $p$  from  $\mathcal{X}$ ;

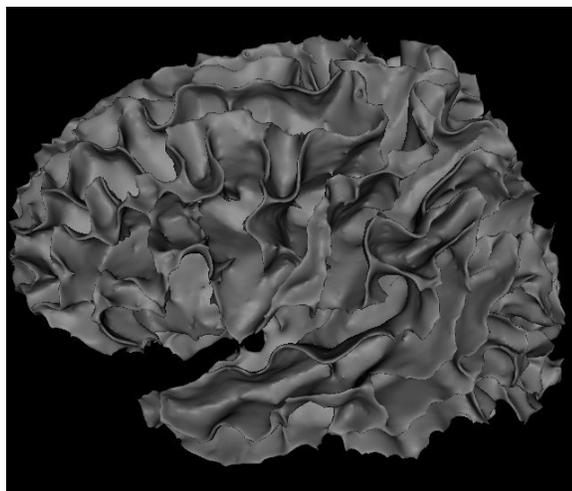
```



(a) Cerebral white matter



(b) Thick surface skeleton



(c) Thin surface skeleton

FIGURE 5.1: Surface skeletonization results

5.3 Curve Skeletonization

Curve skeletonization of a set \mathcal{X} iteratively removes thick-simple points except for curve-end points from \mathcal{X} resulting in discrete curves of one voxel wide. In each iteration, all border points of \mathcal{X} are first classified into simple points and non-simple points. Non-simple points are forbidden to be removed at the current iteration. Simple points are classified into thick-simple points and thin-simple points (thick-surface points and thick-curve points). Curve-end points are identified among thick-simple points. Then only thick-curve points and thick-simple that are not curve-end points are allowed to be removed at the current iteration. Whenever a point is removed, new border points in its neighbor are identified for processing in the next iteration. At the end of each iteration, any points that have not been removed and are not interior points are reset to be border points for processing in the next iteration. Curve skeletonization is presented in algorithm 6.

Unlike many other curve skeletonization algorithms, our algorithm does not require a preprocessing step of surface skeletonization before curve skeletonization. In other words, our curve skeletonization can be directly performed on the original 3D object. The resultant discrete curve is topologically equivalent to the original object and highly central within the original object.

When we limit the number of iterations that the curve skeletonization algorithm can perform with a scale s , the partial curve skeletonization produces output that can be seen as the erosion of the original object with topology and shape preservation at scale s . We will see that such a *topology-and-shape-preserving erosion (TSPE)* plays an important role for topology correction of white matter. It can also be used for morphological smoothing of surface-like objects. Let \mathcal{X} be 3D object with thin protrusions that we want to eliminate. We can first apply TSPE at scale s on \mathcal{X} ending up with \mathcal{X}' . Then we can obtain the smoothed result by dilating \mathcal{X}' such that the voxels in $\mathcal{X} \setminus \mathcal{X}'$ whose distances from \mathcal{X}' are less than or equal to s are added into \mathcal{X}' . The advantage of this smoothing approach over traditional morphological smoothing is that the shape noise can be removed while thin surface parts of the original object are preserved. Figure 5.2 and 5.3 respectively demonstrate the results of curve skeletonization and partial curve skeletonization of a cerebral WM object.

There are exceptions to the assertion that a thin surface skeleton is one voxel thick and a curve skeleton is one voxel wide. In these situations, the surface skeleton may be thicker than one voxel and the curve skeleton may be wide than one voxel. The skeleton cannot be further thinned, otherwise either the topology of the object will be changed or the shape of the skeleton will be significantly modified from the original object.

Algorithm 6: Curve skeletonization

Data: A set of points \mathcal{X}^0 in n -adjacency
 $\mathcal{X} \leftarrow \mathcal{X}^0$; /* \mathcal{X} denotes the eroded set in the process */
 Classify all points in \mathcal{X} into border points and interior points;
 $terminate \leftarrow false$;
repeat
 $terminate \leftarrow true$;
 Classify border points into simple points and non-simple points;
 Classify simple points into thick-simple points and thin-simple points;
 Identify curve-end points among thick-simple points and classify thin-simple points into thick-surface points and thick-curve points;
 Let \mathcal{T} be the set of all thick-curve points and thick-simple points except for curve-end points;
 $continue \leftarrow true$;
 repeat
 $continue \leftarrow false$;
 foreach point p in the set \mathcal{T} **do**
 if p is still simple relative to \mathcal{X} **then**
 Remove p from \mathcal{X} ;
 Classify interior points in p 's neighbor $\mathcal{N}(p)$ into border points and interior points;
 $continue \leftarrow true$;
 $terminate \leftarrow false$
 until $continue = false$;
 Reset all non-interior points as border points;
until $terminate = true$;

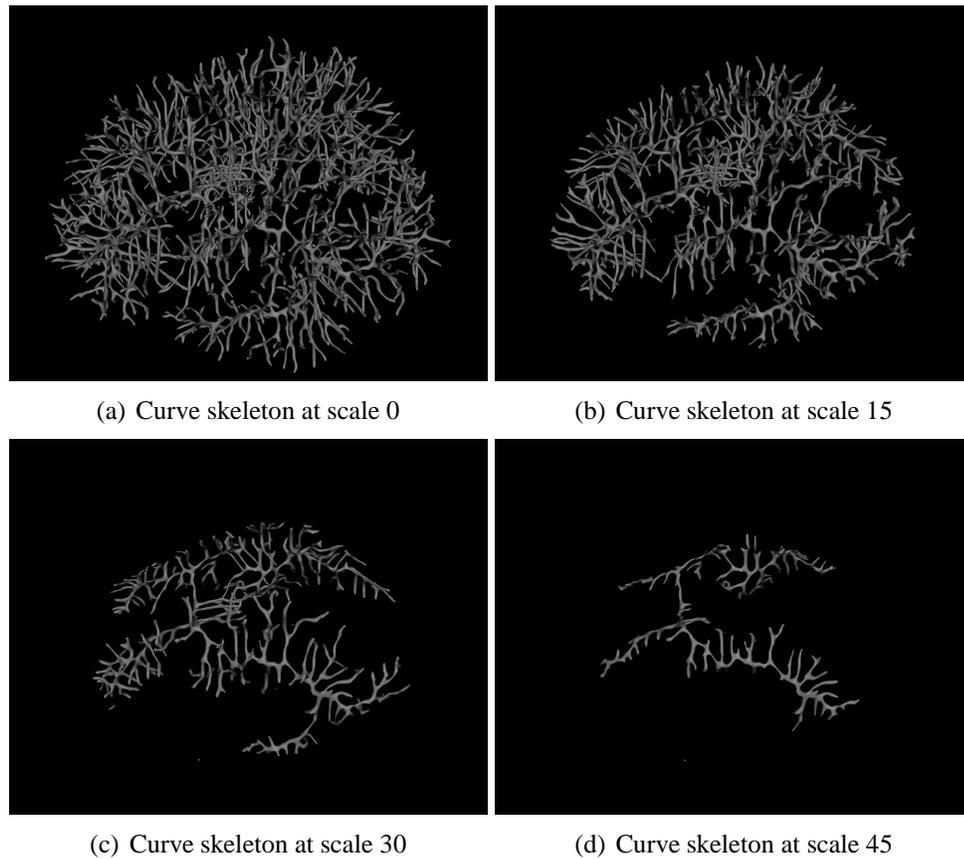
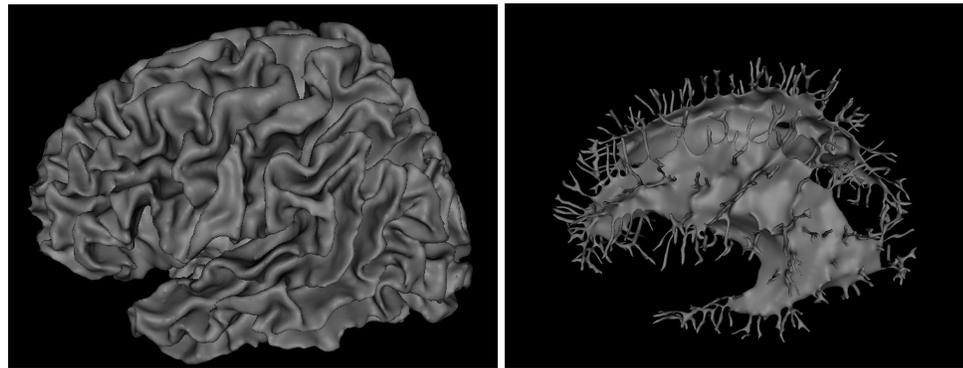


FIGURE 5.2: Curve skeletonization results. For a curve skeleton shown at scale s , all curve skeleton points at scale less than s is not shown. All curve skeletons are dilated for visual inspection.

5.4 Related Work

Several other works [103, 104, 105] also apply distance-ordered homotopic thinning for surface skeletonization of 3D objects. The difference between their work and ours is that our skeletonization is based on a more systematic point classification in which a new notion called thick-simple points are introduced. Based on such a framework of point classification, we perform thinning of thick-surface skeleton in a breath-first-searching order such that the resultant surface skeleton adheres to the shape of the original object to a higher extent and jitter effect is largely alleviated. For curve skeletonization, our algorithm works directly on the original 3D object instead of a two-stage manner of surface skeletonization



(a) Cerebral WM matter

(b) Partial curve skeleton at scale 15. Curve skeleton points at scale less than 15 is not shown

FIGURE 5.3: Partial curve skeletonization results.

followed by curve skeletonization [105, 106]. Based on our systematic point classification, this one-stage curve skeletonization results in curve skeletons that are central within the original object to a higher extent than the two-stage methods. In addition, partially running the one-stage curve skeletonization algorithm gives rise to topology-and-shape-preversing erosion that can be applied for other morphological image analysis tasks such as topology correction and smoothing of surface-like objects.

CHAPTER VI

Cell Complex Based Morphometric Analysis

In this chapter, we present a cell complex based morphometric analysis (CCBMA) method. Cell complex is a set of interconnected polyhedra, polygons, segments, and points. CCBMA first transforms a 3D binary object into a cell complex and then simplifies the complex such that it only consists of segments and points without removing any points in the complex. Meanwhile, meaningful metrics on each segment can be calculated that provide various geometrical information on the segment with respect to the original object. In our work on neuroanatomical segmentation, CCBMA plays a critical role for white matter localization and extraction. It may also be applied for many other problem solving in computer vision and image understanding.

This chapter starts with the motivation of CCBMA. Then the algorithm of CCBMA is presented in the subsequent sections. The input for cell complex analysis can be any 3D object but we use the thin-surface skeleton of the white matter object as input in our work for white matter localization and extraction. The application of CCBMA in our neuroanatomical segmentation will be presented in the next chapter.

6.1 Motivation

Since the 1960's, skeletonization of 2D object has been widely used for various tasks in computer vision, pattern recognition, image segmentation, and image compression. This is due to the capability of skeletonization to reduce the dimensionality of the original object

while maintaining the information on the extent and connectivity of the object. To be more specific, after skeletonization on 2D regions, we obtain curve skeletons in which for each point we have a metric that measures the shortest distance of the point from the boundary of original regions.

In 3D, however, things become much more complicated. After both surface skeletonization and curve skeletonization, we can also obtain a metric that measures for each point in the skeleton the shortest distance of the point from the boundary of the original 3D object. This metric is very useful when we are working with surface skeletons. In many cases, however, what we really need is further skeletonization into curve skeletons, since the metric *distance-from-boundary* is only marginally useful unless the original object is a tubular structure. The reason is that the metric provides little information on the wideness and connectivity of the original object. Cell complex analysis is hence motivated by pursuit of metrics that provides meaningful information of wideness and connectivity of the 3D object while reducing the dimensionality of the object.

6.2 Cell Complex

CCBMA is essentially a series of transformation on a space called *cell complex*. A cell complex is a topological space composed of points, segments, polygons, polyhedrons and the generation to any dimension of polygons in two dimensions. The generation of polygon to any dimension is termed *polytope*. Here, we are following the definition of cell complex given in [107]. A more abstract definition is given in [108].

An n -dimensional polytope is bounded by a number of $(n - 1)$ -dimensional faces. Each pair of $(n - 2)$ -faces meet at an $(n - 2)$ -dimensional face, and so on. A n -dimensional face is also a n -dimensional polytope. A 3-dimensional face is called a *cell*, a 1-dimensional face is called an *edge*, a 0-dimensional face is called a *vertex*, and a 2-dimensional face is just called a *face*. Note that *cell* can also generally denote a polytope at any dimension. A precise definition is omitted here for simplicity. In this thesis, we only deal with polytopes up to 3-dimension (i.e. we only deal with polyhedrons, polygons, segments, and points).

Given the definition of polytopes (cells) and faces, we give a formal definition of *cell complex* [107].

Definition 6.2.1. Cell complex A cell complex or simply complex in R^d is a set \mathcal{C} of polytopes (called cells) in R^d satisfying two conditions: (1) Every face of a cell is a cell in \mathcal{C} , and (2) two cells in \mathcal{C} either do not intersect, or their intersection is a cell of smaller dimension which is their common face of maximal dimension.

One special kind of polytope is a *convex polytope*, which is the convex hull of a finite set of points. In the rest of thesis, we mean convex polytope whenever we refer to polytope. If the maximal dimension of the constituting cells in a cell complex is k then the complex is called a k -complex. Our cell complex analysis can be seen as a process of transforming a 3-complex to a 1-complex. First, we need to construct a 3-complex from a 3D object.

6.3 Cell Complex Construction

In this section, we describe the procedure for constructing a cell complex from a 3D object \mathcal{X} in 26-adjacency. Cell complex construction for other adjacencies is a similar procedure to that for 26-adjacency and the cell complex transformation algorithms are same for all adjacencies. We only deal with 26-adjacency in this thesis.

Given a set of voxels \mathcal{X} in 26-adjacency, we first create a set of vertices \mathcal{V} each corresponding to a voxel in \mathcal{X} . In the following, whenever we say a point, a vertex or a voxel, we refer to the same thing without distinction. Second, we create an edge connecting every two vertices if they are 26-adjacent. Third, we create a triangle for every three distinct edges if they share three distinct vertices. Fourth, we create a tetrahedron for every four distinct triangles if they share four distinct edges.

By now, we have a set of polytopes up to 3-dimension. Even if the input \mathcal{X} is a surface skeleton, we may still have 3-dimensional polytope. At this point, the set of polytope is probably not a cell complex yet and what we do next is to perform following operations on the set to create a cell complex.

In a 3D lattice where each voxel is a grid point, we can identify each cube bounded by 8 voxels. All tetrahedra that locate within the same cube are contracted to a single polyhedron. All triangles that are in the same cube and within the same plane are also contracted into a single polygon. This simple procedure will finally result in a cell complex.

6.4 Generating 2-complex

Given a 3-complex \mathcal{C} , we perform cell complex analysis by first transforming the 3-complex to a 2-complex, as presented in algorithm 7. This transformation can be seen as a “surface skeletonization” procedure in the context of cell complex. The result is a 2-complex that only consists of polygons, segments, and points while all polyhedra are eliminated. A central rule for polyhedron elimination is that only a border polyhedron can be removed at any time. A *border polyhedron* has one 2-face that is shared by no other polyhedron. To eliminate a polyhedron, we can simply remove one of its 2-faces that is incident with only one polyhedron. Note that removal of a 2-face from the complex does not involve removal of any 1-face or 0-face of the complex. In other words, all segments and points remain in the complex after polyhedra elimination. A 2-face that is incident with only one polyhedron is called *simple face* because removal of the face and the incident polyhedron does not change the topology of the complex.

Algorithm 7: 3-complex to 2-complex transformation

Data: A 3-complex \mathcal{C}

Initialize the depth of all 2-faces to be 0;

Insert simple faces into an empty min-priority queue Q ;

while Q is not empty **do**

Pop up a 2-face f from Q ;

if f is still a simple face **then**

Remove f and its associated polyhedron c from the complex;

foreach 2-face $f' \neq f$ that was bounding c **do**

$depth(f') \leftarrow depth(f) + d(f, f')$;

Insert f' into Q using $depth(f')$ as the key if it is a simple face;

Similar to the distance-ordered thinning for 3D surface skeletonization described in the previous chapter, we also iteratively eliminate polyhedra and simple faces in an ascending order of distance from the outside of the complex. This iterative elimination is implemented utilizing a min-priority queue Q in which each element is a simple face to be removed with the key being the *depth* of simple faces from the outside of the complex. Initially, all 2-faces have their depth reset as 0 and simple faces are inserted into Q . Then the algorithm iteratively pops up a 2-face f from the queue and performs polyhedron elimination if f is still a simple face. Whenever a simple 2-face f and its associated polyhedron c are removed from the complex, the metric depth of remaining 2-faces bounding c are updated and new simple faces are inserted into the queue. For any remaining 2-face f' that was originally bounding c , its depth is updated as $depth(f') = depth(f) + d(f, f')$, where $depth(f)$ denotes the depth of f from the outside of the original complex and $d(f, f')$ denotes the distance between the center of the 2-faces f and f' . When the queue is empty, all polyhedra have been eliminated and there is no simple 2-faces anymore in the new complex. At the end of the algorithm, each remaining 2-face f has been assigned with a depth metric.

6.5 Generating 1-complex

Given a 2-complex generated in the above procedure, we further reduce the dimension of the complex generating a new 1-complex composed of only segments and points, as presented in algorithm 8. This procedure is similar to the above one except that here we recognize and remove *border polygons* and *simple edges*. A border polygon is a polygon in the complex that has a 1-face (an edge) that is shared with no other polygons; an edge that is incident with only one polygon is called a simple edge. We thereby transform a 2-complex to a new 1-complex by iteratively removing simple edges and their associated polygons. Apparently, this procedure also preserves the topology of the original 2-complex.

Like the procedure for transforming a 3-complex to a 2-complex, a min-priority queue is also used for ordered elimination of border polygons and simple edges. Here, each element in the queue Q is a 1-face and has to be a simple edge when it is inserted into

the queue. Whenever a 1-face is popped up from the queue for removal, it has to be still a simple edge. We propose two elimination order embodied in the definition of the key of the min-priority queue. We define three metrics for each 1-face in the complex: *radius*, *wideness*, and *connectivity*. Initially, all 1-faces in the 2-complex have their radius and connectivity reset to be 0. Whenever a simple edge e and its associated border polygon f are removed, the metric radius and connectivity of remaining 1-faces bounding f are updated and new simple edges are inserted into the queue. For any remaining 1-face e' that was originally bounding f , the radius and connectivity of e' are updated as $radius(e') = \max(radius(e'), radius(e) + d(e, e'))$ and $connectivity(e') = connectivity(e') + connectivity(e) + d(e, e')$, where $radius(e)$ and $connectivity(e)$ denote the radius and connectivity of e respectively and $d(e, e')$ denotes the distance between the centers of two 1-faces. We can use the radius or the connectivity as the key for the min-priority queue and end up with two different orders of simple edge removal. At the end of the algorithm, all polygons have been removed and there are no simple edges anymore in the new 1-complex. The algorithm also preserves the topology of the complex.

Whenever a simple edge e and the associated border polygon f are removed, we also record a “parent-child” relationship between any remaining 1-face e' that was bounding f and the newly removed 1-face e . When the connectivity is used for the key of the min-priority queue, we need to adjust the radius of 1-faces in the resultant 1-complex as follows. For any 1-face e in the 1-complex, if e has two or more child 1-faces in the intermediate 2-complex, then the wideness of e is $wideness(e) = (radius(e_1) + d(e, e_1) + radius(e_2) + d(e, e_2))/2$, where e_1 and e_2 are the two child 1-faces of e that have greatest radii; otherwise $wideness(e) = radius(e)$.

With the above two algorithms, we can transform a 3-complex into a 1-complex that significantly reduces the complexity of the representation of a 3D object. Meanwhile, each remaining 1-face (segment) has been assigned three metrics: radius, wideness, and connectivity. They give information on how wide and how strongly connected the original 3D object is at the place where e locates. If desired, we can also transmit the depth of a polygon in the intermediate 2-complex to the bounding segments that still remain in the final 1-complex. Therefore, we end up with a simple form of representation of the original 3D object while largely preserving structural, geometrical and morphometrical information of

the original object. It is demonstrated in the next chapter that this technique of cell complex analysis can be applied for robust neuroanatomical segmentation.

Algorithm 8: 2-complex to 1-complex transformation

Data: A 2-complex \mathcal{C} and an *option* on whether using radius or connectivity as the key of the min-priority queue

Initialize the radius and connectivity of all 1-faces to be 0;

Insert simple edges into an empty min-priority queue Q ;

while Q is not empty **do**

 Pop up a 1-face e from Q ;

if e is still a simple edge **then**

 Remove e and its associated polygon f from the complex;

foreach 1-face $e' \neq e$ that was bounding f **do**

$radius(e') \leftarrow \max(radius(e'), radius(e) + d(e, e'))$;

$connectivity(e') \leftarrow connectivity(e') + connectivity(e) + d(e, e')$;

 Set e as the child of e' ;

if e' is a simple edge **then**

if the option is using radius as the key **then**

 Insert e' into Q using $radius(e')$ or $connectivity(e')$ according to the algorithm option;

foreach 1-face e in the new 1-complex **do**

if e has two or more children **then**

 Find the two children of e that have the largest radius: e_1 and e_2 ;

$wideness(e) \leftarrow (radius(e_1) + d(e, e_1) + radius(e_2) + d(e, e_2))/2$;

else

$wideness(e) \leftarrow radius(e)$;

CHAPTER VII

Topology Correction

Although human cortices are highly convoluted and fold in different ways, the cortical surface should be topologically equivalent to a sphere if the opening at the brain stem is artificially closed. Topology correctness is an important requirement in the cortical surface reconstruction process while topology defects or errors, mainly in the form of handles, may arise in MR image segmentation due to various image artifacts. In this chapter, we present a volume-based and multiscale morphological approach for topology correction of white matter [109], based on which topologically correct gray matter can be produced and the topologically correct cortical surfaces can be generated with the existing isosurfaces algorithm [110]. Extraction of topologically correct cortex and reconstruction of cortical surfaces will be presented in the next chapter.

7.1 Topology Correction Preserving Surface-likeness

Our method for topology correction is mainly motivated by the observation about the surface-likeness of white matter (WM) and gray matter (GM). The surface-likeness of cerebral WM is apparent by comparing the cerebral WM and its surface skeleton, as shown in figure 7.1. To preserve the shape of surface-like objects, the cost of handle cut in blue in figure 7.2 should be greater than that of a cut in red because the object is “wider”, in other words more like a surface, at the blue part than at the red part, although the object is thinner at the blue part. Similarly, we should fill the tunnel (i.e. cut the associate background

handle) in the right object in figure 7.2 instead of cutting the foreground handle. In this chapter, we describe a morphological topology correction (MTC) filter that cuts all handles at a specific scale of cost evaluated in terms of the wideness of the object.

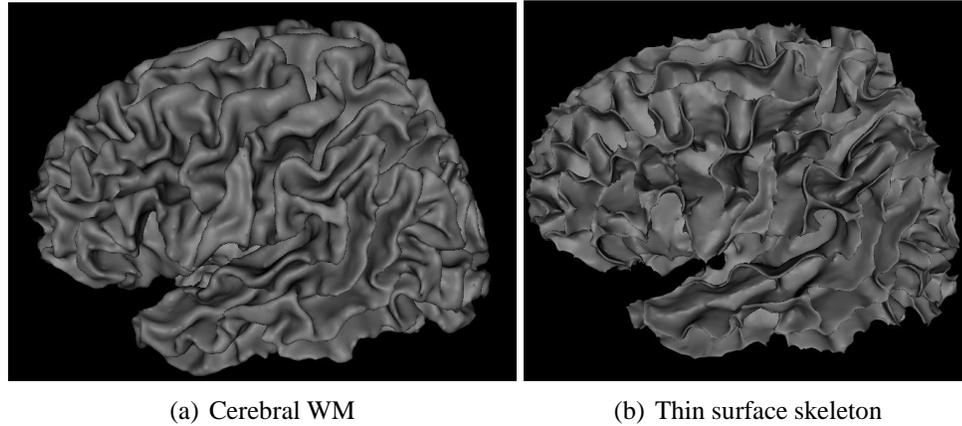


FIGURE 7.1: Surface-likeness of cerebral white matter

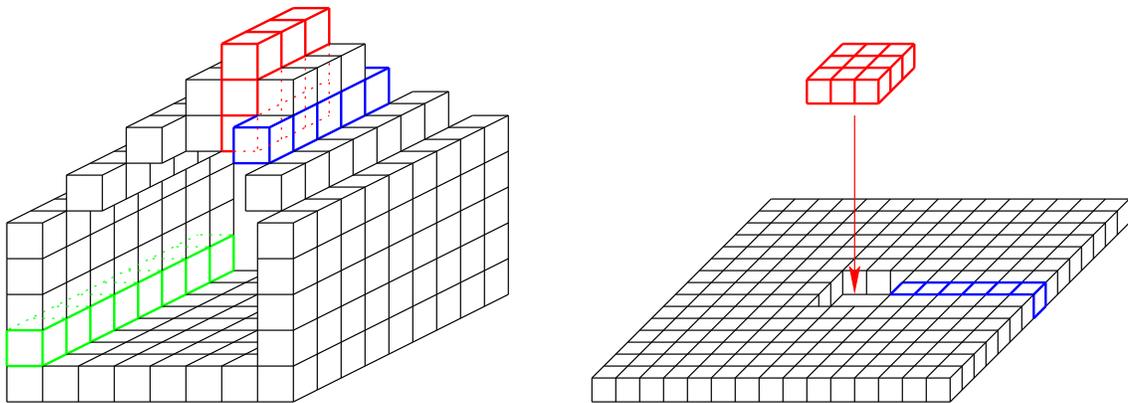


FIGURE 7.2: Topology correction adapted to the surface-likeness of objects

7.2 Three-fold Topology Correction

Traditionally, volume-based methods of topology correction of the white matter are two-fold in that there are two basic types of tunnel filling: filling the tunnels of the white

matter or filling the tunnels of the complement of the white matter. Note that the second is equivalent to cutting the handles of the white matter. Whenever a tunnel of an object (the white matter or its complement) is filled, the points used to fill the tunnel are always from the complement of the object.

Brain MRI segmentation, however, is usually able to separate the grey matter from the rest of the complement of the white matter and there is certain information provided by the prior segmentation that is not used by the traditional two-fold methods. Let three sets \mathcal{W} , \mathcal{G} and \mathcal{BG} respectively represent WM, GM, and background(including CSF) produced by the prior brain tissue segmentation. When a tunnel of the white matter is filled in the two-fold method, there are generally three possibilities on the composition of the points used to fill the tunnel: the points are only from \mathcal{G} ; the points are only from \mathcal{BG} ; or the points are from both \mathcal{G} and \mathcal{BG} . Considering the radiological property of T1-weighted brain MRI (the average gray level of air, cerebrospinal fluid, grey matter, and white matter are in the ascending order) and the layered organization of WM, GM, and background regions, it is reasonable to assume that the points from \mathcal{BG} have less credibility of actually belonging to the white matter than points from \mathcal{G} . In other words, it is reasonable to prefer to use exclusively the points from \mathcal{G} to fill the tunnels in \mathcal{W} . Points in \mathcal{BG} are used to fill a tunnel in \mathcal{W} only when necessary (i.e. when the tunnel is passed through by one or more handles in \mathcal{BG}). Based on this rationale, a three-fold topology correction method has been invented that involves three types of tunnel filling: filling the tunnels of the union of \mathcal{G} and \mathcal{BG} using points from \mathcal{W} (i.e. cutting the handles in \mathcal{W}); filling tunnels of \mathcal{W} using points from \mathcal{G} ; and filling tunnels of the union of \mathcal{G} and \mathcal{W} using points from \mathcal{BG} . Note that the third type creates more chances for the second type tunnel filling.

7.3 Multiscale Topology Correction

Our algorithm gradually eliminates topology handles with increasing scales of cost. The core algorithm is a morphological topology correction (MTC) filter that fills tunnels in an object (i.e, cuts handles in the complement of the object) at the specific scale of

cost. At each of a sequence of increasing scales (e.g., scales $s = 0, 1, 2$, etc.), MTC is successively applied for correcting topology of WM. MTC associates a larger correction cost to topology corrections (e.g., cuts) at locations having a larger surface-like property (i.e. wider regions).

The input to the topology correction algorithm is the pre-segmentation result in three sets \mathcal{W} , \mathcal{G} and \mathcal{BG} representing WM, GM, and background (including CSF) respectively. Necessary preprocessing operations are performed to enforce that \mathcal{W} forms only one connected component and contains no cavities and \mathcal{W} should not be connected to \mathcal{BG} . Algorithm 9 illustrates the topology correction of the white matter (\mathcal{W}) in a multiscale manner starting from scale s of 0. At the end, \mathcal{W} should be homotopic to a ball. An object is homotopic to a ball if we can keep removing simple points from the object ending up with a single point. The details of MTC is described in the following section and presented in algorithm 10.

Algorithm 9: Topology correction of white matter

Data: presegmentation result in three sets \mathcal{W} , \mathcal{G} and \mathcal{BG}

Result: Topologically corrected white matter set

$s \leftarrow 0$;

while true do

Perform MTC at scale s to fill tunnels of $\mathcal{G} \cup \mathcal{BG}$ (i.e. the complement of \mathcal{W}) using points from \mathcal{W} . The points used to fill the tunnels are moved from \mathcal{W} to \mathcal{G} . Put it in other way, this step cuts the handles of \mathcal{W} at scale s ;

if \mathcal{W} is homotopic to a ball then

└ Terminate the loop;

Perform MTC at scale s to fill the tunnels of $\mathcal{G} \cup \mathcal{W}$ (i.e. the complement of \mathcal{BG}) using points from \mathcal{BG} . The points used to fill the tunnel are moved from \mathcal{BG} to \mathcal{G} . In other words, this step cuts the handles of \mathcal{BG} at scale s ;

Perform MTC at scale s to fill the tunnels of \mathcal{W} using points from \mathcal{G} . The points used to fill the tunnels are moved from \mathcal{G} to \mathcal{W} ;

Fill cavities (if any) in \mathcal{W} . In rare situations new cavities may be created in \mathcal{W} by the above step;

if \mathcal{W} is homotopic to a ball then

└ Terminate the loop;

└ $s \leftarrow s + 1$;

7.4 Morphological Topology Correction (MTC) Filter

A typical application of the MTC filter is to fill the tunnels of object \mathcal{X} using points from a set $\mathcal{M} \subseteq \overline{\mathcal{X}}$ at a specific scale s . Put it in another way, the MTC filter cuts the handles of the complement of \mathcal{X} by moving a subset of points from \mathcal{M} to \mathcal{X} .

A minimal set of connected points $F \subseteq \mathcal{M}$ used to fill a tunnel (or multiple tunnels simultaneously) is referred to as a *fill* of \mathcal{X} . It can be also referred to as a *cut* of $\overline{\mathcal{X}}$ in the handle-cutting perspective. In other words, a fill of tunnels in \mathcal{X} is a cut of handles in the complement of \mathcal{X} . A characteristic of a fill F of \mathcal{X} is that it contains no multisimple points relative to $\overline{\mathcal{X}} \setminus F$, that is, adding any point in F into $\overline{\mathcal{X}} \setminus F$ will create at least a new handle in $\overline{\mathcal{X}} \setminus F$. This implies that a fill must be minimal in that removal of any proper subset of the fill will lead to filling fewer or no tunnels.

Locating fills at a specific scale depends a morphological operation called iterative shape-and-topology-preserving geodesic dilation (ISTPGD). Iterative shape-and-topology preserving dilation (ISTPD) of \mathcal{X} at scale s corresponds to iterative shape-and-topology preserving erosion (ISTPE) of its complement $\overline{\mathcal{X}}$ at the same scale while the latter can be seen as the partial curve skeletonization in which the outer loop is performed only $s/2$ iterations (see algorithm 6 in chapter VI). To fill tunnels of \mathcal{X} at scale s using points in \mathcal{M} , MTC first performs ISTPGD on \mathcal{X} with respect to \mathcal{M} at scale s . ISTPGD behaves as the same way as ISTPD except for an additional condition that any point added to \mathcal{X} (i.e. eroded from $\overline{\mathcal{X}}$) must be in the set \mathcal{M} . Let \mathcal{X}' be the dilated set of \mathcal{X} and $\mathcal{Y} = \overline{\mathcal{X}'}$ be the eroded set of $\overline{\mathcal{X}}$. We then recognize all thin-curve points relative to \mathcal{Y} . Among all thin-curve points, there may be some handle points relative to \mathcal{Y} . Removal of any handle point relative to \mathcal{Y} breaks one or more handles in \mathcal{Y} . Since \mathcal{Y} is topologically equivalent to $\overline{\mathcal{X}}$, we can trace a cut of $\overline{\mathcal{X}}$ from each handle point relative to \mathcal{Y} . Furthermore, the mechanism of ISTPGD determines that $\overline{\mathcal{X}}$ is approximately s -voxels wide at the place where each handle point locates. Therefore, we can trace a cut of $\overline{\mathcal{X}}$ (i.e. a fill of \mathcal{X}) approximately s voxels wide, from each handle point. In the next section, we describe a procedure of geodesic dilation for extracting all fills of \mathcal{X} at scale s starting from handle points relative to \mathcal{Y} .

7.5 Geodesic Dilation With Topology Control

After ISTPGD is applied on $\overline{\mathcal{X}}$ with respect to \mathcal{M} at a specific scale resulting in eroded object \mathcal{Y} , where $\mathcal{Y} \subseteq \overline{\mathcal{X}}$, the set of all thin-curve points relative to \mathcal{Y} are recognized and denoted as \mathcal{R} and referred to as the *residual set*. Let the *body set* be $\mathcal{B} = \mathcal{Y} \setminus \mathcal{R}$. In order to extract fills of tunnels in \mathcal{X} at scale s (i.e. cuts of handles in $\overline{\mathcal{X}}$ at scale s), we first extract cuts of handles in \mathcal{Y} at scale s , based on which the final fills at scale s are extracted.

Although removal of one handle point in \mathcal{R} from \mathcal{Y} breaks one or more handles in \mathcal{Y} at scale s , simultaneous removal of all handle points in \mathcal{R} from \mathcal{Y} may be not necessary to break all handles in \mathcal{Y} at scale s , may break \mathcal{Y} into several connected components, and even may create new handles. For example, in the object (in 6-adjacency) in figure 7.3, points 7 to point 30 are all identified as residual point at first. But removal of the residual point 7 and 30 simultaneously creates a new tunnel in the object. Our solution to this exception is that any thin-curve point is taken as a residual point only if $T_{\overline{m}}(p, \overline{\mathcal{B}}) = 1$. In this way, it is guaranteed that removal of all residual points will not create new handles. To find the minimal set of handle points for cutting handles in \mathcal{Y} and the final fills of \mathcal{X} at scale s , we propose a procedure including three stages of geodesic dilations with topology control. These three stage of dilations together with the prior topology correction steps are illustrated in figure 7.4.

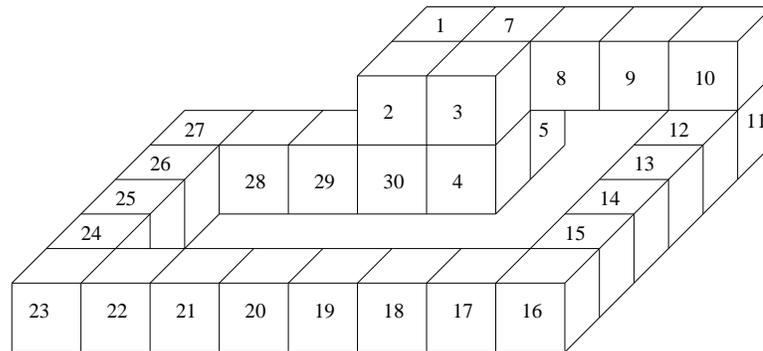


FIGURE 7.3: An example of residual points.

The following three dilation stages iterate in the same manner. The dilation in every stage involves a seed set \mathcal{S} and a condition set \mathcal{C} . In each iteration of every stage, any points

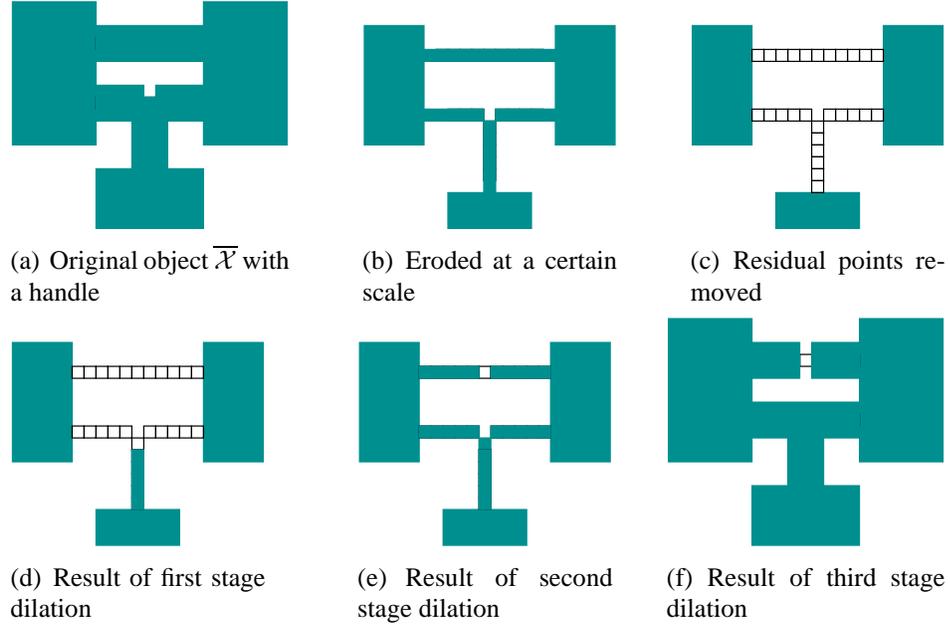


FIGURE 7.4: Illustration of all stages of topology correction

in \mathcal{C} that are adjacent to \mathcal{S} are marked first and then are moved from \mathcal{C} to \mathcal{S} if they satisfy some additional conditions. The iteration terminates in each stage if no more points can be moved.

In stage 1, \mathcal{B} is the seed set and \mathcal{R} is the condition set. The task of dilation in this stage is to recognize and recover the finger points from the residual set to the body set. In each iteration, a marked point p is identified as a finger point and recovered if $T_m(p, \mathcal{R}^+) \leq 1$, and p is m -adjacent to a body component \mathcal{B}_i with $degree(\mathcal{B}_i) = 1$ and is multisimple relative to \mathcal{B} at the moment of recovery. Here $degree(\mathcal{B}_i)$ denotes the *degree* of a body component \mathcal{B}_i and is defined as $degree(\mathcal{B}_i) = \sum_{j=1}^{N_{\mathcal{R}}} T_m(r_j, \mathcal{B}_i)$, where r_j represents a thin-curve point in \mathcal{R} and $N_{\mathcal{R}}$ represents the total number of points in \mathcal{R} . $degree(\mathcal{B}_i)$ can be seen as the number of ports at which \mathcal{B}_i is connected to \mathcal{R} . \mathcal{R}^+ refers to the union of \mathcal{R} and all body components with degree greater than 1. Whenever a point is recovered, it is moved out of \mathcal{R} to \mathcal{B} .

In stage 2, a minimal set of handle points $\mathcal{R}^* \in \mathcal{R}$ is identified and $\mathcal{R} \setminus \mathcal{R}^*$ is recovered from \mathcal{R} to \mathcal{B} . When multiple handle points relative to \mathcal{Y} exist whose removal from \mathcal{Y} break the same handle in \mathcal{Y} , the dilation procedure only chooses the handle point in the middle and

recover others. In this stage, the condition set is \mathcal{R} and the seed set is \mathcal{B} . In each iteration of stage 2, a marked point is recovered if it is multisimple relative to \mathcal{B} at the moment of recovery.

In stage 3, the seed set is \mathcal{B} while the condition set is $\overline{\mathcal{X}} \setminus \mathcal{B}$. A marked point is recovered if it is multisimple relative to \mathcal{B} at the moment of recovery. Eventually, the object $\overline{\mathcal{X}}$ is recovered as the newest body set \mathcal{B} except for the points in the cuts that cut the handles of $\overline{\mathcal{X}}$ (i.e. the fills that fill the tunnels of \mathcal{X}).

Since a recovered point has to be multisimple relative to \mathcal{B} , the number of handles in \mathcal{B} is not changed. Merging of body components may happen and is desirable. In rare situations, however, there might be more than one connected component in the resultant body set and we only keep the largest component in the case of using points from WM to fill tunnels of the union of GM and background (i.e. cutting handles in WM).

7.6 Related Work

In order to generate topologically correct cortical surfaces, there have been two basic approaches proposed in the literature. The first approach imposes topology constraints in the procedure of segmentation [111, 112], typically in deformable model based methods, starting with a model of the cortical surface with correct topology. The topology of the model is preserved in the deformation procedure of the segmentation. These methods share the disadvantages of the original segmentation methods and may lead to unpredictable morphometrical states. Retrospective topology correction is the mainstream approach in the literature for extracting topologically correct cortical surfaces. They take as input the result of the pre-segmentation and eliminate all topology defects resulting in cortical surfaces homotopic to a sphere.

Topology correction methods can be classified into two types: surface-based and volume-based methods. Surface-based methods first extract the cortical surfaces represented as a triangle mesh and perform topology correction on the surface. Volume-based methods first

Algorithm 10: Morphological topology correction

Data: Two sets \mathcal{X} and $\mathcal{M} \in \overline{\mathcal{X}}$ in m -adjacency and a specific scale s .

Result: A new set \mathcal{X}' with tunnels at scale s eliminated.

/ The task of the algorithm is to fill tunnels in \mathcal{X} using points from \mathcal{M} at scale s */*

Perform shape-and-topology-preserving geodesic dilation of \mathcal{X} with respect to \mathcal{M} .
 Let \mathcal{Y} be the complement of the dilated set;
 Recognize thin-curve points $\mathcal{R} \in \mathcal{Y}$. Let the body set $\mathcal{B} = \mathcal{Y} \setminus \mathcal{R}$;

foreach point p in \mathcal{R} **do**

- if** $T_m(p, \overline{\mathcal{B}}) > 1$ (m is the adjacency of \mathcal{Y}) **then**
 - Move p from \mathcal{R} into \mathcal{B} ;

/ First stage dilation */*

$terminate \leftarrow false$;

while $terminate = false$ **do**

- $terminate \leftarrow true$;
- Mark all points in \mathcal{R} that are m -adjacent to \mathcal{B} ;
- foreach** marked point p **do**
 - if** p is m -adjacent to a body component \mathcal{B}_i with $degree(\mathcal{B}_i) = 1$, p is multisimple relative to \mathcal{B} , and $T_m(p, \mathcal{R}^+) \leq 1$ **then**
 - Move p from \mathcal{R} to \mathcal{B} ;
 - $terminate \leftarrow false$;

/ Second stage dilation */*

$terminate \leftarrow false$;

while $terminate = false$ **do**

- $terminate \leftarrow true$;
- Mark all points in \mathcal{R} that are m -adjacent to \mathcal{B} ;
- foreach** marked point p **do**
 - if** p is multisimple relative to \mathcal{B} **then**
 - Move p from \mathcal{R} to \mathcal{B} ;
 - $terminate \leftarrow false$;

/ Third stage dilation */*

$terminate \leftarrow false$;

while $terminate = false$ **do**

- $terminate \leftarrow true$;
- Mark all points in $\mathcal{C} = \overline{\mathcal{X}} \setminus \mathcal{B}$ that are m -adjacent to \mathcal{B} ;
- foreach** marked point p **do**
 - if** p is multisimple relative to \mathcal{B} **then**
 - Move p from \mathcal{C} to \mathcal{B} ;
 - $terminate \leftarrow false$;

$\mathcal{X}' \leftarrow \mathcal{B}$;

perform topology correction on the image volume and then extract the isosurface of the topologically correct volume.

Surface-based methods include the methods of Guskov [113], Wood [114], and Jaume [115, 116]. Guskov's method is based on wavefront propagation. The topology handle, or equivalently hole, is detected by tracking the splitting and merging of the front. Wood's method is based on the extended Reeb graph, a graphical representation of the topology of the surface. Topology handles are eliminated by removing a loop in the surface and the cost of the removal is measured in terms of the length of the loop. Each topology handle has to be detected and corrected separately, which means that the computation complexity depends on the number of handles in the surface. Jaume followed Guskov's and Wood's method with improved computation efficiency and topology correction accuracy.

Volume-based methods include the methods of Shattuck [117], Han [118], Segonne [119, 120], and Kriegeskorte [121]. Shattuck's method is also based on Reeb graph. Topology correction is performed in a multiscale manner on the white matter. Starting from the smallest scale, the handles whose elimination cost is smaller than the scale are eliminated at each scale. The volume are then updated and topology correction continues with increased scale if the task hasn't been completed. Handle elimination cost is measured in terms of the number of removed voxels. One of the major limitations of Shattuck's method is that the cut of the handle can only be oriented along cartesian axes. Han also took a multiscale approach and broke the above limitation. Candidate handle cuts are localized by successive morphological opening. Therefore, the cost of handle elimination is mainly measured by the distance-to-surface metric. A graph-based method is then used to determine the final handle cuts attempting to minimize the number of removed voxels at a specific scale. Segonne and Kriegeskorte's methods also evaluate the cost of handle elimination in terms of the distance-to-surface metric and topology correction is conducted in a region growing form. Using the distance-to-surface metric as the cost measurement ends up with cutting handles along thin regions of the object. Due to the surface-likeness of the white matter, these methods may incur significant modification to the shape of the white matter.

The major novelty of the present method lies in the way the handle elimination cost is measured. The present method considers the special "surface-like" shape of the white matter and assigns higher cost to the handle cuts that would lead to more modification

to the “surface-likeness” of the white matter. The “surface-likeness” of an object can be quantitatively measured in terms of the similarity between the object and its surface skeleton.

To our knowledge, all existing methods correct topology defects by elimination of handles either in the foreground or the background. The present method conducts handle elimination on three associated objects: the white matter object, the gray matter object, and the background object. It is reasonable to believe that the pre-segmentation provides credible information that voxels labeled as “background” have less possibility of actually being white matter than those labeled as “gray matter”. Topology defects in the white matter volume can be corrected in three ways: cutting handles in the white matter volume, cutting handles in the gray matter volume, or cutting handles in the union of the gray matter and the background. The third way is assigned a lower priority than the other two. In other words, the topology correction solution without modification to the background is preferred.

CHAPTER VIII

Segmentation Pipeline

In this chapter, we present a neuroanatomical segmentation pipeline integrating various techniques presented in previous chapters for extraction of cerebral white matter and gray matter as well as reconstruction of cortical surfaces. The segmentation pipeline starts with relative thresholding to obtain an initial voxel classification. The following steps can be regarded a series of correction procedures on the classification errors produced in the initial step. We propose an technique called “terrain analysis” to recognize thin parts of WM and CSF that are missed by relative thresholding. The entire set of voxels labeled as WM is then processed by the multiscale morphological topology correction procedure. The topology correct WM set contains the cerebral white matter, white matter in other parts of the brain, as well as some positive errors. Cerebral white matter is then extracted from its superset based on the cell complex analysis while preserving the correct topology. Cerebral gray matter extraction benefits from the prior segmentation of cerebral white matter and exploits the layered structure of GM/WM and the nearly uniform thickness of cerebral cortex. Finally, we use a topologically consistent marching cubes isosurface algorithm [110] to generate the triangulated surface representation of cortical surfaces.

8.1 Relative Thresholding

Relative thresholding serves as the initial voxel classification procedure in the entire segmentation pipeline. It classifies all voxels in the gray level T1-weighted MR image into

three types: white matter, gray matter and background (including CSF). It is essentially a procedure of intensity comparison between near voxels along the gradient flow with respect to two relative thresholds. First, a relative threshold between GM and WM is used to classify all voxels into WM and non-WM. Then, the non-WM voxels are further classified into GM and background with a relative thresholding between CSF and GM.

Relative thresholding is effective because it is based on a structure and image modeling approach that exploits various *a priori* knowledge. First, we know *a priori* that CSF, GM and WM form a layered structure. Second, we know *a priori* that the gray level of CSF, GM and WM is in ascending order in any local regions in T1-weighted MR images. Third, the intensity inhomogeneity that complicates image segmentation has a beneficial property that it can be seen as a signal field with very low frequency. Fourth, the thickness of cerebral cortex is nearly uniform. The detailed formulation of relative thresholding exploiting these *a priori* knowledge is presented in chapter III.

The major advantage of relative thresholding is that it is robust to intensity inhomogeneity without needing to treat the image artifact explicitly. In other words, intensity inhomogeneity is transparent to relative thresholding such that the procedure is performed as if there was no intensity inhomogeneity in the image at all. Relative thresholding is also able to adapt to large intensity variations within a given brain tissue and thus tends to produce more accurate segmentation. It can also be seen as a new edge detection method that avoids shortcomings of the traditional methods. First, it produces coherent regions labeled with brain tissue types. Second, it is able to recognize blurred edges and tissue boundaries where intensities vary smoothly. Third, it is able to suppress spurious edges between voxels of same tissue types.

Relative thresholding is a very efficient procedure. The user can select different relative thresholds and see the result in real-time. When automatic segmentation achieves a less than acceptable results, user intervention with a few global parameters actually provides a simple and effective mechanism to steer the segmentation. Automatic selection of relative thresholds is under study.

For convenience of description in the following section, we denote the set of WM, GM and background voxels generated by relative thresholding respectively as \mathcal{W}^0 , \mathcal{G}^0 , and \mathcal{B}^0 . The initial tissue segmentation result using relative thresholding is illustrated in figure 8.1.

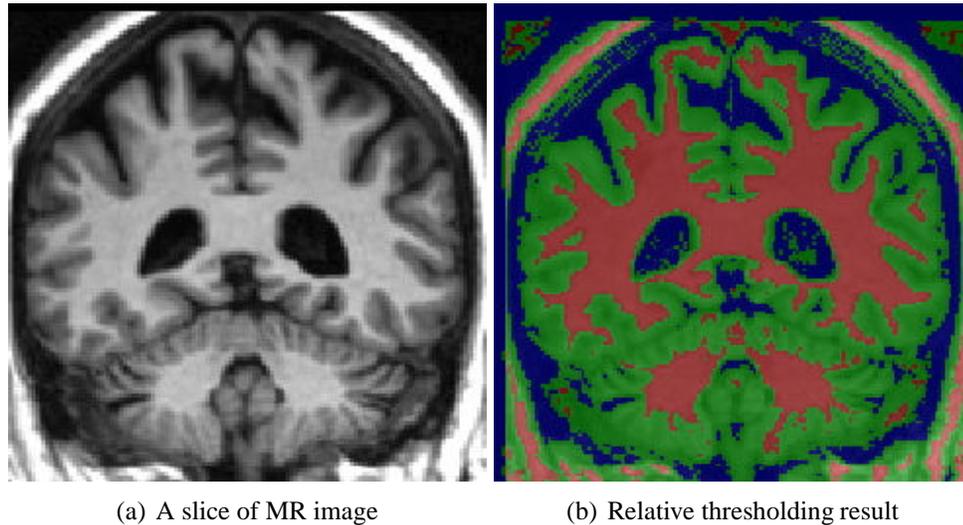


FIGURE 8.1: Relative thresholding result

8.2 Terrain Analysis

Due to limited resolution in image acquisition, there are parts of white matter and CSF blurred by partial volume effect plus smoothing filtering and thereby missed by relative thresholding. These misclassifications are mainly reflected as positive errors of gray matter. In this section, we present a technique called *terrain analysis* to recover these missed structures.

Terrain analysis borrows some notions in terrain classification for further and more reliable classification of voxels in \mathcal{G}^0 into thin WM, thin background, and GM. Here we are interested in three terrain types: ridge, valley, and slope. A point in the terrain surface is intuitively taken as a ridge if the altitudes of its two neighbor points along the pair of principle directions are both lower than that of the ridge point. By the *pair of principle*

directions we mean two nearly opposite directions that are steepest and whose projections onto the sea-level plane form an angle close to 180° . Similarly, a point in the terrain surface is a *valley point* if the altitudes of its two neighbor points along the pair of principle directions are both higher than that of the valley point. For a *slope point*, the altitude of one of its neighbors along the pair of principle directions is higher than that of the slope point while the altitude of the other is lower.

We can perform terrain analysis in a 2D image taking the gray level as the altitude and the image plane as the sea-level plane. Terrain analysis can also be generalized in 3D images with additional computation. Given a T1-weighted MR image that is Gaussian-smoothed at a small scale, we can make an informal observation that thin WM parts correspond to ridges, thin CSF parts correspond to valleys, and GM parts correspond to slopes. This observation is reasonable in terms of the order of average gray levels of CSF, GM, and WM in T1-weighted MR images and the layered structure of the three brain tissues. Figure 8.2 gives an example on how terrain analysis can improve the initial tissue segmentation using relative thresholding.

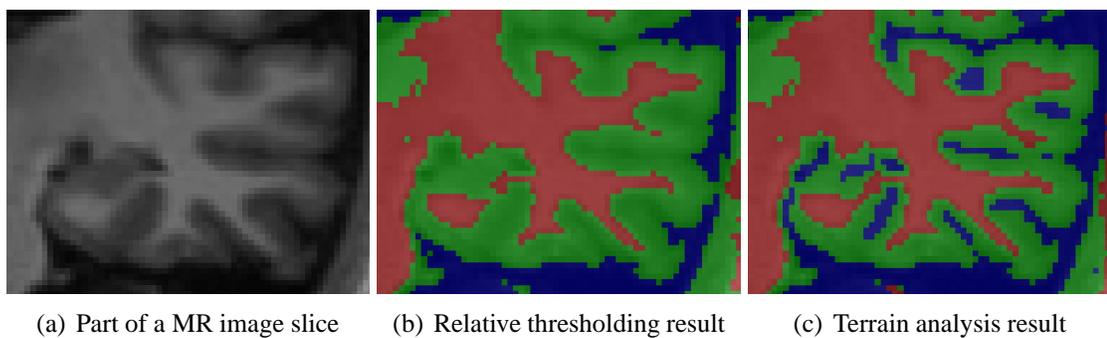


FIGURE 8.2: Terrain analysis result. New valley and ridge points are added and shown with blue and red respectively.

There are two options in terrain analysis: the scale, and whether it is performed in 2D or 3D. The scale represents the size of the neighborhood in which terrain analysis is performed. If terrain analysis is performed in 2D, then the pair of principle directions can only be in the x , y , or z plane that intersect with the point in question. In 3D, the pair of principle directions can be in the 3D space. Let s be the scale of terrain analysis. Then the size of the neighborhood $\mathcal{N}^s(p)$ at scale s of a point p is $(2s + 1) \times (2s + 1)$ in 2D and

$(2s + 1) \times (2s + 1) \times (2s + 1)$ in 3D. A constraint on a candidate pair of principle directions is that the two directions must form an angle $\alpha \in [135^\circ, 180^\circ]$.

Given a point p and a candidate pair of principle directions (d_1, d_2) , we can determine the two sets of points \mathcal{D}_1 and \mathcal{D}_2 at scale s . Each point in \mathcal{D}_i is in the neighborhood $\mathcal{N}^s(p)$ and along the direction d_i , for $i = 1, 2$. We then calculate the average gray levels (or weighted average with weights set according to how the direction pass through the points), m_1 and m_2 , for points in \mathcal{D}_1 and \mathcal{D}_2 respectively. Let y_p be the gray level of point p . We can calculate a metric *steepness* for p at scale s with respect to the candidate pair of principle directions (d_1, d_2) as $steepness(p, s, d_1, d_2) = |m_1 - y_p| + |m_2 - y_p|$. We can then determine the terrain type for p at scale s with respect to (d_1, d_2) as:

$$\begin{aligned} terrainType(p, s, d_1, d_2) &\leftarrow \text{ridge, if } m_1 > y_p \wedge m_2 > y_p; \\ terrainType(p, s, d_1, d_2) &\leftarrow \text{valley, if } m_1 < y_p \wedge m_2 < y_p; \\ terrainType(p, s, d_1, d_2) &\leftarrow \text{slope, otherwise.} \end{aligned}$$

To perform terrain analysis for point p at scale s , we calculate the steepness and determine the terrain type for p with respect to each candidate pair of principle directions at scale s' , for $s' = 1, 2, \dots, s$, and select the terrain type with the greatest steepness as the final result. Note that the gray level image is Gaussian-smoothed at the same scale as that used in relative thresholding for intensity comparison.

In practice, we first perform terrain analysis at scale 2 for points in \mathcal{G}^0 , which is divided into three sets: the set of ridge points \mathcal{R}^2 , the set of the valley points \mathcal{V}^2 , and the set of slope points \mathcal{S}^2 . Then we perform terrain analysis at scale 1 for points in $\mathcal{V}^2 \cup \mathcal{S}^2$, which is divided into \mathcal{R}^1 , \mathcal{V}^1 , and \mathcal{S}^1 . Eventually we update the set of WM, GM, and background as $\mathcal{W}^1 = \mathcal{W}^0 \cup \mathcal{R}^2$, $\mathcal{B}^1 = \mathcal{B}^0 \cup \mathcal{V}^1$, and $\mathcal{G}^1 = \mathcal{S}^1 \cup \mathcal{R}^1$.

8.3 Topology Correction Of White Matter

Topology correct white matter should contain only one handle that is formed by two cerebral hemispheres and the brainstem, while cerebral white matter should contain no

handle. However, topology defects usually arise in the results generated by relative thresholding and terrain analysis. We use a multiscale morphological approach to eliminate the topology defects in the white matter \mathcal{W}^1 . The method is multiscale in that handles/tunnels of smaller size are eliminated in prior to those of larger size. The cost to eliminate a handle/tunnel (i.e. the size of the handle/tunnel), is determined by the wideness of the handle/tunnel so that the topology correction procedure largely preserves the surface-likeness of the white matter. Unlike two-fold topology correction methods that involve a foreground object and its complement, our method performs in a three-fold manner in that it involves three disjoint objects, WM, GM and background, so that more information provided by the prior voxel classification step is utilized in the topology correction step. Such a three-fold mechanism and consideration of surface-likeness of the white matter in the multiscale framework gives rise to robust topology correction that produces reasonable topology correction solutions.

Topology defects, particularly tunnels in core regions of white matter, may severely affect the connectivity of the white matter. Therefore, eliminating such topology defects provides more reliable data for subsequent processing steps that analyze the connectivity of the white matter for cerebral white matter extraction. Subsequent processing of cerebral white matter preserves the correct topology and cerebral gray matter processing is dependent on the topology correct cerebral white matter. In all, topology correction of white matter is meaningful not only in the sense of the correct topology of the final result but also in the sense of feeding more reliable intermediate result to subsequent processing steps.

For the convenience of description, we denote the set of WM voxels after topology correction as \mathcal{W}^2 . It contains three parts: cerebral white matter, non-cerebral white matter, and voxels that are mislabeled as white matter. In the next section, we describe how to extract cerebral white matter from \mathcal{W}^2 .

Figure 8.3 demonstrates the behavior of the method on eliminating handles in the white matter and the resultant cortical surface after topology correction. The handles in the white matter are removed by filling the associated tunnels (i.e. the gray matter handles). After initial tissue classification, there is usually a handle formed by the two cerebral hemispheres, the corpus callosum and the brainstem. Although the handle is much thicker at brainstem than at corpus callosum, our method is able to break this handle at the brainstem where it

is more narrow than other places along the handle. This is shown in figure 8.4. Finally, note that topology correction is performed on the super set of the true white matter. This is illustrated in figure 8.5, where a handle in the non-brain tissue is cut.

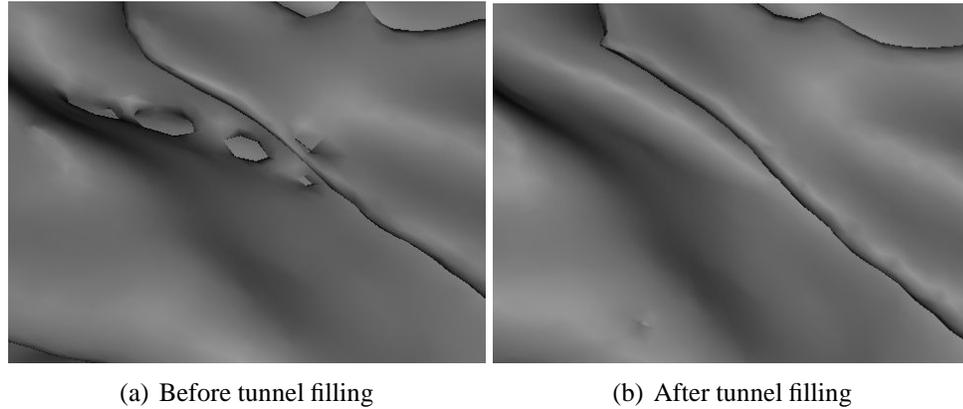


FIGURE 8.3: Tunnel filling in topology correction

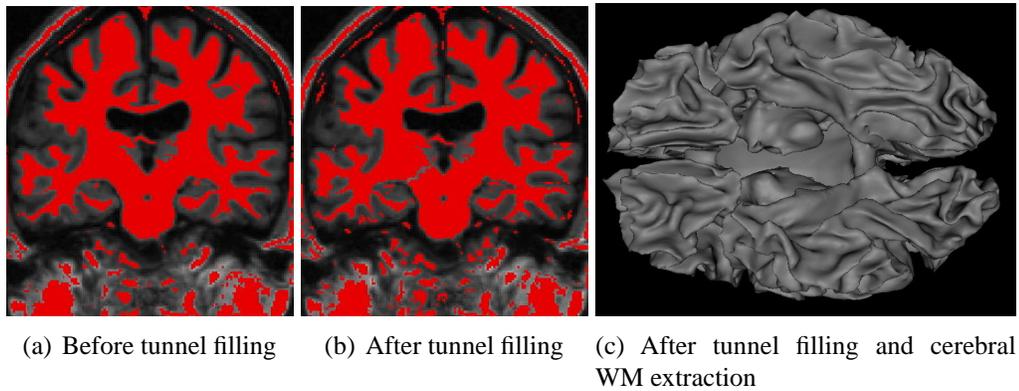


FIGURE 8.4: Brainstem breaking in topology correction

8.4 Cerebral White Matter Segmentation

After topology correction of white matter, the set \mathcal{W}^2 is composed of the following subsets: cerebral white matter, brainstem, cerebellar white matter, and voxels misclassified

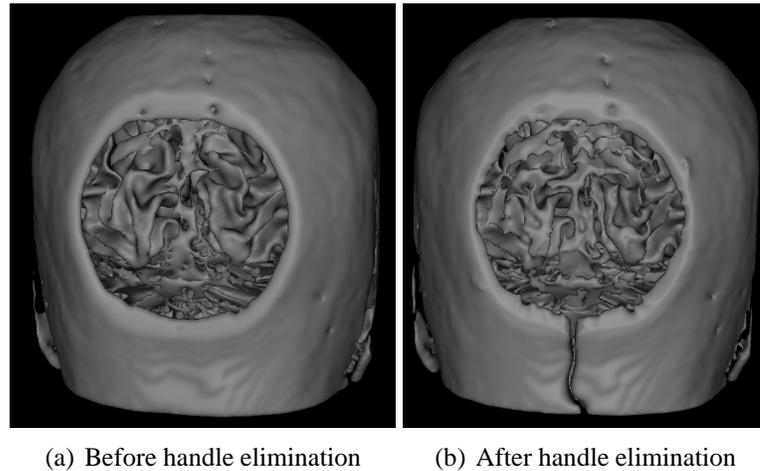


FIGURE 8.5: Handle elimination in topology correction

as white matter. Furthermore, cerebral white matter can be divided into two cerebral hemispheres and the corpus callosum. Locating and extracting cerebral white matter is based on the assumption that there are three scales of connectivities between those subsets in \mathcal{W}^2 :

- the low scale of connectivity is at the brainstem and between components of WM voxels and those of misclassified WM voxels;
- the medium scale of connectivity is at the corpus callosum between two hemispheres;
- the high scale of connectivity is within each cerebral hemisphere itself.

We will utilize the high scale of connectivity within cerebral hemispheres to locate the cerebral white matter and the low scale of connectivity to extract the cerebral white matter, while the connectivity is measured with the algorithm of cell complex based morphometric analysis (CCBMA) .

The entire procedure of cerebral white matter extraction consists of the following steps:

1. Thick-surface skeletonization of \mathcal{W}^2 resulting in thick surface skeleton \mathcal{S}_2 ;
2. Thin-surface skeletonization of \mathcal{S}_2 resulting in thin surface skeleton \mathcal{S}_1 ;
3. CCBMA of \mathcal{S}_1 ending up with a 1-complex \mathcal{C} in which each edge is assigned three metrics: connectivity, radius and wideness;

4. Locate the center of each cerebral hemisphere;
5. Dilate the center of both cerebral hemispheres to obtain a core cerebral white matter \mathcal{K} ;
6. Dilate \mathcal{K} to reconstruct the thin surface skeleton of cerebral white matter \mathcal{S}_1^w ;
7. Dilate \mathcal{S}_1^w to reconstruct the thick surface skeleton of cerebral white matter \mathcal{S}_2^w ;
8. Dilate \mathcal{S}_2^w to reconstruct cerebral white matter and restore its topology.

The final result of cerebral white matter is denoted as \mathcal{W}^3 .

We have presented surface skeletonization in chapter V and CCBMA in chapter VI. In the rest of this section, we will describe step 4 in subsection 8.4.1, step 5 and step 6 in subsection 8.4.2, step 7 and step 8 in subsection 8.4.3. Figure 8.6 gives an illustration of step 5 through step 8. Localization of the cerebral hemisphere centers is demonstrated in figure 8.7.

8.4.1 Locating Cerebral White Matter

Since \mathcal{W}^2 contains no handles and surface skeletonization and CCBMA are topology-preserving, we are guaranteed that the 1-complex \mathcal{C} contains no loops and can be seen as a tree \mathcal{T} consisting of a vertex set \mathcal{V} and an edge set \mathcal{E} . Let $v_1 \in \mathcal{V}$ and $v_2 \in \mathcal{V}$ be any two distinct vertices in the tree, then there must be a path $P^c(v_1, v_2) = \{v_0^c, v_1^c, \dots, v_n^c\}$ in \mathcal{T} connecting v_1 and v_2 , where $v_0^c = v_1$ and $v_n^c = v_2$. The path $P^c(v_1, v_2)$ is referred to as the *corpus callosum path* because it should pass through the corpus callosum if v_1 and v_2 are the centers of the two cerebral hemispheres. v_1^c is referred to as the *corpus callosum neighbor* of v_1 in $P^c(v_1, v_2)$ and v_{n-1}^c is referred to as the *corpus callosum neighbor* of v_2 in $P^c(v_1, v_2)$. The following formulation defines a weight for each vertex pair in the tree so that we can select the pair of vertices with the greatest weight as the centers of the cerebral hemispheres.

Given a pair of vertices (v_1, v_2) and its corpus callosum path $P^c(v_1, v_2)$, we define a *hemisphere path* of vertex $v \in \{v_1, v_2\}$ with respect to v 's corpus callosum neighbor v^c in

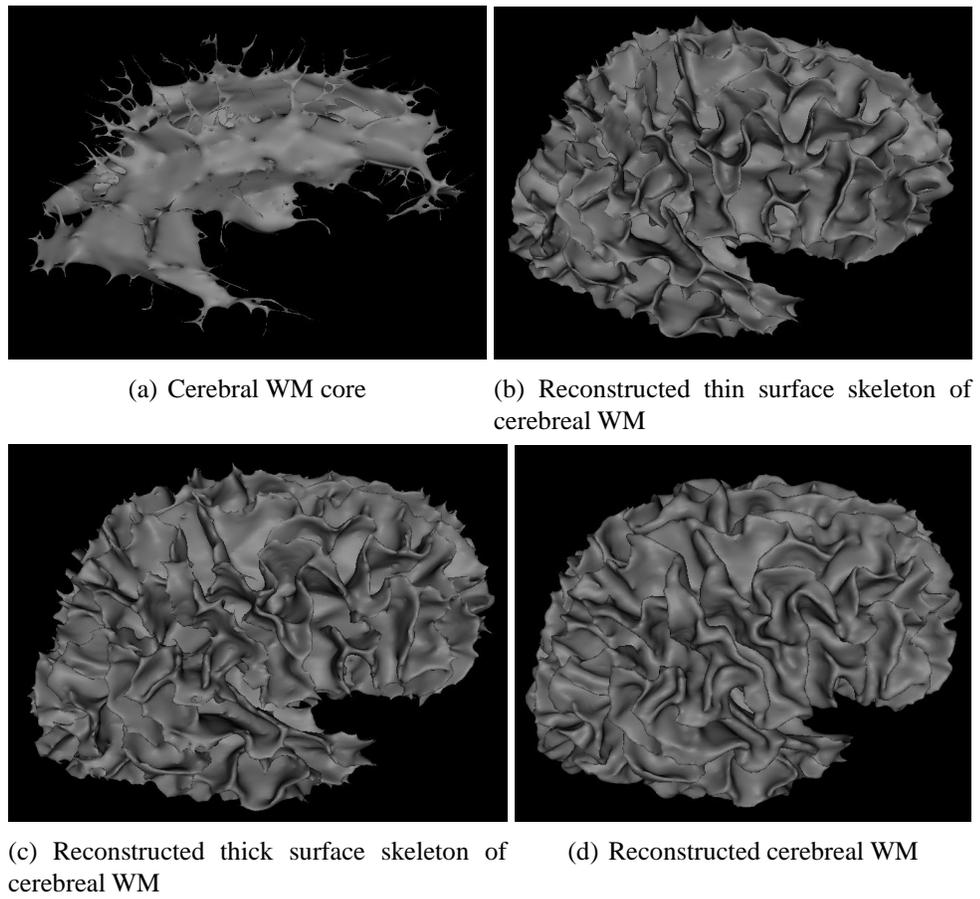


FIGURE 8.6: Cerebral WM segmentation

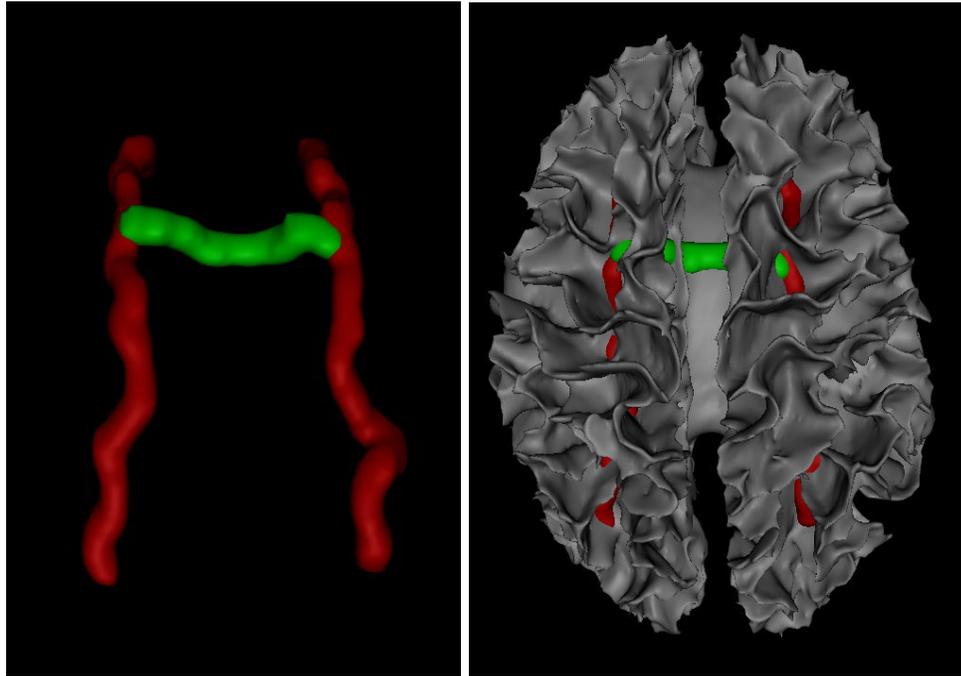


FIGURE 8.7: Maximum hemisphere paths(in red) and corpus callosum path (in green) and their embedding in cerebral WM

$P^c(v_1, v_2)$ as a path in T of length less than d_H (empirically 200mm) that passes through v but not v^c and denote it as $P^h(v, v^c)$.

For a hemisphere path $P^h(v, v^c) = \{v_0^h, v_1^h, \dots, v_i^h = v, v_{i+1}^h, \dots, v_m^h\}$ of vertex v with respect to v 's corpus callosum neighbor v^c in $P^c(v_1, v_2)$, we define its weight as

$$weight(P^h(v, v^c)) = \sum_{j=0}^{m-1} weight(e(v_j, v_{j+1}), v),$$

where $weight(e(v_j, v_{j+1}), v)$ refers to the weight of edge $e(v_j, v_{j+1})$ with respect to vertex v . Let $P = \{v_0 = v_x, v_1, \dots, v_n = v_y\}$ be a path in \mathcal{T} connecting vertex v_x and v_y , then we define the weight of edge $e(v_y, v_z)$ with respect to v_x as

$$weight(e(v_y, v_z), v_x) = \min(connectivity(e(v_0, v_1)), connectivity(v_1, v_2), \dots, connectivity(v_{n-1}, v_n), connectivity(v_y, v_z)).$$

We further define the weight of a vertex $v \in \{v_1, v_2\}$ with respect to its corpus callosum neighbor v^c in $P^c(v_1, v_2)$ as the greatest weight of the hemisphere paths of v with respect to $P^c(v_1, v_2)$ and denote it as $weight(v, v^c)$. Here, the hemisphere path with the greatest weight is referred to the *maximum hemisphere path* of vertex v with respect to $P^c(v_1, v_2)$. Finally, we define the weight of a pair of vertex (v_1, v_2) as the weight of the one of them with lower weight. The criterion of finding the hemisphere centers is that the pair of vertices (v_1^*, v_2^*) with the greatest weight is taken as the centers of the cerebral hemispheres. Exhaustive searching for the center vertices are impractical due to its time complexity. Next, we will present an efficient and effective searching algorithm, which depends on an ordering of the strength of the branches connected to each vertex in the tree.

For any vertex v in the tree \mathcal{T} , let $\mathcal{N}(v)$ be the set of vertices connected to v with a single edge. Each incident vertex $v_i \in \mathcal{N}(v)$ corresponds to a branch of the tree if we take the current vertex v as the root of the tree. We consider the path $P^b = \{v_0 = v, v_1 = v_i, v_2, \dots, v_n\}$ of length less than $radius(e(v, v_i))$ and define the *strength* of the path P^b as:

$$strength(P^b) = \sum_{j=0}^{n-1} weight(e(v_j, v_{j+1}), v),$$

where $weight(e(v_j, v_{j+1}), v)$ refers to the weight of edge $e(v_j, v_{j+1})$ with respect to v .

We further define the strength of the directed edge $\vec{e}(v, v_i)$ as the maximum strength of all paths headed by vv_i whose length are less than $radius(e(v, v_j))$. For each vertex v , we can compute and order the strengths of all directed edges emanated from v . The directed edge with the greatest strength is referred to as the primary edge of v and the corresponding neighbor vertex is referred to as the primary neighbor of v and denoted as $N^{1st}(v)$. Similarly, we can define second and tertiary edge and neighbor of v , if any. The secondary neighbor and tertiary neighbor of v are respectively denoted as $N^{2nd}(v)$ and $N^{3rd}(v)$.

Having computed the strengths of edges emanated from each vertex in \mathcal{T} , we reduce the searching space of hemisphere paths of $v \in \{v_1, v_2\}$ with respect to v 's corpus callosum neighbor v^c in $P^c(v_1, v_2)$ by only considering paths $P^h(v) = \{v_0, v_1, \dots, v_i = v, v_{i+1}, \dots, v_m\}$ such that:

$$\begin{aligned}
v_{i-1} &= N^{1st}(v) \text{ and } v_{i+1} = N^{2nd}(v) && \text{if } v^c = N^{3rd}(v) \\
v_{i-1} &= N^{1st}(v) \text{ and } v_{i+1} = N^{3rd}(v) && \text{if } v^c = N^{2nd}(v) \\
v_{i-1} &= N^{2nd}(v) \text{ and } v_{i+1} = N^{3rd}(v) && \text{if } v^c = N^{1st}(v) \\
v_{j-1} &= N^{1st}(v_j) && \text{if } v_{j+1} \neq N^{1st}(v_j) \\
v_{j-1} &= N^{2nd}(v_j) && \text{if } v_{j+1} = N^{1st}(v_j) \\
v_{j+1} &= N^{1st}(v_j) && \text{if } v_{j-1} \neq N^{1st}(v_j) \\
v_{j+1} &= N^{2nd}(v_j) && \text{if } v_{j-1} = N^{1st}(v_j)
\end{aligned}
\text{ for } j = 1, \dots, i-1 \text{ and } j = i+1, 1, \dots, m-1$$

This is essentially an act of searching for hemisphere paths of v only along strongest available edges starting from v . It is an effective heuristic and dramatically reduces our searching space. In this efficient algorithm, the weight of a vertex $v \in \{v_1, v_2\}$ with respect to its corpus callosum neighbor v^c in $P^c(v_1, v_2)$ is also determined at the same time while its maximum hemisphere path is found.

To find the centers of cerebral hemispheres, we start from each vertex v in \mathcal{T} with degree greater than 1 and assume it is in the target corpus callosum path. Then we initialize the corpus callosum path as $N^{1st}(v)vN^{2nd}(v)$ and have two sides to search for the hemisphere centers respectively. At each side, we further reduce our searching space by marching along the strongest available edges (a similar procedure to that for finding the maximum hemisphere path of a vertex). At one side, we can find a vertex v_x with greatest weight with respect to its corpus callosum neighbor in $P^c(v_x, v)$; at the other side, we can find another such vertex v_y . (v_x, v_y) form a candidate pair of hemisphere centers with weight being the minimum of their weights. The pair of vertex with greatest weight is selected as the target cerebral hemisphere centers. This procedure is presented in algorithm 11.

Sometimes there may be large patches of skull structure in \mathcal{W}^2 which possess large connectivity as well as large wideness. To differentiate these structures from cerebral white matter, which possess large connectivity and limited wideness, we modify the connectivity of each edge in \mathcal{T} used in locating cerebral hemisphere centers such that:

$$\begin{aligned}
connectivity(e) &\leftarrow connectivity(e) && \text{if } wideness(e) < 180mm \\
connectivity(e) &\leftarrow connectivity(e) * 180/wideness(e) && \text{if } wideness(e) \geq 180mm
\end{aligned}$$

Algorithm 11: Finding cerebral hemisphere centers

Data: A tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ spanning the surface skeleton of \mathcal{W}^2

Result: Cerebral hemisphere center v_1^* and v_2^*

$weight^* \leftarrow 0$;

foreach vertex $v \in \mathcal{V}$ such that $degree(v) > 1$ **do**

$weight_x \leftarrow 0$; $v_x \leftarrow N^{1st}(v)$; $v_x^* \leftarrow v_x$; $v_z \leftarrow v$;

while $degree(v_x) > 1$ **do**

if $weight(v_x, v_z) > weight_x$ **then**

$v_x^* \leftarrow v_x$;

$weight_x \leftarrow weight(v_x, v_x^c)$;

$v_z \leftarrow v_x$;

if $v_z \neq N^{1st}(v_x)$ **then**

$v_x \leftarrow N^{1st}(v_x)$

else

$v_x \leftarrow N^{2nd}(v_x)$

$weight_y \leftarrow 0$; $v_y \leftarrow N^{2nd}(v)$; $v_y^* \leftarrow v_y$; $v_z \leftarrow v$;

while $degree(v_y) > 1$ **do**

if $weight(v_y, v_z) > weight_x$ **then**

$v_y^* \leftarrow v_y$;

$weight_y \leftarrow weight(v_y, v_y^c)$;

$v_z \leftarrow v_y$;

if $v_z \neq N^{1st}(v_y)$ **then**

$v_y \leftarrow N^{1st}(v_y)$

else

$v_y \leftarrow N^{2nd}(v_y)$

$weight(v_x^*, v_y^*) \leftarrow \min(weight_x, weight_y)$;

if $weight(v_x^*, v_y^*) > weight^*$ **then**

$weight^* \leftarrow weight(v_x^*, v_y^*)$;

$v_1^* \leftarrow v_x^*$; $v_2^* \leftarrow v_y^*$;

8.4.2 Reconstruction Of Thin-surface Skeleton Of Cerebral White Matter

Having identified the cerebral hemisphere centers v_1^* and v_2^* , we then use a connectivity threshold t_c to determine the cerebral white matter core \mathcal{K} that is connected to v_1^* and v_2^* above a certain scale. Formally, \mathcal{K} is a set of vertices and edges defined as follows:

1. $\{v_1^*, v_2^*\} \subset \mathcal{K}$;
2. For any vertex $v \in \mathcal{K}$, its incident edge $e(v, v') \in \mathcal{C}^1$ is in \mathcal{K} if $connectivity(e) > t_c$ and $radius(e) > t_c/3$;
3. For any edge $e \in \mathcal{K}$, its child edge $e' \in \mathcal{C}^2$ is in \mathcal{K} if $connectivity(e') > t_c$ and $radius(e') > t_c/3$ (the "parent-children" relationship between edges is established in CCBMA);
4. For any edge $e(v_1, v_2) \in \mathcal{K}$, its incident vertices v_1 and v_2 are also in \mathcal{K} .

Given the kernel set of cerebral white matter, we reconstruct the thin surface skeleton \mathcal{S}^w using the following simple rules:

1. $\mathcal{K} \subset \mathcal{S}^w$;
2. For each edge $e \in \mathcal{K} \cap \mathcal{C}^1$, any edge $e' \in \mathcal{C}^1$ that has a path of length less than $radius(e)$ is also in \mathcal{S}^w ;
3. For each edge $e \in \mathcal{S}^w$, its child edges are also in \mathcal{S}^w ;
4. For any edge $e(v_1, v_2) \in \mathcal{S}^w$, its incident vertices v_1 and v_2 are also in \mathcal{S}^w .

The set \mathcal{S}^w contain both vertices and edges. We specially use \mathcal{S}_1^w to denote the voxels corresponding to the vertices in \mathcal{S}^w . Note that \mathcal{S}_1^w is a subset of the thin surface skeleton \mathcal{S}_1 of \mathcal{W}^2 , i.e. $\mathcal{S}_1^w \subseteq \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{W}^2$.

8.4.3 Reconstruction Of Cerebral White Matter

By reconstruction of surface skeleton of cerebral white matter, we have divided the set \mathcal{S}_1 into two subsets: \mathcal{S}_1^w and $\mathcal{S}_1^{\bar{w}} = \mathcal{S}_1 \setminus \mathcal{S}_1^w$. Now we reconstruct the thick surface skeleton of cerebral white matter, denoted as \mathcal{S}_2^w . Let $\mathcal{S}_2^{\bar{w}} = \mathcal{S}_2 \setminus \mathcal{S}_2^w$. First of all, it is obvious that $\mathcal{S}_1^w \subseteq \mathcal{S}_2^w$ and $\mathcal{S}_1^{\bar{w}} \subseteq \mathcal{S}_2^{\bar{w}}$, therefore, the task is essentially to separate $\mathcal{S}_2 \setminus \mathcal{S}_1$ into two parts: those in \mathcal{S}_2^w and those in $\mathcal{S}_2^{\bar{w}}$. The simple rule that we follow here is that for each voxel $v \in \mathcal{S}_2 \setminus \mathcal{S}_1$, if the geodesic distance of v from \mathcal{S}_1^w with respect to \mathcal{S}_2 is less than that from $\mathcal{S}_1^{\bar{w}}$, then v is in \mathcal{S}_2^w ; otherwise, v is in $\mathcal{S}_2^{\bar{w}}$. The geodesic distance between v and \mathcal{S}_1^w with respect to \mathcal{S}_2 is the length of the shortest path in \mathcal{S}_2 connecting v and \mathcal{S}_1^w . Having determined the thick surface skeleton of cerebral white matter (\mathcal{S}_2^w), we reconstruct the cerebral white matter, denoted as \mathcal{W}^{3-} such that:

1. $\mathcal{S}_2^w \subseteq \mathcal{W}^{3-}$;
2. For each voxel $v \in \mathcal{S}_2^w$, any voxel $v' \in \mathcal{W}^2$ whose distance from v is less than or equal to $d(v)$ is also in \mathcal{W}^{3-} , where $d(v)$ is the distance of v from the boundary of \mathcal{W}^2 .

The procedures described above do not enforce particular topology control and the result \mathcal{W}^{3-} is not necessarily topologically equivalent to \mathcal{W}^2 . To restore the correct topology on \mathcal{W}^{3-} , we perform a topology-preserving erosion on \mathcal{W}^2 with respect to \mathcal{W}^{3-} . It is a process that keeps removing voxels from \mathcal{W}^2 until no more voxels can be removed. Each voxel v removed from \mathcal{W}^2 must satisfy the following conditions:

- v is a simple point with respect to \mathcal{W}^2 at the moment of removal;
- v is not in the set \mathcal{W}^{3-} ;

Let \mathcal{W}^3 denote the result of this conditional topology-preserving erosion. It represents our final set of cerebral white matter voxels. It is a superset of \mathcal{W}^{3-} and topologically equivalent to \mathcal{W}^2 , i.e. topologically correct.

8.5 Cerebral Gray Matter Segmentation

Segmentation of cerebral gray matter is based on the prior segmentation of cerebral white matter and utilizes the following *a priori* knowledge:

- The thickness of cerebral cortex is nearly uniform;
- Cerebral gray matter wraps around cerebral white matter so that two tissues form a layered structure;
- The average gray level of cerebral gray matter is lower than that of cerebral white matter at any local region;

Segmentation of cerebral gray matter takes a series of steps as follows, which are also illustrated in figure 8.8.

1. Initialization: $\mathcal{G}^2 \leftarrow \emptyset$. where \mathcal{G}^2 denotes the set of cerebral gray matter voxels;
2. Gradient flow analysis: For each voxel v in $\mathcal{G}^0 \setminus \mathcal{W}^3$, if there exists a path $P = \{v_0 = v, v_1, \dots, v_n\}$ of length less than d_C along the gradient flow such that $v_n \in \mathcal{W}^3$. Then we update the set of cerebral gray matter voxels as $\mathcal{G}^2 \leftarrow \mathcal{G}^2 \cup \{v\}$; The threshold d_C is associated with the average thickness of cerebral cortex. Considering the segmentation of cerebral white matter may be not ideally accurate, we set d_C to be a value greater than the average cortex thickness. In practice, we let $d_C = 15mm$.
3. Closing: The set \mathcal{G}^2 generated by the prior step will form a gray matter layer wrapping around the cerebral white matter \mathcal{W}^3 . However, there may be some narrow and thin regions between \mathcal{G}^2 and \mathcal{W}^3 that were labeled as white matter in \mathcal{W}^0 . They may be real cerebral white matter and may be actually cerebral gray matter. Nevertheless, we take them as cerebral gray matter at this moment. We first perform morphological closing at a small scale s on $\mathcal{G}^2 \cup \mathcal{W}^3$ and denote the closed cerebrum mask as \mathcal{M} . Then we update \mathcal{G}^2 as $\mathcal{G}^2 \leftarrow \mathcal{G}^2 \cup (\mathcal{W}^0 \cap \mathcal{M} \setminus (\mathcal{G}^2 \cup \mathcal{W}^3))$. Morphological closing at scale s first performs morphological dilation at scale s on \mathcal{X} resulting in \mathcal{X}' and then performs morphological erosion at scale s on \mathcal{X}' resulting in \mathcal{X}'' . We describe morphological erosion and dilation below.

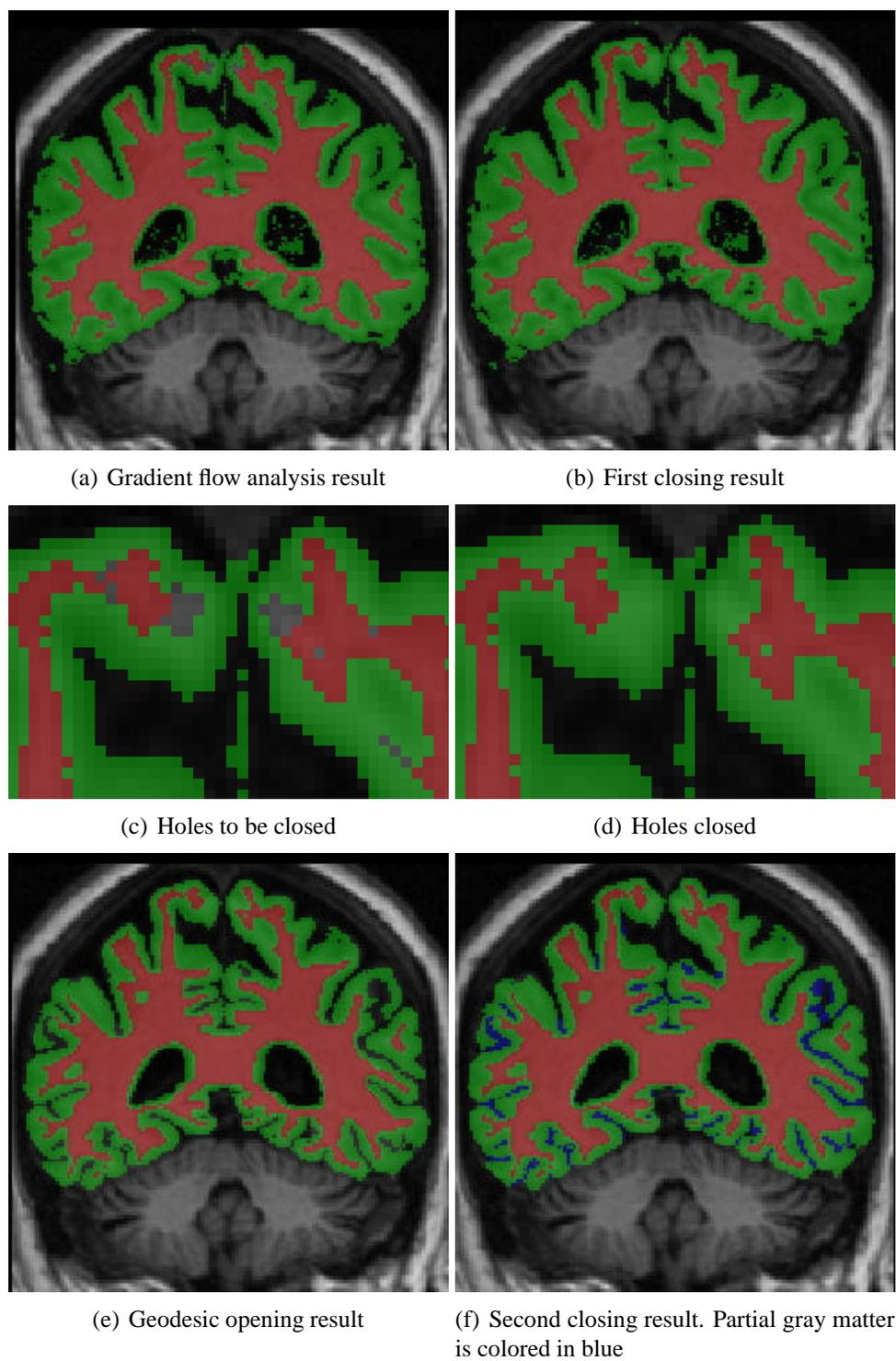


FIGURE 8.8: Cerebral GM segmentation.

4. Geodesic opening: In this step we use a operation called geodesic opening to remove certain false positive errors in \mathcal{G}^2 . First, we remove the voxels that were recognized as valley points in \mathcal{B}^1 from \mathcal{G}^2 such that $\mathcal{G}^2 \leftarrow \mathcal{G}^2 \setminus \mathcal{B}^1$. Ideally, \mathcal{G}^2 should still form a thin layer wrapping around \mathcal{W}^3 , but in practice, there may be some thin and narrow protrusions from \mathcal{G}^2 that do not wrap around any cerebral white matter regions. To eliminate these errors, geodesic opening at scale s first perform geodesic morphological erosion at scale s on $\mathcal{G}^2 \cup \mathcal{W}^3$ with respect to \mathcal{W}^3 resulting in \mathcal{F} , then perform geodesic dilation at scale s on \mathcal{F} with respect to \mathcal{G}^2 resulting in a mask where those thin protrusions at scale s are eliminated. We then update \mathcal{G}^2 such that those false positive voxels are removed. Geodesic erosion and geodesic are described below.
5. Closing: The prior step removes all valley points from \mathcal{G}^2 , which is necessary for geodesic opening to be effective. However, a valley point in sulci may be partially a cerebral gray matter voxel. To recognize such valley points, we perform morphological closing at a small scale on $\mathcal{G}^2 \cup \mathcal{W}^3$ resulting in a new cerebrum mask \mathcal{M} . Then every valley point in \mathcal{M} is taken as a partial cerebral gray matter point and assigned with a value between 0 and 1 indicating how much of the voxel belongs to gray matter. The value is calculated by comparing the intensity of the valley point to that of a near WM point along the gradient flow emanating from the valley point. Formally speaking, we update the set of cerebral gray matter as $\mathcal{G}^2 \leftarrow \mathcal{G}^2 \cup ((\mathcal{B}^1 \setminus \mathcal{B}^0) \cap \mathcal{M})$ and compute a function $f : (\mathcal{B}^1 \setminus \mathcal{B}^0) \cap \mathcal{M} \rightarrow [0, 1]$. For convenience of the description in the following sections, we denote the valley points in the cerebrum mask as $\mathcal{G}_v^2 = (\mathcal{B}^1 \setminus \mathcal{B}^0) \cap \mathcal{M}$.

Morphological erosion on \mathcal{X} at scale s can be seen as a procedure of subtracting voxels from \mathcal{X} whose distances from the boundary of \mathcal{X} are less than or equal to s . Morphological dilation on \mathcal{X} at scale s can be seen as a procedure of adding voxels in the complement of \mathcal{X} to \mathcal{X} whose distances to \mathcal{X} are less than or equal to s . Geodesic erosion on \mathcal{X} with respect to \mathcal{M} at scale s can be seen as a procedure of subtracting voxels in $\mathcal{X} \setminus \mathcal{M}$ from \mathcal{X} whose distances from the boundary of \mathcal{X} are less than or equal to s . Geodesic dilation of \mathcal{X} with respect to \mathcal{M} at scale s can be seen as a procedure of adding voxels in $\mathcal{M} \setminus \mathcal{X}$ to \mathcal{X} whose geodesic distance from \mathcal{X} are less than or equal to s .

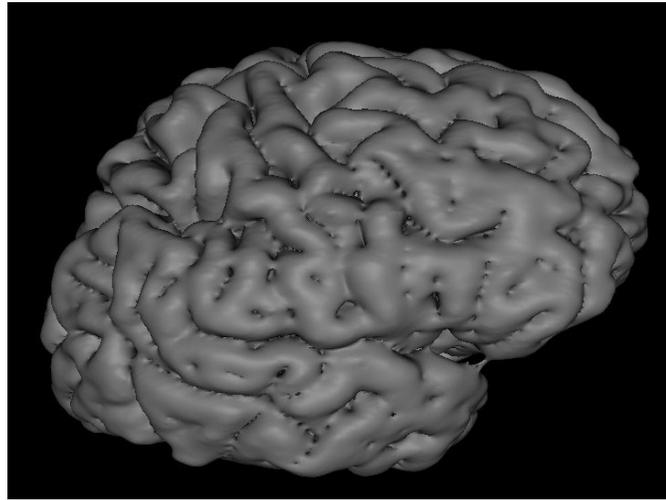
8.6 Cortical Surface Extraction

Extraction of inner cortical surface and outer cortical surface requires the set of topology-correct cerebral white matter voxels and the set of topology-correct cerebral gray matter voxels. When they are available, we use a topologically consistent marching cubes iso-surface algorithm [110] to generate the triangulated surface representation of the cortical surfaces. It is demonstrated in [110] that for (26, 6) adjacency, an isovalue less than 0.25 should be used to avoid topological paradoxes.

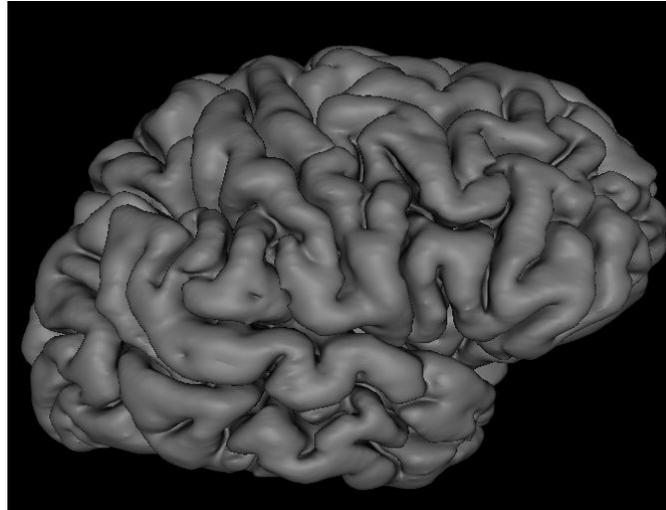
Prior processing steps have generated topology-correct cerebral white matter, denoted as \mathcal{W}^3 , and a mask of cerebral gray matter, denoted as \mathcal{G}^2 . Note that \mathcal{G}^2 is not guaranteed to possess correct topology. Computation of topology-correct cerebral gray matter consists of following steps, whose effect is demonstrated in figure 8.9.

1. Initialization: We initialize a new set of background voxels as the complement of the cerebral white matter set; i.e. $\mathcal{B}^3 \leftarrow \overline{\mathcal{W}^3}$. Let \mathcal{B}^2 denote the background voxel set before topology correction of cerebral gray matter plus the valley points in the cerebrum mask; i.e. $\mathcal{B}^2 = \overline{\mathcal{W}^3 \cup \mathcal{G}^2} \cup \mathcal{G}_v^2$.
2. Erosion: We then perform topology-and-shape-preseving geodesic erosion on \mathcal{B}^3 with respect to \mathcal{B}^2 at scale s , which is a partial curve skeletonization procedure (see chapter V) with additional condition that any voxels removed from \mathcal{B}^3 cannot be in \mathcal{B}^2 . Next, thicksurface points in the eroded background and outside \mathcal{B}^2 are thinned in a procedure similar to the thin-surface skeletonization algorithm. Eroded voxels connected to \mathcal{W}^3 are taken as the cerebral gray matter in the set \mathcal{G}^3 . Since the erosion preserves topology, the topology of $\mathcal{W}^3 \cup \mathcal{G}^3$ is equivalent to that of \mathcal{W}^3 . The scale s is associated with the average cortex thickness we know a priori. In practice we set s to be $6mm$.
3. Verification: We verify that the cerebral white matter is wrapped around by the cerebral gray matte at any places and perform necessary modification to enforce the constraint. Update \mathcal{B}^3 with the new set of \mathcal{W}^3 and \mathcal{G}^3 as $\mathcal{B}^3 \leftarrow \overline{\mathcal{W}^3 \cup \mathcal{G}^3}$.

4. Smoothing: We finally perform morphological smoothing (see chapter V) on \mathcal{B}^3 and update \mathcal{G}^3 accordingly. This step will remove the curve-like protrusions of the background and has the effect of smoothing the outer cortical surface.



(a) Cerebral GM segmentation



(b) Outer cortical surface

FIGURE 8.9: Reconstruction of outer cortical surface from cerebral GM segmentation.

CHAPTER IX

Evaluation

This chapter evaluates our neuroanatomical segmentation algorithms both quantitatively and qualitatively on a variety of MRI data sets with respect to the accuracy, automaticity, robustness and computational efficiency of our tools compared with those of four leading segmentation tools: Freesurfer, SPM5, FSL, and BrainVisa. We refer to our segmentation tool as TAS (Topological Approach to Segmentation) in the following sections.

9.1 Comparative Packages

Below is a brief description of the four tools used for our comparative performance evaluation. Their segmentation methods are presented in chapter II. Although these tools provide various amount of functionality with respect to neuroimage analysis, they all support automatic T1-weighted human brain MR image segmentation. Our comparative evaluation is thus focused on this functionality.

9.1.1 Freesurfer

FreeSurfer [122, 43] is a set of tools for reconstruction of cortical surfaces from structural MRI data and for the overlay of functional data onto the reconstructed surface. Freesurfer is developed in the Nuclear Magnetic Resonance (NMR) Center, Massachusetts General Hospital .

The cortical surface reconstruction pipeline in Freesurfer mainly consists of three steps. First, a brainmask is extracted with alignment of the structure MR image to the Talairach atlas and the bias field is corrected. Then the brain volume is labeled as various cortical or subcortical structures in a procedure based on both a subject-independent probabilistic atlas and subject-specific measured values. Finally, the cortical surfaces are constructed from the prior segmentation, which involves a topology correction procedure.

9.1.2 SPM5

SPM (Statistical Parametric Mapping) is a statistical technique for testing hypotheses about functional imaging data [123]. SPM also refers to the software developed by the Wellcome Department of Imaging Neuroscience, University College London, to carry out such analysis. SPM5 is the latest version of SPM. SPM5 features structural MRI segmentation as well as a series of functional neuroimage analysis.

Structural MRI segmentation in SPM5 can be characterized as a circular procedure that involves alternating three processing steps [42]: a bias correction step that corrects the intensity inhomogeneity, a registration step that normalizes the image to standard tissue probability maps, and a segmentation step that classifies image voxels into different tissue types. As the segmentation result, SPM5 assigns each image voxel three probabilities with respect to three tissue types: CSF, GM and WM.

9.1.3 FSL

FSL (the FMRIB Software Library) is a collection of functional and structural neuroimage analysis tools [124]. For structural segmentation, FSL mainly contains the Brain Extraction Tool (BET) for segmenting brain from non-brain in structural and functional data, and FAST (FMRIB's Automated Segmentation Tool) for bias field correction and brain segmentation into three tissue types: CSF, GM and WM.

Structural MRI segmentation in FSL consists of two steps: using BET to extract the brain and using FAST to classify tissue types. BET performs skull stripping with a surface model [96]. The underlying method of FAST is based on an Expectation-Maximization algorithm combined with a hidden Markov random field (MRF) model [28]. Due to the

regularization of the MRF model, FAST is supposed to be more robust to noise than standard finite mixture model based methods.

9.1.4 BrainVisa

BrainVisa [125, 126] is software developed at Service Hospitalier Frdric Joliot (SHFJ) that embodies an image processing factory and is distributed with a toolbox of building blocks dedicated to the segmentation of T1-weighted MR image.

Structural MRI segmentation in BrainVisa consists of four main steps. First, the user prepares the data for segmentation by specifying several key landmark points including the Anterior Commissure (AC), the Posterior Commissure (PC), an interhemispheric point, and a left hemisphere point. A brain mask is then extracted including only white matter and gray matter integrating bias field correction [127] and histogram analysis [93]. This is followed by a hemisphere partition and removal of cerebellum with morphological image analysis [128]. Finally, cerebral gray matter and white matter are differentiated with histogram analysis [93].

9.2 Data Sets

The evaluation is performed on three group of data sets: a set of BrainWeb data with groundtruth segmentation, a set of IBSR data with manually-guided expert segmentation, and a set of real scans of subjects with mild cognitive impairment or Alzheimers disease.

9.2.1 BrainWeb Data Sets

This is a group of 8 realistic T1-weighted MR simulated images with groundtruth segmentation provided by BrainWeb, a simulated brain database [129]. All 8 MR images are simulated on a normal anatomical model. The resolution of the images are $1mm^3$. In the groundtruth image, all voxels in the image are segmented into the following tissue types: Background, CSF, GM, WM, Fat, Muscle/Skin, Skin, Skull, Glial Matter, and Connective.

A variety of noise levels and levels of intensity inhomogeneity (i.e. intensity non-uniformity(INU)) are artificially introduced in the simulated images, as listed in table 9.1. As stated in BrainWeb documentation [130], “The noise in the simulated images has Rayleigh statistics in the background and Rician statistics in the signal regions. The ‘percent noise’ number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue.” The noise reference tissue used in our data set is white matter. The meaning of the intensity inhomogeneity level is as follows. “For a 20% level, the multiplicative INU field has a range of values of 0.90...1.10 over the brain area. For other INU levels, the field is linearly scaled accordingly (for example, to a range of 0.80...1.20 for a 40% level).” According to BrainWeb, the INU fields are realistic in that they are slowly-varying fields of a complex shape and were estimated from real MRI scans.

TABLE 9.1: Noise levels and IHH levels of the BrainWeb data set

Data set	1	2	3	4	5	6	7	8
Noise level	3%	3%	5%	5%	7%	7%	9%	9%
IHH level	20%	40%	20%	40%	20%	40%	20%	40%

9.2.2 IBSR Data Sets

This is a group of 18 T1-weighted real MR brain data sets and their manually-guided expert segmentations in the Internet Brain Segmentation Repository (IBSR) supported by the Center for Morphometric Analysis (CMA) at Massachusetts General Hospital [131]). The slice resolution of all datasets is $1.5mm$ and the XY resolution varies from $1mm^2$ at low end to $0.837mm^2$ at high end. The MR images have been “positionally normalized” into the Talairach orientation, but all five tools performed on this group of data assumed that the brain were not normalized. The MR images were also processed by the CMA biasfield correction routines, but it is not guaranteed that the intensity inhomogeneity is completely corrected and all five tools also treated the data sets as if no biasfield correction had ever been performed on them.

Each MR image was manually segmented into 44 individual structures including 3rd Ventricle, 4th Ventricle, Brain Stem, and Left and Right: Accumbens area , Amygdala,

Amygdala Anterior, Caudate, Cerebellum Cortex, Cerebellum Exterior, Cerebellum White Matter, Cerebral Cortex, Cerebral Exterior, Cerebral White Matter, Hippocampus, Inf Lat Vent, Lateral Ventricle, Pallidum, Putamen, Thalamus Proper, VentralDC, and Vessel.

The 18 MR images are in various levels of quality. For convenience of the following evaluation, we divided the group into two subgroups: the first 13 MR images with good quality and 5 more MR image with bad quality. Note that the ordering of the IBSR data sets is different from the original order. A map of the order we used to the original order is (1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 3, 4, 13). For example, when we refer the 3rd data set, it is actually the 5th in the original order.

9.2.3 Pathological Data Sets

In addition to the BrainWeb and the IBSR data sets, which were used for both quantitative and qualitative evaluation, we also tested five tools on an auxiliary group of 8 real MR images scanned from subjects with minor cognitive impairment or Alzheimers disease for qualitative evaluation only. The resolution of these data sets is $1.139 \times 1.211 \times 1.211 mm^3$. The source of these data sets is the Neurobiology Research Unit [132] in the University Hospital Rigshospitalet in Denmark. No groundtruth or manual segmentation are provided for these data sets.

9.3 Quantitative Evaluation

In this section, we present a quantitative evaluation on the segmentation accuracy, robustness and computational efficiency of TAS with comparison to other four packages. We use the widely-used Dice metric [42, 43, 133, 134] as the measurement for segmentation accuracy and the standard deviation of the Dice metric over a set of data as the measurement for segmentation robustness. Computational efficiency is simply measured with the running time of each package.

9.3.1 Dice Metric

Let TP refers to the number of true positives, FP to false positives and FN to false negatives, then the Dice metric is given by

$$\text{Dice metric} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \quad (9.1)$$

Note that when the segmentation is given as a probability between 0 and 1 for each image voxel for each tissue class, such as in the case of SPM5, TP, FP and FN are calculated as the sum of the probabilities instead of discrete counting.

For quantitative evaluation using Dice metric, we have to decide the tissue type on which the metric is measured. Currently TAS, like BrainVisa, only perform cerebrum segmentation while FSL and SPM5 segment the entire brain into CSF, GM and WM without extraction of the cerebrum. Freesurfer also performs segmentation on the whole brain but segments the brain into a greater number of tissue types including cerebral white matter and cerebral cortex. In our quantitative evaluation, we must calibrate the segmentation of the five package into a standard behavior so that common tissue types can be used for quantitative metric measurements.

For the BrainWeb data sets, we calibrated the segmentation of five tools to the segmentation of cerebral WM and cerebral GM, and measured the Dice metrics with respect to these two tissue types. To enable this, we manually partitioned the groundtruth whole brain (WM plus GM) at the brainstem to extract the cerebral WM and the cerebral GM. Cerebral WM and cerebral GM also have be extracted for the SPM5 and FSL segmentation results. We use a procedure (described in the next paragraph) that almost “perfectly” partitions the segmentation results based on the groundtruth partition. For BrainVisa and TAS, no transformation in the calibration is required. For Freesurfer, we just simply need to relabel all cerebral cortex voxels and all subcortical voxels excluding cerebral WM as cerebral gray matter.

Let TP-Cerebrum and TP-Cerebellum respectively denote the set of true positives of cerebrum and cerebellum in the segmentation of SPM5 and FSL. Let FP-Brain denote the entire false positives including those in cerebrum and cerebellum. The partition of the brain segmented by FSL and SPM5 is essentially the partition of FP-Brain into false positives in

cerebrum and those in cerebellum, which is described as follows. For each voxel v in FP-Brain, if it has a shorter path in FP-Brain to TP-Cerebrum than any paths in FP-Brain from v to TP-Cerebellum, then v is taken as a false positive (of WM or GM) in cerebrum; otherwise it is taken as a false positive (of GM or WM) in cerebellum.

For the IBSR data, we calibrated the segmentation of five tools to the segmentation of cerebral cortex and cerebral white matter, and measured the Dice metrics with respect to these two tissue types. These quantitative metrics give an evaluation on the accuracy of the cortical surface reconstruction that depend on segmentation of cerebral cortex and cerebral WM and are irrelevant to segmentation of subcortical gray matter tissues. Since Freesurfer explicitly labels cerebral cortex and cerebral white matter, we do not need to do any transformation in the calibration. The calibration of FSL and SPM5 first conducts the brain partition to extract the cerebral WM and cerebral GM. Given the set of cerebral WM and cerebral GM segmented by FSL, SPM5, BrainVisa or TAS, we measured the Dice metrics with respect to cerebral cortex and cerebral WM in the same way as described below.

In the measurement of the Dice metric with respect to cerebral cortex, the true positives are the voxels labeled as cerebral cortex in the manual segmentation and cerebral GM in the automatic segmentation, the false positives are the voxels labeled as cerebral GM in the automatic segmentation but not cerebral GM (i.e. cerebral cortex or other subcortical GM) in the manual segmentation, and the false negatives are the voxels labeled as cerebral cortex in the manual segmentation but not cerebral GM in the automatic segmentation.

In the measurement of the Dice metric with respect to cerebral WM, the true positives are the voxels labeled as cerebral WM in both the manual segmentation and the automatic segmentation, the false positives are the voxels labeled as cerebral WM in the automatic segmentation but neither cerebral WM nor subcortical GM in the manual segmentation, and the false negatives are the voxels labeled as cerebral WM in the manual segmentation but not cerebral WM in the automatic segmentation.

9.3.2 Experiments

Freesurfer

We tested Freesurfer on both the BrainWeb and the IBSR data sets in a fully automatic mode without any user intervention. An issue in collecting Freesurfer segmentation results is the production of the cerebral cortex mask. There is a so-called “aseg” image and a “ribbon” image both recording voxels labeled as cerebral cortex. The “ribbon” data is what Freesurfer suggested [135] to use, but has more false negatives than the “aseg” data, while the latter is an intermediate result and has more false positives than the “ribbon” data. We applied a simple morphological closing operation on the union of the cortex ribbon and the subcortical structures so that certain true cerebral cortex voxels labeled in “aseg” but missed in “ribbon” are covered. This procedure apparently improved the performance of the cerebral cortex segmentation, as shown in table 9.2. We used the “closed” mask of cerebral cortex for our comparative evaluation.

FSL

In our first batch of experiments with FSL, we let FSL automatically extract the brain and perform brain tissue classification on both the BrainWeb and the IBSR data sets. However, FSL generated poor results on the brain extraction and brain tissue classification on 6 IBSR data sets (data set 5 to data set 10). In our second batch of experiments, we used different parameters in FSL, obtained better brain masks for these data, and repeated the brain tissue classification subsequently. Since the brain masks generated in the second batch of experiemnts are still not good enough, we turned to use the brain masks generated by Freesurfer for the brain tissue classification in FSL. This gives rise to best performance on the 6 IBSR data sets. The three batch of experiments on FSL show that the brain extraction algorithm of FSL is not robust on the IBSR data sets, but the brain tissue segmentation performed well given good brain masks. The performance of FSL on the 6 IBSR data set with respect to the three batch of experiments are shown in table 9.3.

TABLE 9.2: Dice metrics collected for Freesurfer over IBSR data sets

IBSR Data sets	Dice metrics with respect to cerebral cortex		
	“ribbon” data	“aseg” data	“closed” data
1	0.6905	0.7916	0.8039
2	0.7115	0.8061	0.8175
3	0.7390	0.7991	0.8362
4	0.8131	0.7838	0.8641
5	0.7361	0.7343	0.7794
6	0.7688	0.7301	0.8068
7	0.7421	0.7462	0.7888
8	0.7291	0.7327	0.7800
9	0.7780	0.7357	0.8128
10	0.7142	0.7336	0.7729
11	0.7839	0.8672	0.8702
12	0.7515	0.8045	0.8458
13	0.7504	0.8841	0.8647
14	0.6830	0.8316	0.8065
15	0.7272	0.8705	0.8487
16	0.7374	0.8514	0.8413
17	0.6626	0.8652	0.8076
18	0.7170	0.8485	0.8337
Mean	0.7353	0.8009	0.8212

TABLE 9.3: Dice metrics collected for FSL using different brain masks

Brain masks	Tissue types	Dice metrics over 6 IBSR data sets					
		5	6	7	8	9	10
Default FSL brainmask	Cerebral cortex	0.6591	0.6806	0.7268	0.6887	0.7713	0.6762
	Cerebral WM	0.8845	0.8928	0.8891	0.8335	0.9095	0.8792
Customized FSL brainmask	Cerebral cortex	0.7608	0.7735	0.7772	0.7740	0.7854	0.7859
	Cerebral WM	0.8722	0.8747	0.8766	0.8711	0.8953	0.8767
Freesurfer brainmask	Cerebral cortex	0.7312	0.7559	0.7898	0.7587	0.8277	0.7471
	Cerebral WM	0.8862	0.8953	0.8912	0.8914	0.9146	0.9009

SPM5

In our first batch of experiments with SPM5, we used the default parameters and let SPM5 automatically perform brain tissue segmentation on the BrainWeb and the IBSR data sets. In the second batch of experiments, we changed the parameter “Bias regularization” from the default “Very light regularization” to “Medium regularization” and reran SPM5 on the IBSR data sets. SPM5 is supposed to be used with greater bias regularization when it is known *a priori* that there is less intensity inhomogeneity in the image. Since the IBSR data sets were processed by bias field correction, the use of “Medium regularization” rather than the default “Very light regularization” improved the performance of SPM5 over almost all IBSR data sets, as shown in table 9.4. We use the best performance for each data set in our comparative evaluation.

TABLE 9.4: SPM5 experiments with different bias regularization

IBSR Data sets	Dice metrics			
	Cerebral cortex		Cerebral WM	
	Very light regularization	Medium regularization	Very light regularization	Medium regularization
1	0.7759	0.7705	0.8762	0.8927
2	0.8005	0.8048	0.8777	0.8940
3	0.8043	0.8080	0.8781	0.8936
4	0.8363	0.8356	0.8958	0.9013
5	0.4127	0.4621	0.7088	0.7312
6	0.4162	0.4207	0.7394	0.7422
7	0.7166	0.7065	0.8687	0.8734
8	0.7644	0.7670	0.8914	0.8923
9	0.7663	0.7595	0.9036	0.9059
10	0.6504	0.6868	0.8609	0.8744
11	0.8451	0.8396	0.8896	0.8927
12	0.8415	0.8541	0.8882	0.9014
13	0.8523	0.8588	0.8794	0.8955
14	0.8365	0.8426	0.8491	0.8678
15	0.8512	0.8439	0.8744	0.8880
16	0.8471	0.8263	0.8736	0.8948
17	0.8284	0.8407	0.8293	0.8551
18	0.8508	0.8539	0.8447	0.8764
Mean	0.7609	0.7656	0.8572	0.8707

BrainVisa

We tested BrainVisa on both the BrainWeb and the IBSR data sets automatically except for manually specifying landmark points including the AC point, the PC point, an inter-hemispheric point and a left hemisphere point. BrainVisa produced an empty brain mask in the 9th IBSR data set and was unable to generate brain masks for the 13th and the 18th data sets. In both cases, we set the Dice metrics to be 0.

TAS

TAS is tested on the BrainWeb and the IBSR data sets automatically except for the user to select the parameters in relative thresholding. The user only need to determine the best two relative thresholds (a relative threshold between GM and WM and another one between background and GM). Since the BrainWeb data sets have various noise level, we also use different smoothing scales in addition to the two relative thresholds for the BrainWeb data. Whenever the user changes the parameters for relative thresholding, the result can be generated in real time, so tuning relative thresholding is an efficient procedure.

9.3.3 Comparison

Segmentation Accuracy Over The IBSR Data Sets

We have collected the Dice metrics with respect to cerebral cortex and cerebral WM of the five packages on the IBSR data sets and the data is listed in table 9.5 and 9.6. Figure 9.1 through figure 9.8 compare the segmentation accuracy of TAS with SPM5, Freesurfer, FSL and BrainVisa respectively. Table 9.5 and 9.6 list the performance of the five packages over the 18 IBSR data sets and over the 13 good IBSR data sets respectively, which are also illustrated respectively in Figure 9.9 and figure 9.10 .

On average, TAS performed best on cerebral cortex segmentation over all 18 IBSR data sets, good and bad, and the 13 good data sets exclusively. In particular, TAS's cerebral cortex performance is consistently better than four other packages over the 13 good data sets except for the 4th data set, where TAS's performance is almost identical to the best, and

the 12th data set, where TAS's performance is close to the best. The cerebral cortex performance of the five packages over the five bad data sets are similar except that BrainVisa generated empty brain mask for the 18th data set.

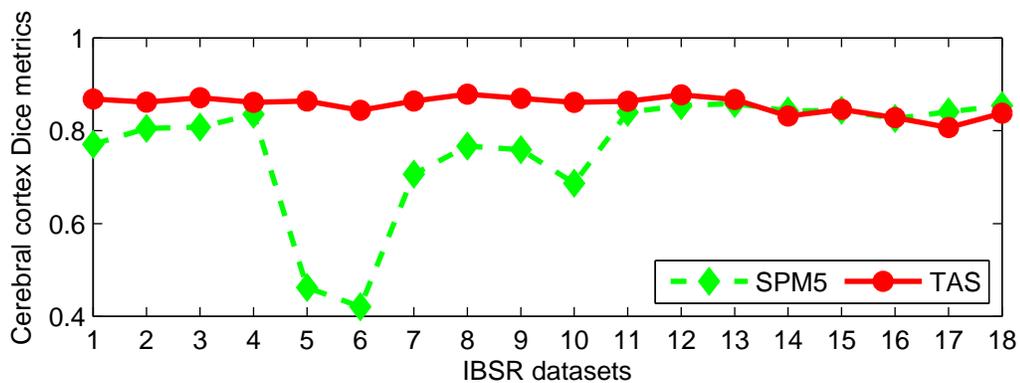
On average, FSL performed best on cerebral WM segmentation over both all 18 IBSR data sets and the 13 good data sets. However, TAS's performance is very close to FSL in both cases. The performance of the five packages over the five bad data sets are similar except that BrainVisa generated empty brain mask for the 18th data set and Freesurfer gave significantly lower performance for the 14th data set.

TABLE 9.5: Dice metrics of five tools with respect to cerebral cortex over the IBDR data sets

IBSR Data sets	Dice metrics with respect to cerebral cortex				
	BrainVisa	SPM5	Freesurfer	FSL	TAS
1	0.7461	0.7705	0.8039	0.7803	0.8682
2	0.7953	0.8048	0.8175	0.8121	0.8619
3	0.7674	0.8080	0.8362	0.8361	0.8714
4	0.7233	0.8356	0.8641	0.8028	0.8612
5	0.2875	0.4621	0.7794	0.7312	0.8638
6	0.6610	0.6610	0.8068	0.7559	0.8441
7	0.7108	0.7065	0.7888	0.7898	0.8638
8	0.6982	0.7670	0.7800	0.7587	0.8790
9	0	0.7595	0.8128	0.8277	0.8700
10	0.7707	0.6868	0.7729	0.7471	0.8611
11	0.8688	0.8396	0.8702	0.8833	0.8634
12	0.8596	0.8541	0.8458	0.8582	0.8772
13	0	0.8588	0.8647	0.8554	0.8673
14	0.8406	0.8426	0.8065	0.8429	0.8315
15	0.8441	0.8439	0.8487	0.8381	0.8457
16	0.8260	0.8263	0.8413	0.8426	0.8281
17	0.8445	0.8445	0.8076	0.8275	0.8070
18	0	0.8539	0.8337	0.8278	0.8379
Mean	0.6247	0.7656	0.8212	0.8121	0.8557
Mean over 13 good data sets	0.6068	0.7365	0.8187	0.8030	0.8656

TABLE 9.6: Dice metrics of five tools with respect to cerebral WM over the IBDR data sets

IBSR Data sets	Dice metrics with respect to cerebral WM				
	BrainVisa	SPM5	Freesurfer	FSL	TAS
1	0.8652	0.8927	0.7964	0.8971	0.8789
2	0.8899	0.8940	0.8208	0.9160	0.8926
3	0.8649	0.8936	0.8138	0.9084	0.8874
4	0.8596	0.9013	0.8489	0.9168	0.9070
5	0.4097	0.7312	0.9240	0.8862	0.9101
6	0.7970	0.7422	0.9115	0.8953	0.8954
7	0.8255	0.8734	0.9147	0.8912	0.9001
8	0.8110	0.8923	0.9203	0.8914	0.9170
9	0	0.9059	0.9179	0.9146	0.9224
10	0.8457	0.8744	0.9069	0.9009	0.8790
11	0.8975	0.8927	0.8711	0.9142	0.8969
12	0.8858	0.9014	0.8099	0.8988	0.9006
13	0	0.8955	0.8647	0.8673	0.8748
14	0.8613	0.8678	0.7824	0.8632	0.8622
15	0.8541	0.8880	0.8746	0.8743	0.8637
16	0.8792	0.8948	0.8479	0.8933	0.8713
17	0.8619	0.8551	0.8592	0.8585	0.8463
18	0	0.8764	0.8124	0.8323	0.8463
Mean	0.6893	0.8707	0.8610	0.8900	0.8862
Mean over 13 good data sets	0.6886	0.8685	0.8708	0.8999	0.8971

**FIGURE 9.1:** Cerebral cortex Dice metrics of SPM5 and TAS on the IBSR datasets.

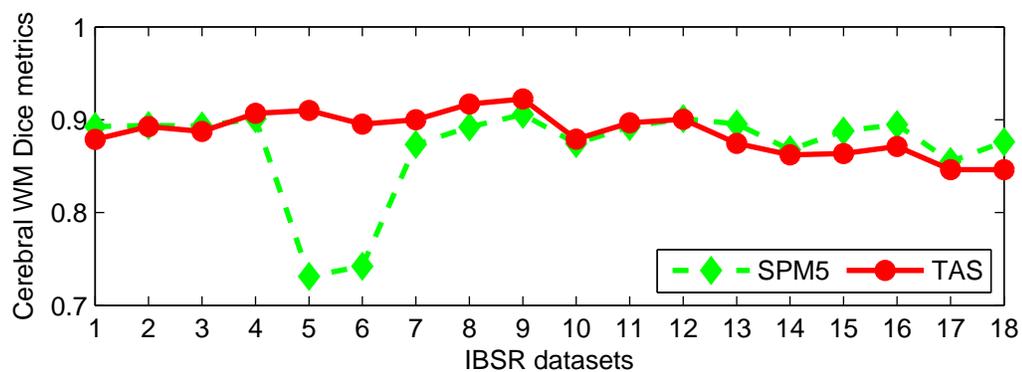


FIGURE 9.2: Cerebral WM Dice metrics of SPM5 and TAS on the IBSR datasets.

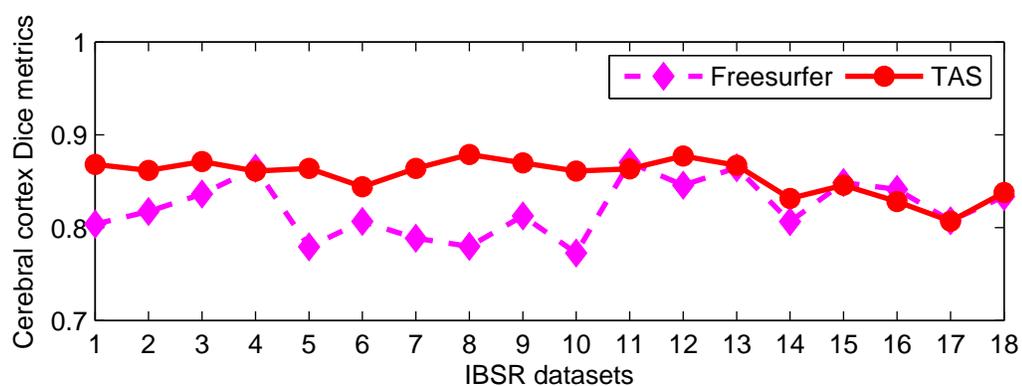


FIGURE 9.3: Cerebral cortex Dice metrics of Freesurfer and TAS on the IBSR datasets.

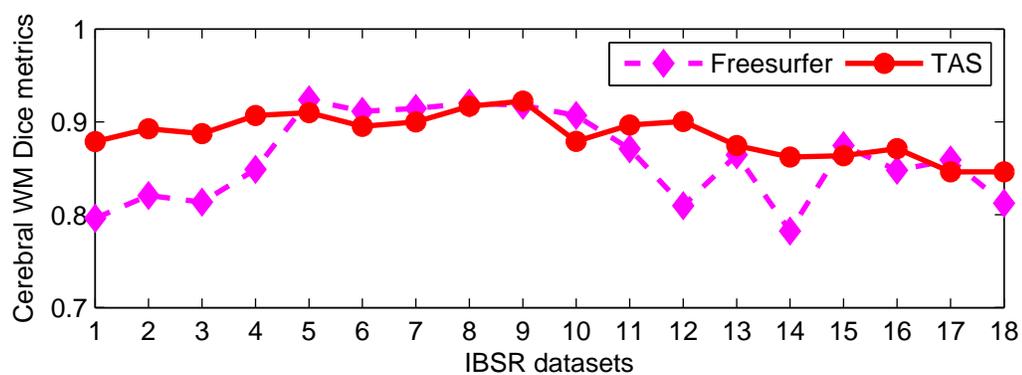


FIGURE 9.4: Cerebral WM metrics of Freesurfer and TAS on the IBSR datasets.

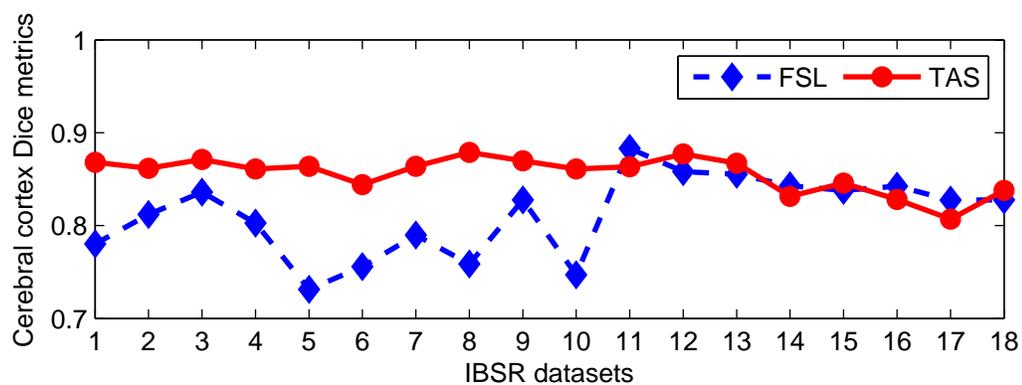


FIGURE 9.5: Cerebral cortex Dice metrics of FSL and TAS on the IBSR datasets.

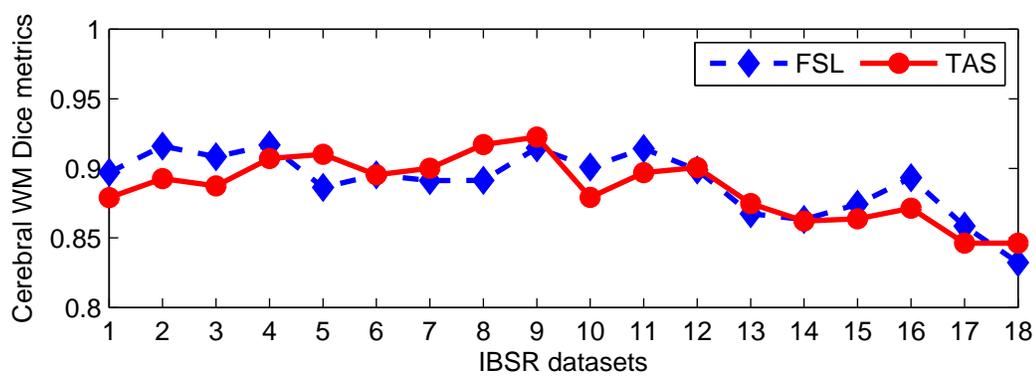


FIGURE 9.6: Cerebral WM Dice metrics of FSL and TAS on the IBSR datasets.

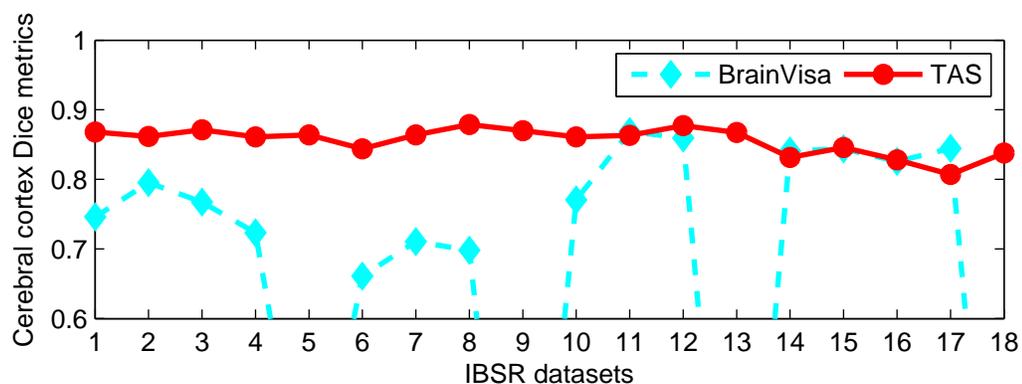


FIGURE 9.7: Cerebral cortex Dice metrics of BrainVisa and TAS on the IBSR datasets.

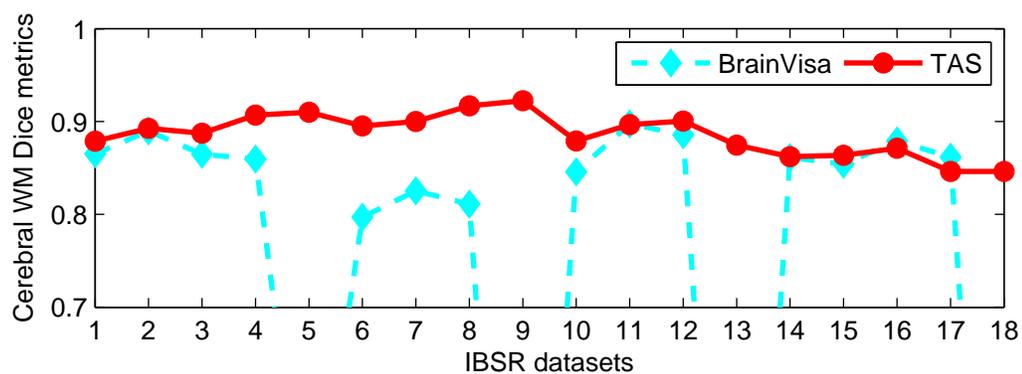


FIGURE 9.8: Cerebral WM metrics of BrainVisa and TAS on the IBSR datasets.

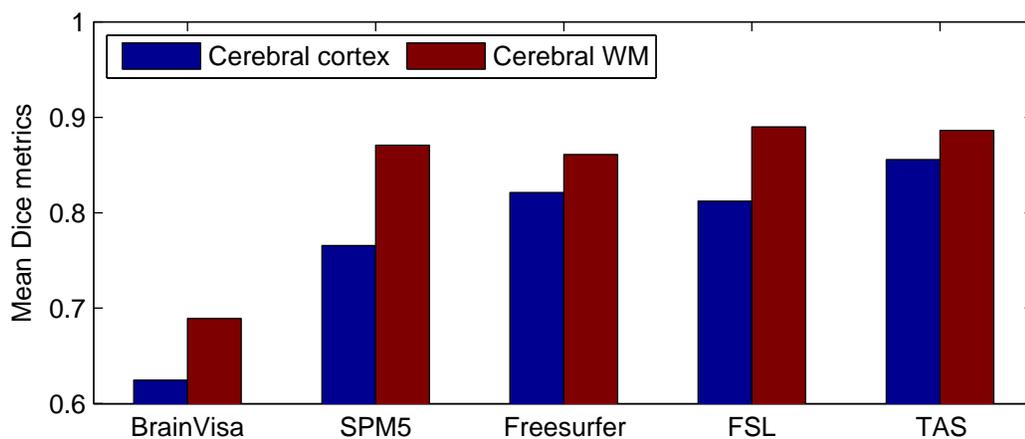


FIGURE 9.9: Mean Dice metrics of five tools on the IBSR datasets

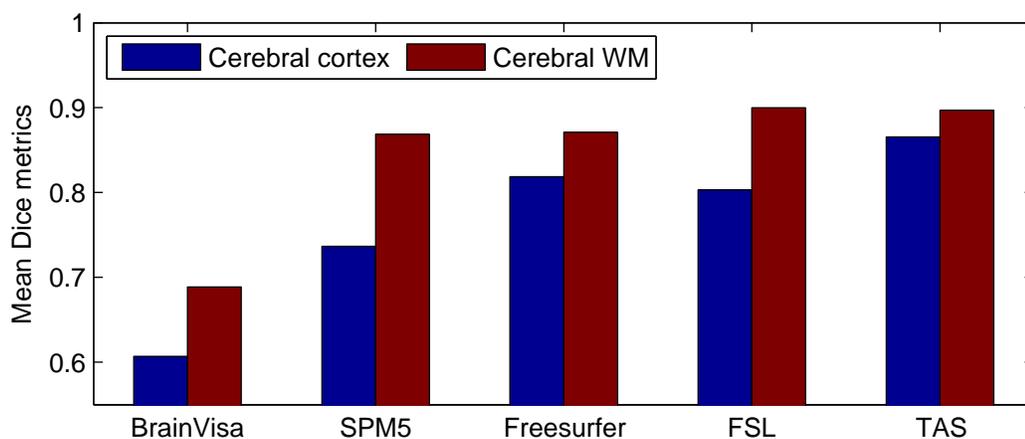


FIGURE 9.10: Mean Dice metrics of five tools on the 13 good IBSR datasets

Segmentation Robustness Over The IBSR Data Sets

We calculated the standard deviations of the Dice metrics over the IBSR data sets and use them together with the mean Dice metrics as the measurements of the segmentation robustness of the five packages over MR images scanned from different subjects. Greater mean Dice metric and lower standard deviation indicate greater robustness with respect to segmentation accuracy over a set of data.

Two groups of the standard deviations are calculated over the total 18 IBSR data sets and over the 13 good IBSR data sets. TAS demonstrated lowest standard deviation with respect to cerebral cortex over both the total 18 IBSR data sets and the 13 good IBSR data sets, as shown in table 9.7, and figure 9.11 and 9.12. The lowest mean and standard deviation of Dice metric with respect to cerebral cortex indicate that TAS possesses the best accuracy robustness with respect to cerebral cortex over the IBSR data sets. For cerebral WM, TAS and FSL performed neck and neck with respect to both the mean and the standard deviation of the Dice metric over both the total 18 IBSR data sets and the 13 good IBSR data sets. TAS and FSL tied for the best accuracy robustness with respect to cerebral WM over the IBSR data sets. Considering both cerebral cortex and cerebral WM, we think that TAS performed most robustly on average over the entire IBSR data sets.

TABLE 9.7: Standard deviation of Dice metrics of five tools over the IBSR data sets

Sample groups		Standard deviations				
data sets	tissue type	BrainVisa	SPM5	Freesurfer	FSL	TAS
all IBSR data sets	Cerebral cortex	0.3155	0.1284	0.0308	0.0429	0.0192
	Cerebral WM	0.3351	0.0505	0.0473	0.0230	0.0225
13 good data sets	Cerebral cortex	0.3047	0.1414	0.0345	0.0476	0.0087
	Cerebral WM	0.3305	0.0593	0.0482	0.0143	0.0147

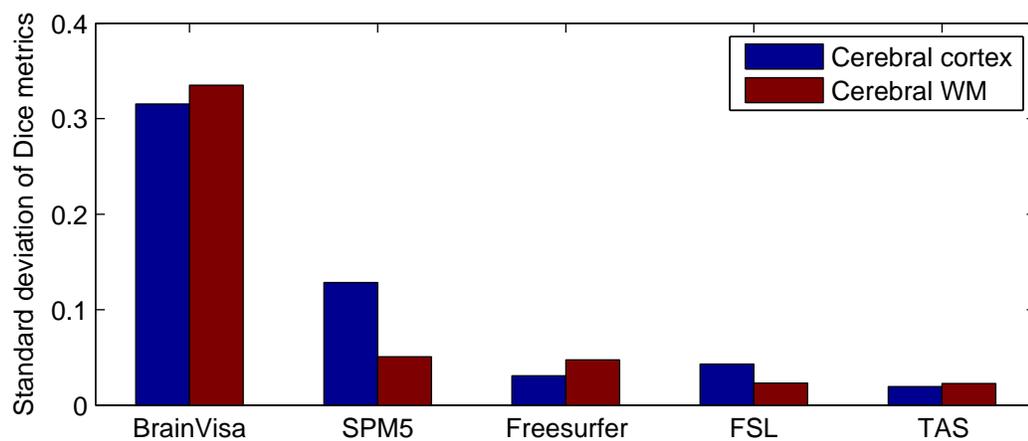


FIGURE 9.11: Standard deviations of Dice metrics of five tools on the IBSR datasets

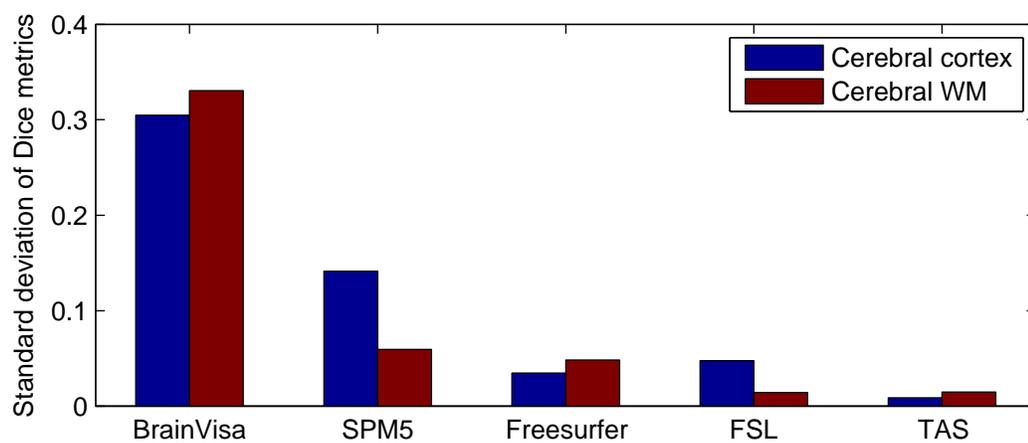


FIGURE 9.12: Standard deviations of of Dice metrics of five tools on the 13 good IBSR datasets

Segmentation Robustness With Respect To Noise And IIH Over The BrainWeb Data Sets

As described in subsection 9.2.1, the BrainWeb data sets vary in noise levels and intensity inhomogeneity (IIH) levels. The performance in Dice metrics of the five package with respect to cerebral GM and cerebral WM is listed in table 9.8 and 9.9 and illustrated in figure 9.13 and figure 9.14.

TABLE 9.8: Dice metrics of five tools with respect to cerebral GM over the BrainWeb data sets

BrainWeb Data sets		Dice metrics with respect to cerebral GM				
Noise level	IIH level	BrainVisa	SPM5	Freesurfer	FSL	TAS
3%	20%	0.9292	0.9173	0.8333	0.9242	0.9084
	40%	0.9247	0.9189	0.8342	0.9268	0.9086
5%	20%	0.9197	0.8989	0.8323	0.9193	0.8908
	40%	0.9201	0.8998	0.8323	0.9193	0.8858
7%	20%	0.8628	0.8673	0.8320	0.9113	0.8816
	40%	0.8740	0.8713	0.8312	0.9127	0.8827
9%	20%	0.8166	0.8255	0.8259	0.8996	0.8658
	40%	0.7836	0.8301	0.8264	0.9019	0.8678

TABLE 9.9: Dice metrics of five tools with respect to cerebral WM over the BrainWeb data sets

BrainWeb Data sets		Dice metrics with respect to cerebral GM				
Noise level	IIH level	BrainVisa	SPM5	Freesurfer	FSL	TAS
3%	20%	0.9550	0.9471	0.8849	0.9672	0.9588
	40%	0.9599	0.9533	0.8889	0.9664	0.9593
5%	20%	0.9552	0.9314	0.8824	0.9567	0.9494
	40%	0.9534	0.9315	0.8863	0.9581	0.9476
7%	20%	0.9325	0.8978	0.8779	0.9448	0.9382
	40%	0.9311	0.9008	0.8796	0.9467	0.9370
9%	20%	0.8926	0.8656	0.8757	0.9332	0.9296
	40%	0.8748	0.8701	0.8740	0.9354	0.9289

Among the five packages, Freesurfer demonstrated lowest performance variation over different noise levels; SPM5 and BrainVisa neck and neck have highest performance variations over different noise levels; and TAS and FSL have medium performance variations over different noise levels, compared to the other three. Although Freesurfer performed

consistently over different noise levels, it also gave results with lowest accuracy on average. For each of the four noise levels, we also tested the packages on images with two different IIH levels. All five packages gave little variation over different IIH levels. The only exception is for BrainVisa to handle with $N=9\%$ and $IIH=40\%$. This is due to a poor brain mask.

It is worth noting that in real MR scans, the intensity inhomogeneity may be in various and unknown patterns and could occur together with other difficulties that may be not present in the simulated BrainWeb data sets. Therefore, we remark that our experiments with the BrainWeb data set do not mean to give a thorough and sufficient evaluation on the five package with respect to the IIH robustness.

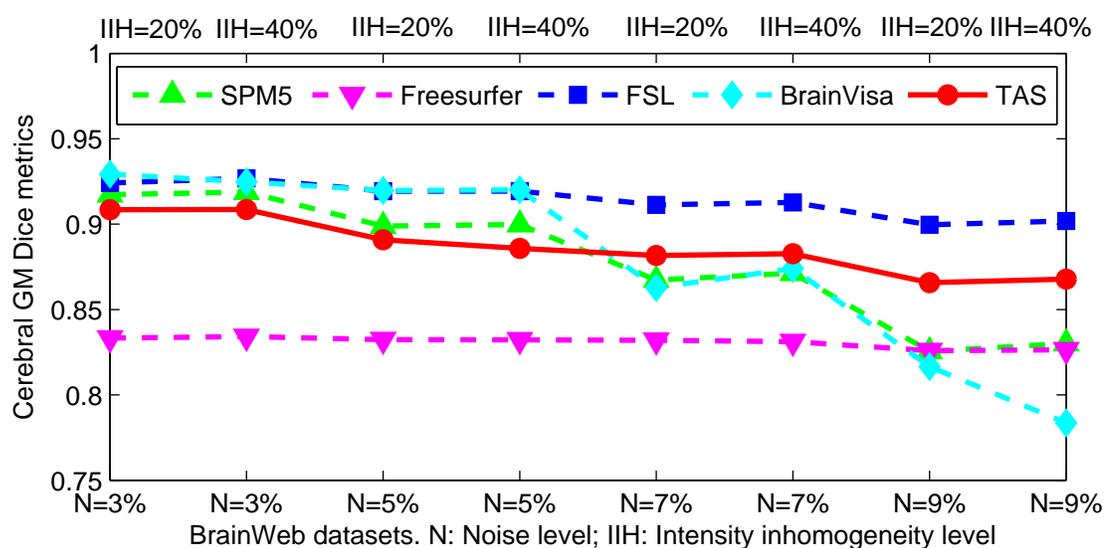


FIGURE 9.13: Cerebral GM Dice metrics of five tools on the BrainWeb datasets.

Computational Efficiency

The execution times of the five packages tested on the IBSR data sets and the BrainWeb data sets are listed in table 9.10. The experiments were all run on a single 2.8Ghz Intel Xeon processor. Among the five packages, BrainVisa took least amount of time but is also associated with the lowest segmentation accuracy and robustness on the IBSR data

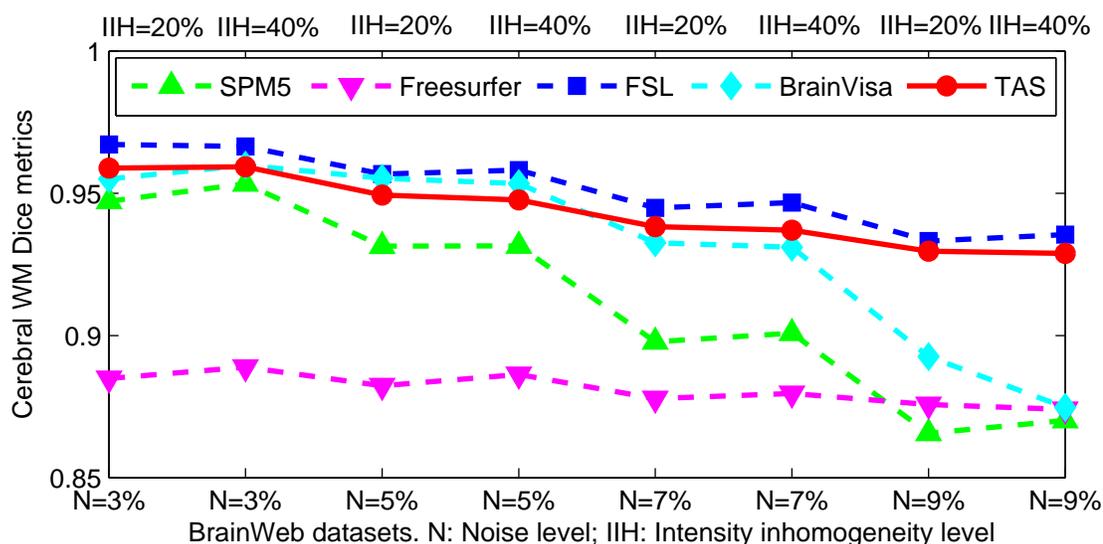


FIGURE 9.14: Cerebral WM Dice metrics of five tools on the BrainWeb datasets.

sets. Freesurfer took much longer execution time than other four but it should be acknowledged that the longer time span covers segmentation of more subcortical structures and reconstruction of cortical surfaces. TAS took much less time than Freesurfer, but required more than BrainVisa, FSL and SPM5. However, it should be noted that the most of the TAS time was spent for topology correction, which was not counted in the execution times of the other three.

TABLE 9.10: Computation times of five packages on the IBSR and the BrainWeb datasets

Data sets	Computation times					
	BrainVisa	SPM5	Freesurfer	FSL	TAS	TAS(topology correction)
IBSR	1.5m	34m	27.2h	5m	17m	14m
BrainWeb	1.6m	20m	24.5h	9m	21m	18m

9.4 Qualitative Evaluation

In this section, we give a qualitative evaluation of the five packages based on the experiments of the packages on the IBSR data sets, Brainweb data sets, and the auxiliary 8

pathological data sets with mild cognitive impairment or Alzheimers disease. We first summarize and compare the segmentation functionalities of the five packages followed by the discussion of their automaticity. Finally, we present various segmentation abnormalities of each package that we observed in the experiments. The segmentation abnormalities are presented in two groups: those that occurred in packages other than TAS (i.e. TAS pros) and those that occurred in TAS (i.e. TAS cons).

9.4.1 Segmentation Functionalities

The following is a summarization and comparison on the main segmentation features of the five packages.

- Bias field correction: Freesurfer, SPM5, FSL, and BrainVisa, all integrate a bias field correction procedure, either prior to tissue classification or combined with the classification. TAS, on the other hand, does not need explicit bias field correction and the relative thresholding is robust to bias field in arbitrary patterns.
- Brain extraction: FSL, Freesurfer and BrainVisa provide separate tools for brain extraction (i.e. skull stripping) prior to brain tissue classification while SPM5 combine brain extraction together with tissue classification. TAS, on the other hand, performs cerebrum extraction after tissue classification. Note that the brain mask generated by BrainVisa is supposed to contain only GM and WM while the brain mask generated by FSL and Freesurfer is supposed to contain CSF as well as GM and WM.
- Tissue classification: FSL and SPM5 segment the brain volume into three tissue types: CSF, GM, and WM. BrainVisa and TAS extract cerebral WM and cerebral GM. BrainVisa also provides cerebral hemisphere partition. Freesurfer segments a whole brain into 37 individual structure including cerebral cortex, cerebral WM, a set of subcortical structures, brainstem and cerebellar structures.
- Cortical surface reconstruction: BrainVisa, Freesurfer and TAS support cortical surface reconstruction while FSL and SPM5 do not. A core mechanism involved in the surface reconstruction is to make sure that topology of the cortical surfaces is correct.

9.4.2 Segmentation Automaticity And User Intervention

All five packages support highly automatic brain segmentation without or with little user intervention. Freesurfer allows the user to start the cortical surface reconstruction without any intervention. In case the segmentation is not satisfiable, Freesurfer supports interactive tools for the user to modify the brain mask and add control points to improve the intensity normalization of WM, a procedure extremely important for the performance of Freesurfer's whole brain segmentation. Freesurfer also supports interactive tools for editing the final results generated by the automatic processing.

FSL also allows the user to start the segmentation without any intervention. In FSL, brain extraction and tissue classification are performed respectively by BET and FAST. The BET performance substantially influences that of FAST. If the user is not satisfied with the brain extraction, FSL allows the user to select different parameters and rerun BET. However, our experiments with BET on the IBSR data sets and the pathological data sets show that BET cannot guarantee good brain extraction even with user intervention. FAST has custom options for the user to select whether to use the k-means segmentation or *a priori* probability maps for initial segmentation and to guide the k-means segmentation with manual intervention.

In SPM5, brain segmentation can also be automatically started with the default parameters and SPM5 often generates good results. An important custom parameter of SPM5 is the one that control the extent of bias field regularization. When any parameter is changed, the segmentation procedure has to be started over from scratch.

BrainVisa requires the user to prepare the data by first specifying several landmark points including the AC point, the PC point, an inter-hemispheric point and a left hemisphere point. When the data is prepared by the user, BrainVisa automatically performs segmentation. BrainVisa supports interactive tools for the user to edit the segmentation results.

TAS recommends the user to first determine the two relative thresholds and occasionally the smoothing scale for relative thresholding, which can be performed in a real-time procedure, and starts the subsequent segmentation without any user intervention. When the user want to process a set of data which are apparently acquired with the same or similar

parameters, the user can determine the relative thresholding parameter for only one of the data sets and use them for the rest. In our experiments, for example, we used the same set of parameters for the 8 pathological data sets.

Compared to the user intervention mechanisms in the other four packages, user intervention in TAS is in the form of global parameter selection and has the following advantages. First, it is straightforward and requires little even no expertise to understand the meaning of the parameters and the criterion for selecting optimal ones. Second, it is very easy to operate by sliding a value bar. Third, it is very efficient and the user can obtain the effect of parameter selection in real time. Fourth, the parameters have global effect for segmentation and the user does not need to repeat similar operations for different local regions.

9.4.3 Segmentation Abnormalities

TAS Pros

First of all, we are interested in why TAS consistently gives better performance with respect to cerebral cortex segmentation over the 13 good IBSR data sets. By examining the segmentation results, we found that there were a “shrinking” effect on the cerebral cortex segmentation for Freesurfer, FSL, SPM5 and BrainVisa, which gives rise to significant amount of false negatives, while this problem did not occur or was much milder in TAS. We think the underlying reason is that TAS uses a new image modeling mechanism that can adapt to wider variations of GM intensities while the statistical methods used in other packages were misled by such variations and missed a great deal of GM voxels with lower intensities. This phenomenon is shown in figure 9.15 with a representative IBSR data set and the segmentation results of the five packages.

In figure 9.15, pink stands for correct WM, red for false WM negative, light green for correct GM, very light green for false WM negative and false GM negative, dark green for false GM positive, and gray false for GM negative. Note that for SPM5, light green represents correct GM segmentation, darker green represents false GM negative, and gray level represents false GM positive. The differences between the segmentation of TAS and

other four packages on the same subject are shown in figure 9.16. Note that the green part is mainly due to the false positives of TAS, particularly around the lateral ventricles. It can be seen that all the other four packages have significantly more errors (mainly false negatives), colored in red, than TAS around the cerebral cortex.

Another common problem in FSL, BrainVisa and SPM5 is the poor brain extraction. Some examples are shown in figure 9.17. Poor brain mask is mainly responsible for the poor performance for SPM5 and BrainVisa over the IBSR data sets indicated as the valley points in figure 9.1 and 9.2. Since we used the relatively better brain masks generated from Freesurfer for FSL, there are no deep valleys of performance for FSL in figure 9.5 and 9.6. Freesurfer did not encounter poor skull stripping, but unclean brain mask may still be generated, as shown in figure 9.17(b) where some non-brain voxels with high intensities are taken as brain tissues. TAS on the other hand, does not depend on a brain extraction pre-processing step and robustly generated clean cerebrum masks as the union of the cerebral white matter and the cerebral gray matter over all tested data sets.

We also found some other interesting abnormalities with Freesurfer, as shown in figure 9.18. For example, in figure 9.18(c) Freesurfer cut off a significant amount of cerebral WM and cortex at the top of the brain. In figure 9.18(d) Freesurfer was unable to correctly recognize the complete lateral ventricle of the subject with Alzheimers disease. In figure 9.19, Freesurfer generated poor GM/WM segmentation even for a simulated image with excellent quality (noise level is 3% and IIH level is 20%) while TAS and FSL generated excellent results. These abnormalities, we believe, are probably because of the over-regularization of the *a priori* probability maps used in Freesurfer.

TAS Cons

We found two types of abnormalities in the TAS segmentation results. One is the consistently existence of a rim of GM around the lateral ventricles, as shown in figure 9.19(d) and 9.15(f). This is associated with our structure modeling of CSF, GM and WM as a layered structure. This abnormality actually also occurs consistently with SPM5. Another abnormality is that sometimes a significant amount of voxels in the amigdala area are missed in the segmentation, as shown in figure 9.20. This is mainly due to the fact that

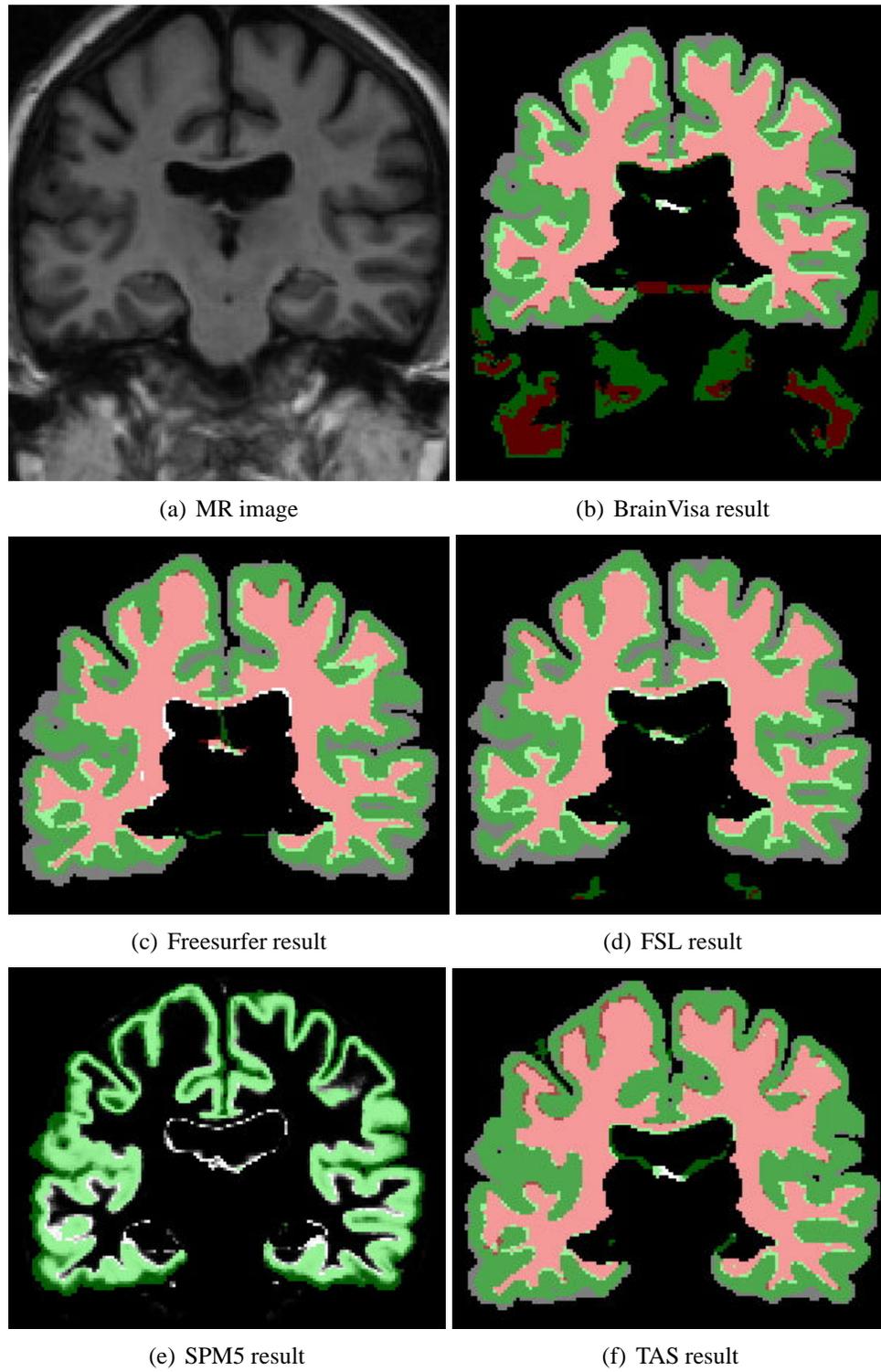


FIGURE 9.15: GM-shrinking phenomenon.

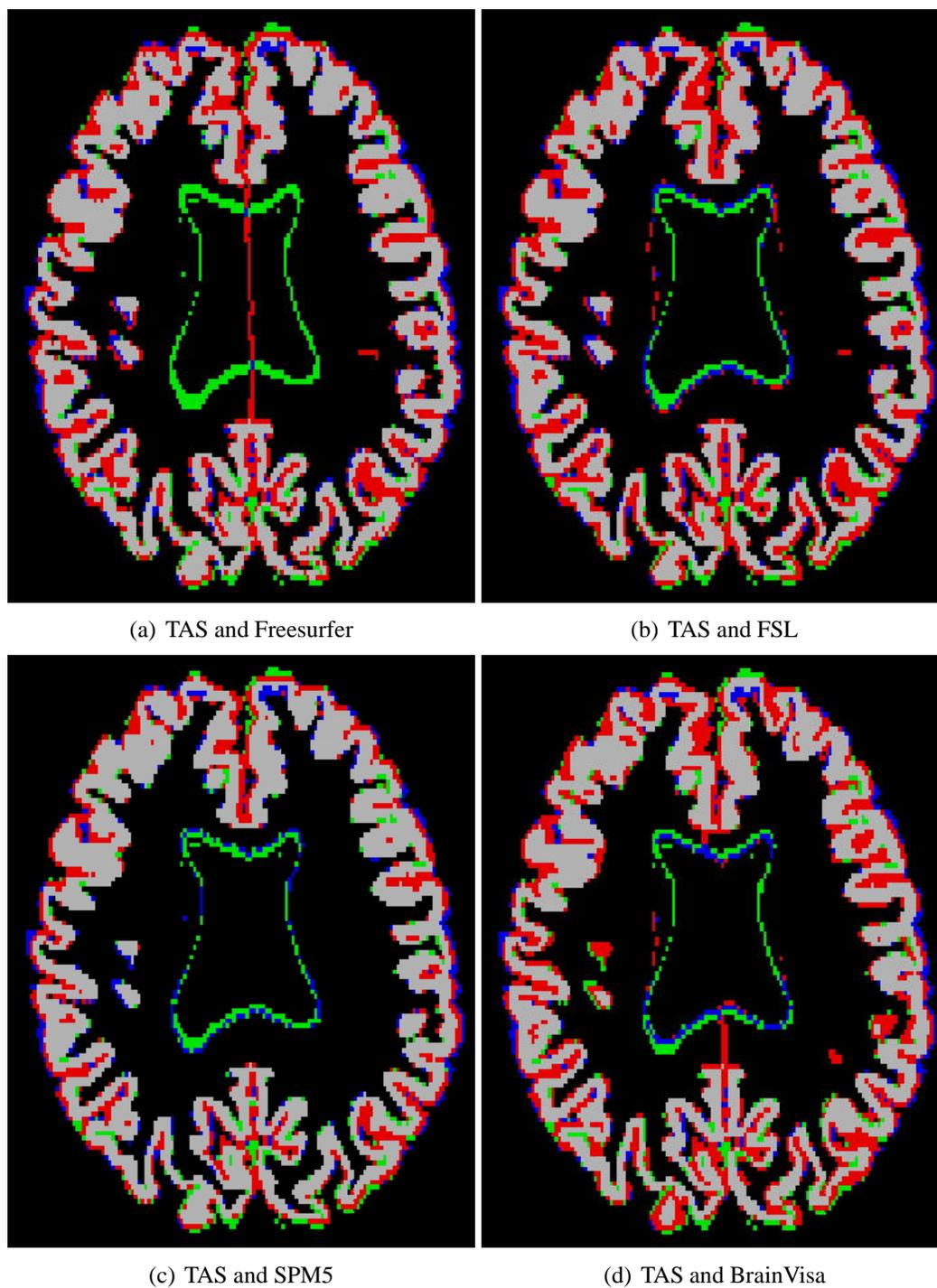


FIGURE 9.16: Segmentation differences between segmentation of TAS and other four tools. Black: both correct non-GM; gray: both correct GM; green: TAS incorrect while other correct; red TAS correct while other incorrect; Blue: both incorrect.

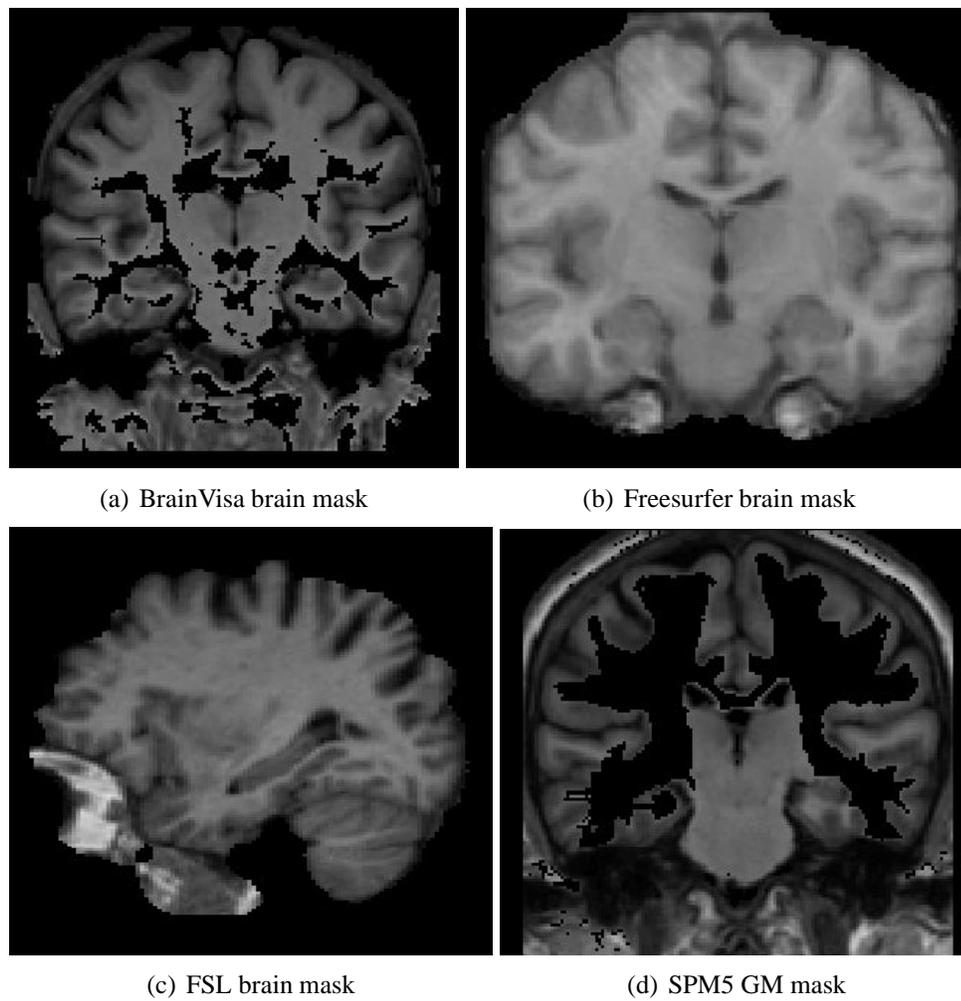


FIGURE 9.17: Poor brain masks

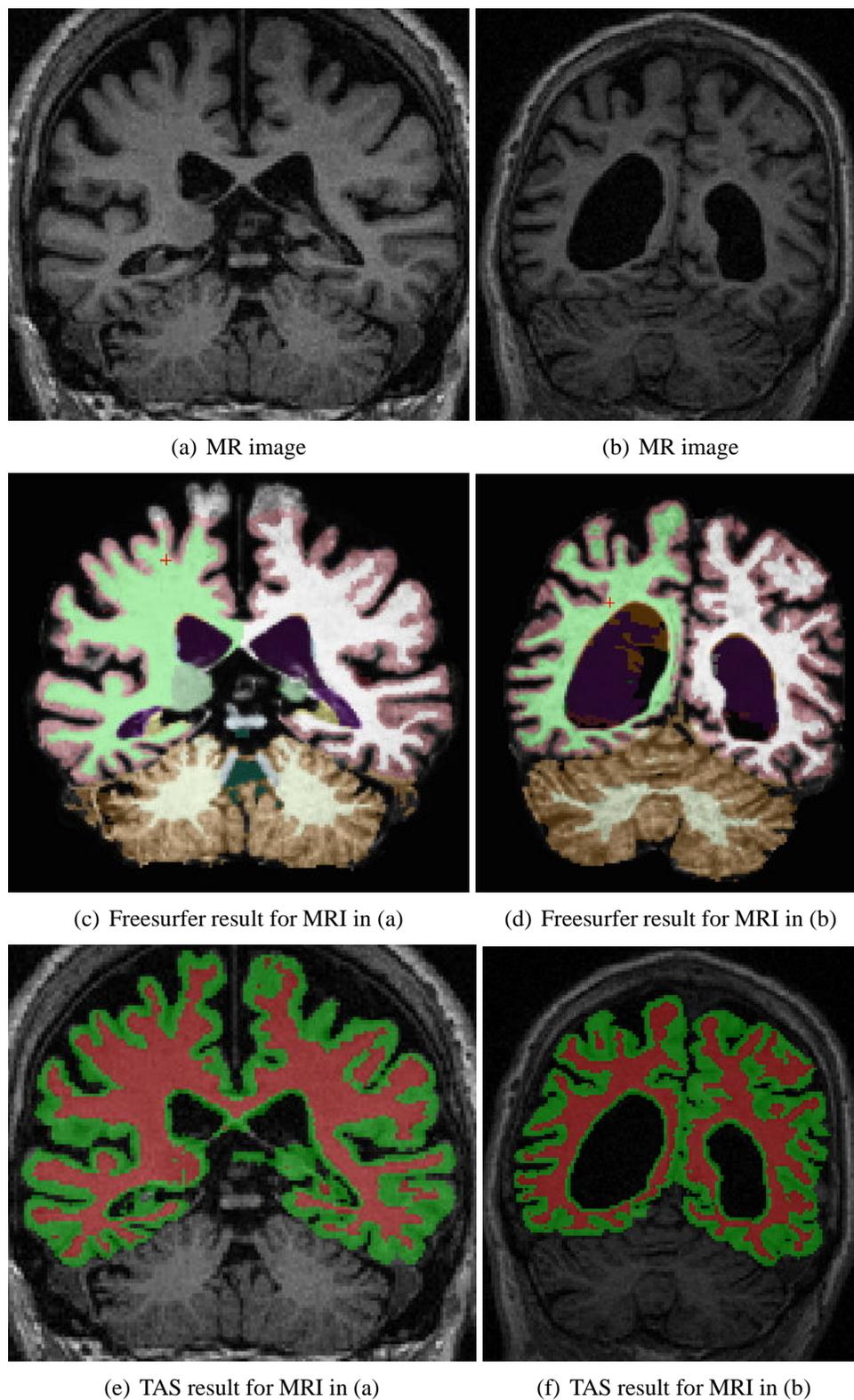


FIGURE 9.18: Freesurfer abnormalities in the pathological data sets and comparison with TAS

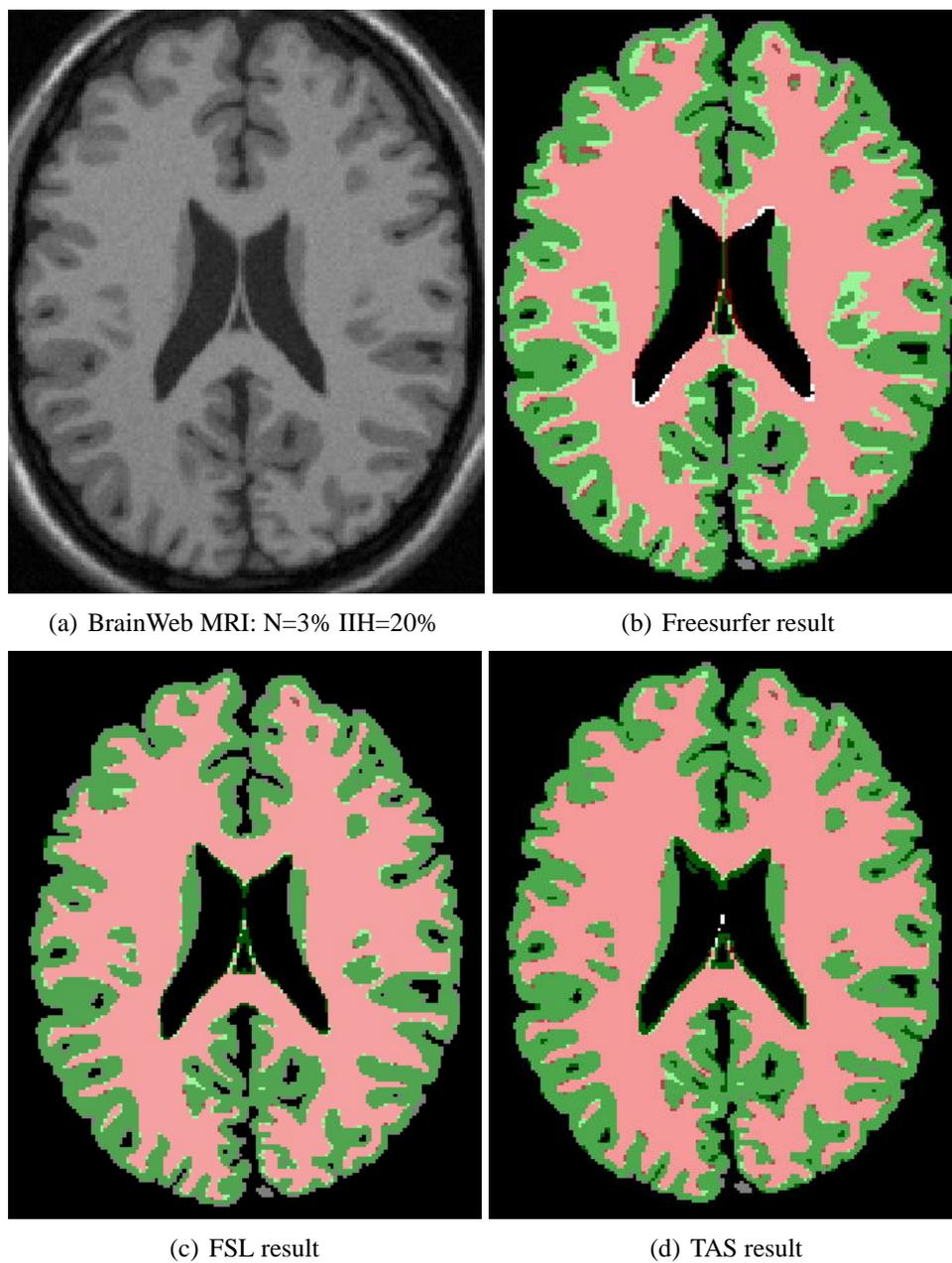


FIGURE 9.19: Freesurfer segmentation on the BrainWeb MRI with comparison to TAS and FSL

the gray matter is usually thicker than average cerebral cortex and that the white matter surrounded by the gray matter in the amigdala area is sometimes too thin for TAS to obtain good segmentation.

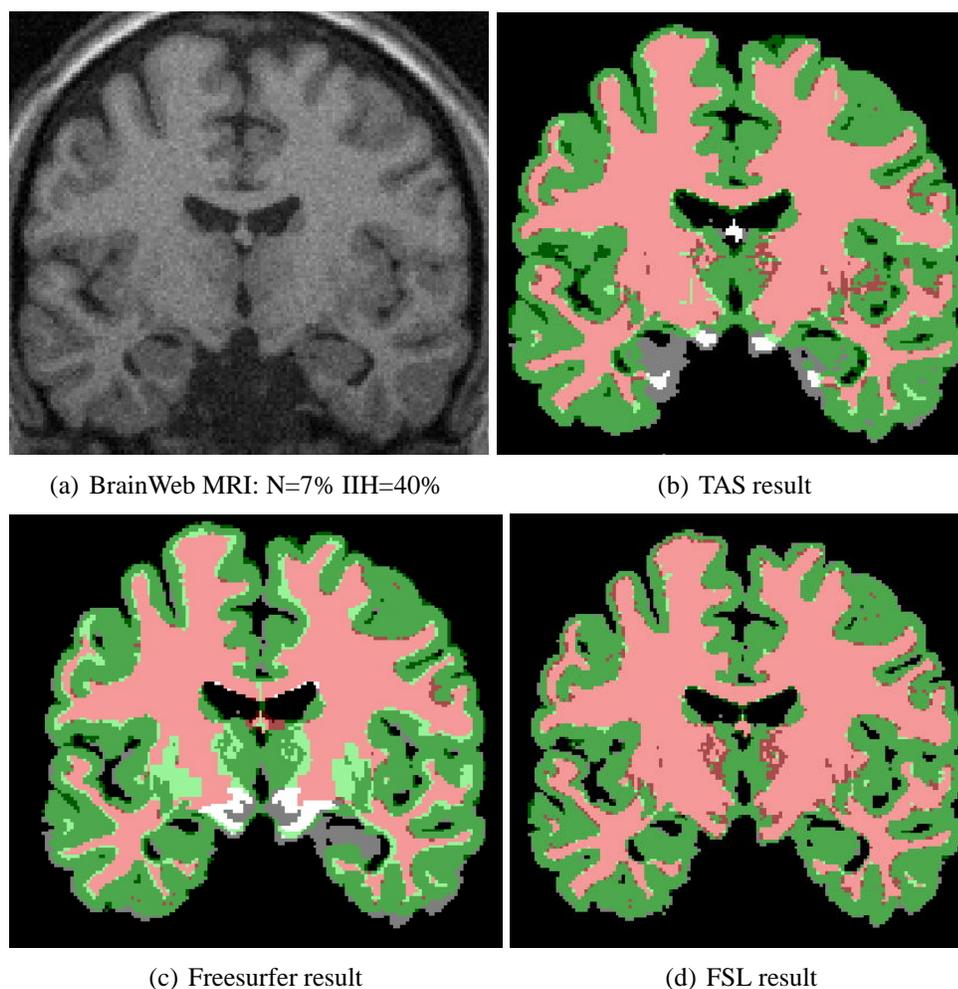


FIGURE 9.20: TAS segmentation missing part of amigdala on the BrainWeb MRI with comparison to Freesurfer and FSL

Unlike the abnormalities of other packages that often occurred at the cerebral cortex, the two abnormalities with TAS have little or no adversary influence to cerebral cortex segmentation and cortical surface reconstruction. These abnormalities have been well located and we know why they occur. It is part of our future work to eliminate these abnormalities in TAS.

CHAPTER X

Conclusion

10.1 Summary Of The Segmentation Pipeline

In this dissertation, we presented an image analysis pipeline for neuroanatomical MR image segmentation, which consists of an initial brain tissue classification procedure with relative thresholding complemented by terrain analysis, a series of segmentation error correction procedures, and a procedure for cortical surface reconstruction. Segmentation error correction includes a multiscale morphological topology correction procedure for white matter and two non-cerebrum tissue elimination procedures for extraction of cerebral white matter and cerebral gray matter respectively. The topology correction algorithm and the cerebrum extraction procedure depend on several generic morphological and morphometric analysis algorithms, particularly a 3D curve skeletonization algorithm and its variants, and a cell-complex-based morphometric analysis algorithm.

10.2 Segmentation Performance

We have evaluated our segmentation method quantitatively and qualitatively on various MR images including simulated and real, normal and pathological. We also compared

our segmentation results with those of four leading segmentation tools. The comparison demonstrated that our method tends to produce more accurate segmentation on cerebral gray matter segmentation and comparable segmentation on cerebral white matter to the leading packages. In addition, our segmentation method exhibited least performance fluctuation across different subjects. Quantitative evaluation of our segmentation method on different noise levels and different intensity inhomogeneity levels also demonstrated high segmentation robustness of our method. It also produced least segmentation abnormalities with respect to cerebral cortex segmentation among the compared tools. Our method took about 20 minutes on average for cortical surface reconstruction and is computationally efficient compared to other tools. Finally, our segmentation method is highly automatic in that little or no user intervention is required to produce accurate and robust segmentation in limited time.

10.3 Contributions

The main contribution of this dissertation can be characterized as the presentation of a work flow for neuroanatomical MR image segmentation in which brain tissue classification is conducted prior to brain extraction and is independent of explicit bias field correction, design and implementation of a set of original algorithms that were applied in different stages in the work flow, and a comparative evaluation that demonstrated that our method is highly accurate, robust, automatic and computationally efficient.

The relative thresholding algorithm is based on a new structure modeling of neuroanatomy and a new image modeling of the T1-weighted MR images exploiting various structural, geometrical and radiological *a priori* knowledge. Brain tissue classification with relative thresholding is free from three typical problems that occur in traditional intensity based segmentation methods. First, it is independent of prior brain extraction and thus avoids performance instabilities caused by poor brain extraction in many traditional methods. Second, relative thresholding is robust against intensity inhomogeneities without explicit bias field correction. Third, relative thresholding is also able to adapt to large intensity vari-

ations within a given brain tissue and thus tends to produce more accurate segmentation. On the other hand, relative thresholding can be seen as a special edge (or intensity difference) based segmentation method that overcomes several critical disadvantages of edge based segmentation approaches. First, it produces coherent regions labeled with brain tissue types. Second, it is able to recognize blurred edges and tissue boundaries where intensities vary smoothly. Third, it is able to suppress spurious edges between voxels of same tissue types. In these respects, we see relative thresholding as a fusion of intensity based segmentation and edge based segmentation. The idea can be effective in other image segmentation problems, particularly where there are intensity inhomogeneities and blurred edges.

The cell complex based morphometric analysis simplifies a 3D object into a 1D structure and gives a quantitative measurement on the wideness and connectivity on every location in the 3D object. This is a significant advancement over the fact that traditionally only a “thickness” metric (i.e. the distance to the boundary) can be calculated for each point in the 3D object. This new 3D morphometric instrument will potentially promote more applications of morphological analysis for various problems in computer vision and image understanding. By applying this new morphometric analysis on the white matter generated, we are able to eliminate non-brain tissues and divide the entire white matter at the brain stem based on the *a priori* knowledge of strong connectivity of cerebral white matter. Cerebrum extraction using cell-complex-based morphometric analysis provides higher robustness than other brain extraction such as traditional morphological image analysis, deformable model based methods, and atlas based methods.

The white matter topology correction algorithm is based on the *a priori* observation that human white matter, particularly cerebral white matter, is a surface-like object. Preservation of this morphological property is taken as the major criterion for eliminating topology defects. In addition, our topology correction algorithm involves WM, GM and background in the procedure, in contrast to the traditional procedure where only the foreground and background are involved. Our three-fold procedure exploiting the surface-likeness morphological *a priori* knowledge tends to more robustly produce reasonable solutions to topology defect elimination than other methods.

The 3D curve skeletonization algorithm is performed directly on a 3D object in contrast to the traditional methods that depend on prior surface skeletonization and tend to generate skeletons with better “medialness”. A variant of our curve skeletonization is referred to as “shape and topology preserving erosion” in which the skeletonization procedure is conducted in certain iterations instead of until conversion. This variant algorithm is used as an important component in the white matter topology correction algorithm as well as for generating topology correct gray matter in our neuroanatomical pipeline. A similar procedure can also be used in 3D object smoothing to eliminate noisy protrusions on the 3D object. This framework of 3D curve skeletonization is based on a systematic point classification of discrete 3D objects. In this classification approach, we proposed the central notion of a *thick-simple point*. This notion enables deeper and wider topology and geometry characterization of any points in a 3D digital object.

As a brief summary of our contributions in this dissertation, we essentially opened a new window on the general methodology for neuroanatomical segmentation in MRI and proposed new perspectives on particular issues such as brain extraction, bias field correction, brain tissue classification, image modeling, topology correction, as well as morphological and morphometric analysis. These new thoughts and the practice in the specific problem solving for neuroanatomical MR image segmentation are also meaningful in a wider area of computer vision and image understanding.

10.4 Future Work

First, as addressed in our qualitative evaluation (subsection 9.4.3), there is consistently a rim of gray matter around the lateral ventricles and partial amigdala areas that may be missed in our cerebrum segmentation. Future work will recognize the lateral ventricles and separate true gray matters from false in the gray matter rim. In addition, more robust segmentation in the amigdala area or automatic recovery techniques will help to find the missed amigdala area. Second, future work will extend segmentation of WM and GM in the cerebrum to

segmentation of other anatomical structures in the entire brain, such as cerebellum and deep subcortical structures.

Currently typical medical resolution of MRI is about $1mm^3$, while research models can have much higher resolutions. An interesting topic for us to study is the performance scalability with respect to higher MRI resolution, particularly of the relative thresholding method. It is also worth verifying whether a fixed pair of relative thresholds can be used for all MR images scanned on different subjects with same image acquisition parameters. Automatic relative threshold selection method for data with unknown imaging parameters is under study.

BIBLIOGRAPHY

- [1] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*. New York: John Wiley and Sons, 1973.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA.: Addison-Wesley, 1992.
- [3] A. Worth, N. Makris, V. Caviness, and D. Kennedy, "Neuroanatomical segmentation in mri: technological objectives," *Int'l J. Patt. Recog. Artificial Intell.*, vol. 11, pp. 1161–1187, 1997.
- [4] A. P. Zijdenbos and B. M. Dawant, "Brain segmentation and white matter lesion detection in mr images," *Critical Reviews in Biomedical Engineering*, vol. 22, no. 5-6, pp. 401–65, 1994.
- [5] R. G. Steen, W. E. Reddick, and R. J. Ogg, "More than meets the eye: significant regional heterogeneity in human cortical t1," *Magn. Reson. Imaging*, vol. 18, pp. 361–386, 2000.
- [6] S. Cho, D. Jones, W. E. Reddick, R. J. Ogg, and R. G. Steen, "Establishing norms of age-related changes in proton t1 of human brain tissue in vivo," *Magn. Reson. Imaging*, vol. 15, pp. 1133–1143, 1997.
- [7] D. N. Kennedy, J. W. Belliveau, J. Rademacher, B. R. Buchbinder, P. A. Filipek, B. R. Rosen, and J. V. S. Caviness, "Anatomic variability of the primary visual cortex," *Society of Magnetic Resonance in Medicine*, 1991.
- [8] J. B. Arnold, J. S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, and D. A. Rottenberg, "Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects," *Neuroimage*, vol. 13, no. 5, 2001.
- [9] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proc. IEEE*, vol. 5, 1979.
- [10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [11] M. Trivedi and J. C. Bezdek, "Low-level segmentation of aerial images with fuzzy clustering," *IEEE Trans. Syst. Man Cybern*, vol. 16, no. 4, 1986.

- [12] J. C. Bezdek, L. O. Hall, and L. P. Clark, "Review of mr image segmentation techniques using pattern recognition," *Med. Phys.*, vol. 20, 1993.
- [13] W. Pratt, *Digital Image Processing*. Hoboken, NJ: Wiley-Interscience, 1991.
- [14] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Survey*, vol. 31, no. 3, 1999.
- [15] Z. Liang, J. R. MacFall, and D. P. Harrington, "Parameter estimation and tissue segmentation from multispectral mr images," *IEEE Transactions on Medical Imaging*, vol. 13, 1994.
- [16] K. S. Cheng, J. S. Lin, and C. W. Mao, "The application of competitive hopfield neural network to medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 15, 1996.
- [17] A. P. Dhawan and L. Arata, "Segmentation of medical images through competitive learning," *Comput. Meth. Prog. Biomed.*, vol. 40, 1993.
- [18] M. Morrison and Y. Attikiouzel, "A probabilistic neural network based image segmentation network for magnetic resonance images," in *Proc. Conf. Neural Networks*, vol. 3, 1992.
- [19] Y. Wang, T. Adah, S. Kung, and Z. Xzabo, "Quantification and segmentation of brain tissues from mr images: A probabilistic neural network approach," *IEEE T. Im. Process.*, vol. 7, 1998.
- [20] J. Mao and A. K. Jain, "A self-organizing network for hyperellipsoidal clustering (hec)," *IEEE Trans. Neural Netw.*, vol. 7, 1996.
- [21] K. Held, E. R. Kops, B. J. Krause, W. M. Wells, R. Kikinis, and et al., "Markov random field segmentation of brain mr images," *IEEE Transactions on Medical Imaging*, vol. 16, no. 6, 1997.
- [22] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE T. Signal Process.*, vol. 40, 1992.
- [23] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport, "Statistical approach to segmentation of single-channel cerebral mr images," *IEEE Transactions on Medical Imaging*, vol. 16, 1997.
- [24] F. Forbes and N. Peyrard, "Hidden markov random field model selection criteria based on mean field-like approximations," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 25, no. 9, 2003.

- [25] J. L. Marrogui, E. A. Santana, and S. Botello, "Hidden markov measure field models for image segmentation," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 25, no. 11, 2003.
- [26] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, 1984.
- [27] J. Besag, "On the statistical analysis of dirty pictures (with discussion)," *J. of Royal Statist. Soc. series B*, vol. 48, no. 3, 1986.
- [28] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation maximization algorithm," *IEEE Trans. on Medical Imaging*, vol. 20, no. 1, pp. 45–57, 2001.
- [29] B. Ruan, B. Moretti, J. Fadili, and D. Bloyet, "Fuzzy markovian segmentation in application of magnetic resonance images," *Computer Vision and Image Understanding*, vol. 85, no. 1, pp. 54–59, 2002.
- [30] M. Ibrahim, N. John, M. R. Kabuka, and A. A. Younis, "Hidden markov models-based 3d mri brain segmentation," *Image and Vision Computing*, vol. 24, no. 10, pp. 1065–1079, 2006.
- [31] Z. Peng, W. Wee, and J. H. Lee, "Automatic segmentation of mr brain images using spatial-varying gaussian mixture and markov random field approach," in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, p. 80, 2006.
- [32] R. D. Morris, X. Descombes, and J. Zerubia, "The ising/potts model is not well suited to segmentation tasks," in *IEEE Digital Signal Processing Workshop*, 1996.
- [33] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "A unifying framework for partial volume segmentation of brain mr images," *IEEE Transactions on Medical Imaging*, vol. 22, no. 1, 2003.
- [34] A. W. Toga, *Brain Warping*. San Diego, CA: Academic Press, 1999.
- [35] J. Talairach and P. Tournoux, *Co-planar Stereotaxic Atlas of the Human Brain. 3-Dimensional Proportional System: An Approach to Cerebral Imaging*. Stuttgart, NY: Thieme Medical Publisher, Inc., 1988.
- [36] C. Davatzikos, "Spatial normalization of 3d images using deformable models," *J. Comp. Assist. Tom.*, vol. 20, 1996.

- [37] P. M. Thompson and A. W. Toga, "A surface-based technique for warping 3-dimensional images of the brain," *IEEE Transactions on Medical Imaging*, vol. 15, no. 4, 1996.
- [38] J. C. Gee, M. Reivich, and R. Bajcsy, "Elastically deforming an atlas to match anatomical brain images," *J. Comp. Assist. Tomogr.*, vol. 17, no. 2, 1993.
- [39] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision, Graphics and Image Processing*, vol. 46, 1989.
- [40] G. E. Christensen, S. C. Joshi, and M. I. Miller, "Volumetric transformation of brain anatomy," *IEEE Transactions on Medical Imaging*, vol. 16, 1997.
- [41] D. L. Collins, T. M. Peters, and A. C. Evans, "An automated 3d non-linear image deformation procedure for determination of gross morphometric variability in human brain," *SPIE*, vol. 2359, 1994.
- [42] J. Ashburner and K. J. Friston, "Unified segmentation," *NeuroImage*, vol. 26, pp. 839–851, 2005.
- [43] B. Fischl, D. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. vander Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A. Dale, "Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain," *Neuron*, vol. 33, no. 3, pp. 341–355, 2002.
- [44] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision, 2nd ed.* Thomson-Engineering, 1998.
- [45] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 2nd ed.* New Jersey: Prentice Hall, 2002.
- [46] D. Ziou and S. Tabbone, "Edge detection techniques: An overview," *International Journal of Pattern Recognition and Image Analysis*, vol. 8, no. 4, pp. 537–559, 1998.
- [47] J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics* (B. S. Lipkin and A. Rosenfeld, eds.), New York: Academic Press, 1970.
- [48] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing" presented at a talk at the Stanford Artificial Project in 1968, unpublished but often cited, orig. in *Pattern Classification and Scene Analysis*, Duda, R. and Hart, P., John Wiley and Sons, '73, pp271-2 *Scene Analysis*, Duda, R. and Hart, P., John Wiley and Sons, '73, pp271-2.
- [49] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. 20, no. 5, pp. 562–569, 1971.

- [50] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 8, Nov 1986.
- [51] E. Hildreth and D. Marr, "Theory of edge detection," in *Proceedings of Royal Society of London*, vol. 207, 1980.
- [52] H. Liu, "Two- and three-dimensional boundary detection," *Computer Graphics and Image Processing*, vol. 6, pp. 123–134, 1977.
- [53] P. Ander and S. Zucker, "Tracing surfaces for surfacing traces," in *Proceedings of IEEE First International Conference on Computer Vision*, pp. 241–249, 1987.
- [54] E. Artzy, G. Frieder, and G. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm," *Computer Graphics and Image Processing*, vol. 15, pp. 1–24, 1981.
- [55] J. D. Cappelletti and A. Rosenfeld, "Three-dimensional boundary following," *Computer Vision, Graphics, and Image Processing*, vol. 48, no. 1, pp. 80–92, 1989.
- [56] D. N. Kennedy, P. A. Filipeck, and V. S. Caviness, "Anatomic segmentation and volumetric calculations in nuclear mr imaging," *IEEE Transactions on Medical Imaging*, vol. 8, no. 1, pp. 1–7, 1989.
- [57] J. L. Lee and J. J. Rodriguez, "Edge-based segmentation of 3-d magnetic resonance images," in *Proceedings of IEEE International Conference on Image Processing*, pp. 876–880, 1994.
- [58] M. Bomans, K.-H. Hohne, U. Tiede, and M. Riemer, "3-d segmentation of mr images of the head for 3-d display," *IEEE Transactions on Medical Imaging*, vol. 9, no. 2, pp. 177–183, 1990.
- [59] H.-H. Ehrlicke, "Problems and approaches for tissue segmentation in 3-d mr imaging," in *Proceedings of SPIE – Volume 1233, Medical Imaging IV: Image Processing*, pp. 128–137, 1990.
- [60] T. Pavlidis and Y.-T. Liow, "Integrating region growing and edge detection," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 12, no. 3, pp. 225–233, 1990.
- [61] J. Xuan, T. Adali, and Y. Wang, "Segmentation of mr brain image: Integrating region growing and edge detection," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 3: 544–547, 1995.
- [62] J. S. Suri, S. Singh, and L. Reden, "Computer vision and pattern recognition techniques for 2-d and 3-d mr cerebral cortical segmentation (part i): A stage-of-the-art review," *Pattern Analysis and Applications*, no. 5, pp. 46–76, 2002.

- [63] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Med. Im. Anal.*, vol. 1, 1996.
- [64] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l J. Comp. Vis.*, vol. 1, no. 4, 1987.
- [65] L. D. Cohen and I. Cohen, "Finite element methods for active contour models and balloons for 2d and 3d images," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 15, no. 11, 1993.
- [66] D. L. Pham, C. Xu, and J. L. Prince, "A survey of current methods in medical image segmentation," tech. rep., The Johns Hopkins University, Baltimore MD 21218, January 1998.
- [67] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 17, no. 2, 1995.
- [68] V. Caselles, F. Catte, F. Coll, and F. Dibos, "A geometric model for active contours," *Numerische Mathematik*, vol. 66, 1993.
- [69] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Comp. Physics*, vol. 79, 1988.
- [70] L. D. Cohen, "On active contour models and balloons," *CVGIP: Imag. Under.*, vol. 53, no. 2, 1991.
- [71] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int'l J. Comp. Vis.*, vol. 22, 1997.
- [72] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE T. Imag. Proc.*, vol. 7, no. 3, 1998.
- [73] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours," *Signal Processing - An International Journal*, vol. 71, no. 2, 1998.
- [74] R. Ronfard, "Region-based strategies for active contour models," *Int'l J. Comp. Vis.*, vol. 13, no. 2, 1994.
- [75] C. S. Poon and M. Braun, "Image segmentation by a deformable contour model incorporating region analysis," *Phys. Med. Biol.*, vol. 42, 1997.
- [76] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, 1989.
- [77] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, Feb 2001.

- [78] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr, "Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional," *International Journal of Computer Vision*, vol. 50, no. 3, 2002.
- [79] T. Kapu, W. E. L. Grimson, R. Kikinis, and W. M. Wells, "Enhanced spatial priors for segmentation of magnetic resonance imagery," in *Proc. 1st Int. Conf. Med. Im. Comput. Comp. Assist. Interven.(MICCAI98)*, 1998.
- [80] D. MacDonald, N. Kabani, D. Avis, and A. C. Evans, "Automated 3-d extraction of inner and outer surfaces of cerebral cortex from mri," *NeuroImage*, vol. 12, no. 3, 2000.
- [81] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan, "Segmentation and measurement of the cortex from 3-d mr images using coupled-surfaces propagation," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, 1999.
- [82] B. H. Brinkmann, A. Manduc, and R. A. Robb, "Optimized homomorphic unsharp masking for mr grayscale inhomogeneity correction," *IEEE Trans. Med. Imag.*, vol. 17, pp. 161–171, 1998.
- [83] M. S. Cohen, R. M. DuBois, and M. M. Zeineh, "Rapid and effective correction of rf inhomogeneity for high field magnetic resonance imaging," *Hum. Brain Mapp.*, vol. 10, pp. 204–211, 2000.
- [84] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in mri data," *IEEE Trans. Med. Imag.*, vol. 17, pp. 87–97, 1998.
- [85] J. B. Arnold, J. S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, and D. A. Rottenberg, "Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects," *NeuroImage*, vol. 13, no. 5, 2001.
- [86] D. L. Pham and J. L. Prince, "An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities," *Patt. Rec. Let.*, 1999.
- [87] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, "A modified fuzzy c-means algorithms for bias field estimation and segmentation of mri data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 3, 2002.
- [88] W. M. Wells, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of mri data," *IEEE Transactions on Medical Imaging*, vol. 15, 1998.
- [89] R. Guillemaud and M. Brady, "Estimating the bias field of mr images," *IEEE Transactions on Medical Imaging*, vol. 16, 1997.

- [90] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation maximization algorithm," *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, 2001.
- [91] K. H. Hohne and W. A. Hanson, "Interactive 3-d segmentation of mri and ct volumes using morphological operations," *J. Comp. Assit. Tom.*, vol. 16, 1992.
- [92] B. D. Ward, "Intracranial segmentation," tech. rep., Medical College of Wisconsin, 1999.
- [93] J.-F. Mangin, O. Coulon, and V. Frouin, "Robust brain segmentation using histogram scale-space analysis and mathematical morphology," in *Proc. 1st MICCAI* (W. M. Wells, A. Colchester, and S. Delp, eds.), (MIT, Boston), pp. 1230–1241, 1998.
- [94] H. K. Hahn and H.-O. Peitgen, "The skull stripping problem in mri solved by a single 3d watershed transform," in *MICCAI*, (Pittsburgh, Pennsylvania, USA), 2000.
- [95] F. Ségonne, A. M. Dale, E. Busa, M. Glessner, U. Salvolini, H. K. Hahn, and B. Fischl, "A hybrid approach to the skull-stripping problem in mri," *NeuroImage*, vol. 22, pp. 1160–1075, 2004.
- [96] S. M. Smith, "Fast robust automated brain extraction," *Human Brain Mapping*, vol. 17, no. 3, pp. 143–155, 2002.
- [97] K. Li, A. D. Malony, and D. M. Tucker, "Automatic brain mr image segmentation by relative thresholding and morphological image analysis," in *Proceedings of the First International Conference on Computer Vision Theory and Applications - VISAPP*, (Setubal, Portugal.), pp. 354–364, February 2006.
- [98] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Comput. Vision Graphics Image Process*, vol. 46, pp. 357–393, 1989.
- [99] G. Bertrand, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recognition Letters*, vol. 15, pp. 1003–1011, 1994.
- [100] F. Segonne, J. Pons, E. Grimson, and B. Fischl, "Active contours under topology control genus preserving level sets," in *Proceedings of CVIBA*, pp. 135–145, 2005.
- [101] G. Malandain, G. Bertrand, and N. Ayache, "Topological segmentation of discrete surfaces," *International Journal of Computer Vision*, vol. 10, no. 2, pp. 183–197, 1993.
- [102] P. K. Saha and B. B. Chaudhuri, "3d digital topology under binary transformation with applications," *Computer Vision and Image Understanding*, vol. 63, pp. 418–429, 1996.

- [103] C. Pudney, "Distance-ordered homotopic thinning: A skeletonization algorithm for 3d digital images," *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 401–413, 1998.
- [104] S. Svensson, "Reversible surface skeletons of 3d objects by iterative thinning of distance transforms," *Digital and Image geometry*, pp. 400–411, 2001.
- [105] G. Borgefors *et al.*, "Computing skeletons in three dimension," *Pattern recognition*, vol. 32, no. 7, pp. 1225–1236, 1999.
- [106] S. Svensson, I. Nystrom, and G. S. di Baja, "Curve skeletonization of surface-like objects in 3d images guided by voxel classification.," *Pattern Recognition Letters*, vol. 23, no. 12, pp. 1419–1426, 2002.
- [107] J.-D. Boissonnat and M. Yvinec, *Algorithmic Geometry*. Cambridge University Press, 1998.
- [108] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.
- [109] K. Li, A. D. Malony, and D. M. Tucker, "A multiscale morphological approach to topology correction of cortical surfaces," in *Proceedings of the Third International Workshop on Medical Imaging and Augmented Reality*, pp. 52–59, 2006.
- [110] B. K. Natarajan, "On generating topologically consistent isosurfaces from uniform samples," *Visual Computer*, vol. 11, pp. 52–62, 1994.
- [111] C. Davatzikos and J. Prince, "An active contour model for mapping the cortex," *IEEE Trans. on Medical Imaging*, vol. 14, no. 1, pp. 112–115, 1995.
- [112] X. Han, C. Xu, and J. L. Prince, "A topology preserving level set method for geometric deformable models," *IEEE Trans. on Pattern Analysis Machine Intelligence*, vol. 25, no. 6, pp. 755–768, 2003.
- [113] I. Guskov and Z. Wood, "Topological noise removal," *Graphics Interface*, 2001.
- [114] Z. Wood, H. Hoppe, M. Desbrun, and P. Schroder, "Removing excess topology," *ACM Trans. Graphics*, vol. 23, pp. 190–208, 2004.
- [115] S. Jaume, *Topology simplification algorithm for the segmentation of medical scans*. PhD thesis, University of Louvain, 2004.
- [116] S. Jaume, P. Rondao, and B. Macq, "Open topology: A toolkit for brain isosurface correction," in *MICCAI Open Source Software Workshop*, 2005.
- [117] D. W. Shattuck and R. M. Leahy, "Automated graph-based analysis and correction of cortical volume topology," *IEEE Trans. on Medical Imaging*, vol. 11, pp. 1167–1177, 2001.

- [118] X. Han, C. Xu, U. Braga-Neto, and J. L. Prince, "Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm," *IEEE Trans. on Medical Imaging*, vol. 21, no. 109-121, 2002.
- [119] F. Segonne, E. Grimson, and B. Fischl, "Topological correction of subcortical segmentation," in *Proc. MICCAI'03*, pp. 675–702, 2003.
- [120] F. Segonne, *Segmentation of medical images under topological constraints*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [121] N. Kriegeskorte and R. Goebel, "An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical mr volumes," *NeuroImage*, vol. 14, pp. 329–346, 2001.
- [122] *Freesurfer*. [Online]. <https://surfer.nmr.mgh.harvard.edu/>.
- [123] *SPM5*. [Online]. <http://www.fil.ion.ucl.ac.uk/spm/>.
- [124] *FSL*. [Online]. <http://www.fmrib.ox.ac.uk/fsl/>.
- [125] Y. Cointepas, J.-F. Mangin, L. Garnero, J.-B. Poline, and H. Benali, "Brainvisa: Software platform for visualization and analysis of multi-modality brain data," *Neuroimage*, vol. 13, no. 6, 2001.
- [126] *BrainVisa*. [Online]. <http://brainvisa.info/>.
- [127] J.-F. Mangin, "Entropy minimization for automatic correction of intensity nonuniformity," in *IEEE Work. MMBIA*, pp. 162–169, 2000.
- [128] J.-F. Mangin, J. Régis, and V. Frouin, "Shape bottlenecks and conservative flow systems," in *IEEE Work. MMBIA*, (San Francisco, CA), pp. 319–328, 1996.
- [129] C. Cocosco, V. Kollokian, R.-S. Kwan, and A. Evans, "Brainweb: Online interface to a 3d mri simulated brain database," *NeuroImage*, vol. 5, no. 4, 1997.
- [130] *BrainWeb*. [Online]. <http://www.bic.mni.mcgill.ca/brainweb/>.
- [131] *The Center for Morphometric Analysis (CMA) at Massachusetts General Hospital*. [Online]. <http://www.cma.mgh.harvard.edu/ibsr/>.
- [132] *The Neurobiology Research Unit*. [Online]. <http://nru.dk>.
- [133] D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, D. A. Rottenberg, and R. Leahy, "Magnetic resonance image tissue classification using a partial volume model," *NeuroImage*, vol. 13, no. 5, 2001.

- [134] K. van Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based tissue classification of mr images of the brain," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, 1999.
- [135] Communication with Freesurfer authors.