The Segmentation of Sparse MR Images

Patrick Craig Marais

A thesis submitted to the Department of Engineering Science, University of Oxford, in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Department of Engineering Science
University of Oxford

Trinity Term
1998

The Segmentation of Sparse MR Images

Patrick C. Marais

A thesis submitted for the degree of Doctor of Philosophy
at the University of Oxford

Robotics Research Group
Department of Engineering Science

Wolfson College
Trinity Term 1998

Abstract

This thesis develops a methodology for the segmentation of anatomical structures within "sparse" MR images. Sparse images were acquired in large numbers prior to the emergence of high-resolution MRI and they form the basis of many long term imaging studies.

The term sparse refers to the fact that the volumetric image has very poor spatial resolution in the direction perpendicular to the slice plane. This leads to a significant degradation in image quality and effectively destroys the spatial continuity of the imaged object. Consequently, generic segmentation schemes — particularly those based on voxel classification — will yield poor results unless they have been augmented in some manner.

Our Segmentation approach is based on a deformable simplex mesh surface, which iteratively interpolates extracted boundary point data. Prior information is mobilised at two levels. Boundary points are found using a matching algorithm based on a database of pre-specified piecewise constant models. These models represent possible idealised intensity profiles for the object boundary. In addition to the boundary model, there is a shape template. The template is generated from a training set of pre-segmented structures, which means that only shapes similar to those in the training set will be recovered. The segmentation proceeds in two phases. The first recovers the normal shape component, determined by the training set, whilst the second deforms smoothly from this constrained solution to produce a more veridical boundary representation.

The segmentation scheme is applied to a number of sparse brain images. Qualitative validation — accomplished by registering the surface extracted from the sparse data to a high resolution scan acquired at the same time-point — indicates that a good approximation to the underlying boundary is obtainable from such images.

Acknowledgements

This work was supported in part by funds from the Foundation for Research Development (FRD) in South Africa, an ORS award (UK), and the European Community Biomorph Project (No. 95-0845). Additional material resources were supplied by the MRC Schizophrenia Research Unit (Radcliffe Infirmary, Oxford), headed by our principal collaborator, Dr Tim Crow. MR data was provided by Prof Lynn DeLisi, of the Dept. of Psychiatry, SUNY. Additional data was obtained from the Biomorph data pool, which was sourced by several different institutes.

The simplex mesh software around which much of this work built was kindly provided by Dr Herve Dellingette of INRIA, Sophia-Antipolis. If this software had not existed, my life would have been considerably more complicated! Within Oxford itself, there are many who have influenced the final form of this work. Dr Jacques Feldmar, now engaged in entrepreneurial pursuits, was a great source of ideas and inspiration. My co-workers have all contributed in some way or another, for which they have my gratitude. In particular, I would like to thank Sebastien Gilles for several useful suggestions with regards to boundary matching. Finally, I extend my sincerest thanks to my supervisors, Mike Brady and Andrew Zisserman. Without their guidance and support, this work would not have been possible.

This thesis is dedicated to those, both past and present, who have inspired me to pursue my dreams. My gratitude to you is eternal.

Contents

1 Introduction **1**

1.1 Background . . . 2

1.1.1 The Structure of the Brain . . . 2

1.1.2 Brain Asymmetry and Schizophrenia . . . 4

1.1.3 Clinical Studies of Brain Asymmetry . . . 5

1.2 A Framework for the Segmentation of Sparse MRI . . . 7

1.3 Thesis Structure . . . 10

2 Preliminaries **11**

2.1 Magnetic Resonance Imaging . . . 11

2.1.1 Nuclear Magnetic Resonance . . . 12

2.1.2 Image Formation . . . 14

2.1.3 MRI Slice Resolution . . . 17

2.1.4 MRI Contrast . . . 17

2.1.5 MRI Artifacts . . . 18

2.2 Characterisation of Sparse MRI Data . . . 20

2.2.1 Sparse MR Images . . . 20

2.2.2 Sparse MR Image Database . . . 20

2.2.3 Segmentation Issues . . . 21

2.3 Overview of Segmentation Methods . . . 23

2.3.1 Voxel-Classification Schemes . . . 23

2.3.2 Constraining Shape . . . 26

2.4 Conclusion . . . 29

3 Snakes **31**
3.1 The Snake Framework . . . 32
3.1.1 Modified Framework . . . 33
3.2 Segmentation . . . 34
3.2.1 Initialisation Issues . . . 34
3.2.2 1D Feature Detection . . . 35
3.2.3 Position Update . . . 36
3.2.4 Correspondence Over Slices . . . 37
3.2.5 Results . . . 38
3.3 Surface Construction and Mid-Saggital Plane Estimation . . . 38
3.3.1 MSP Estimation . . . 38
3.3.2 Surface Construction . . . 41
3.4 Discussion . . . 43
3.4.1 Edge Model Robustness . . . 43
3.4.2 Coping with Topology and Geometry . . . 44
3.5 Conclusion . . . 45

4 Constructing a Shape Template **46**
4.1 The Simplex Mesh . . . 47
4.2 Learning Shape Variation . . . 48
4.2.1 Principal Components Analysis . . . 50
4.2.2 The Point Distribution Model . . . 53
4.3 Generation of the Mesh Training Set . . . 54
4.4 Establishing Point Correspondences . . . 56
4.4.1 Correspondent Extraction . . . 58
4.4.2 Dealing with Correspondence Errors . . . 60
4.5 Analysis of Registration Schemes . . . 62
4.5.1 Type I: Bounding Box Registration . . . 64
4.5.2 Type II: ICP Registration . . . 65
4.6 Results . . . 67
4.6.1 Vertex Redistribution . . . 67
4.6.2 Resampling Errors . . . 68
4.6.3 Projection Error . . . 69
4.6.4 Qualitative Assessment of Correspondences . . . 72

4.7 Discussion 73
4.8 Conclusion 78

5 Boundary Detection **79**

5.1 The Brain Surface Boundary 79
5.2 The Boundary Model Framework 80
5.2.1 Model Input Data 81
5.2.2 The Models 82
5.3 Edge Heuristics 82
5.4 Node Modelling 85
5.5 Specifying a model 87
5.5.1 Model Construction and Representation 89
5.5.2 Matching the Model to Data 89
5.5.3 Model Decomposition 91
5.5.4 Coarse Matching 93
5.5.5 Local Matching 94
5.5.6 Model Constraints 98
5.5.7 Matching for Multiple Boundary Classes 99
5.5.8 Discretisation Issues 100
5.6 Dealing with Cortical Folding 102
5.7 Boundary Detection Results 103
5.7.1 Model Setup and Parameters 103
5.7.2 Results 104
5.8 Discussion 104
5.9 Conclusion 109

6 Modelling the Partial Volume Effect **110**

6.1 Related Work 110
6.2 The MR Imaging model 111
6.3 The Prediction Scheme 111
6.4 The Boundary Detection Strategy 114
6.5 Experiments on Synthetic Data 115
6.5.1 Test Data 116
6.5.2 Quantifying Edge Estimation Errors 116

6.5.3 Prediction Using Matched Models . . . 123
6.6 Discussion . . . 126
6.7 Conclusion . . . 126

7 Surface-based Segmentation **127**
7.1 Utilising the Shape Constraint . . . 127
7.2 The Segmentation Framework . . . 128
7.3 Template Initialisation . . . 130
7.3.1 Computing the Template Bounding Box . . . 131
7.3.2 Computing the Shape Instance Bounding Box . . . 131
7.3.3 Initialisation Examples . . . 133
7.4 Computing the Surface Update . . . 136
7.4.1 The Implications of Truncation and Sparsity . . . 136
7.4.2 Computing the Closest Point Deformation . . . 137
7.4.3 Stage I: ASM Segmentation . . . 141
7.4.4 Stage II: Simplex Segmentation . . . 143
7.5 Results . . . 145
7.5.1 The Template . . . 145
7.5.2 Test Database . . . 146
7.5.3 Parameters . . . 146
7.5.4 Sparse Segmentation . . . 147
7.6 Discussion . . . 155
7.7 Conclusion . . . 158

8 Conclusion **159**
8.1 Future Work . . . 160
8.1.1 The Shape Template . . . 160
8.1.2 The Boundary Model . . . 160
8.1.3 PVE Prediction . . . 161
8.1.4 Segmenting Other Structures . . . 161
8.1.5 Asymmetry/Shape Analysis . . . 162

APPENDICES **164**

A B-Spline snakes 164

A.1 Definition . . . 164

A.2 Area measurements . . . 166

B The Simplex Mesh Formalism 168

B.1 Geometric Properties . . . 170

B.2 Metric Parameters . . . 171

B.3 The Simplex Mesh and Segmentation . . . 172

B.3.1 Internal Forces . . . 173

B.3.2 External Forces . . . 174

C Mesh Measures 176

C.1 Surface Area . . . 176

C.2 Volume . . . 176

C.3 Transforming the PCA Eigensystem . . . 177

D Mahalanobis PCA criterion 179

E Cosine Foreshortening for Curved Surfaces 181

E.1 An Example . . . 184

F T2 Boundary Model Database 186

Bibliography 186

List of Figures

1.1 Two Views of the Brain . . . 3
1.2 The Ventricular System . . . 4
1.3 Laterality Measures . . . 5
1.4 The Laterality Index . . . 6
1.5 Simplex Mesh (ventricular) Model . . . 8
1.6 Model Constraints . . . 9
1.7 In-plane boundary detection . . . 9

2.1 Effect of a Magnetic Field . . . 13
2.2 RF Resonance and Relaxation . . . 13
2.3 K-Space . . . 15
2.4 An MRI Image . . . 17
2.5 Bias Field . . . 19
2.6 Imaging Artifacts . . . 19
2.7 Partial Volume Effects . . . 21
2.8 Data Truncation . . . 22
2.9 Voxel Segmentation . . . 24

3.1 Snake Feature Search . . . 35
3.2 Snake Segmentation . . . 39
3.3 Predicted Quantities . . . 39
3.4 Motion Prediction . . . 40
3.5 Mid-Saggital Line Cross-Sections . . . 40
3.6 Mid-Saggital Estimation . . . 41
3.7 Steps in Surface Construction . . . 42

4.1 Principal Components . . . 51
4.2 Snake Segmentation . . . 55

4.3 Mesh Surface Representation . . . 57
4.4 Correspondent Selection . . . 59
4.5 Correspondent Inversions . . . 60
4.6 Example of Vertex Redistribution for a Curve . . . 61
4.7 Comparison of PCA Methods . . . 65
4.8 Normal Selection Vertex Redistribution . . . 68
4.9 Closest Point Vertex Redistribution . . . 69
4.10 Global Registration . . . 70
4.11 PDM Deformation Modes . . . 71
4.12 Projection Errors . . . 72
4.13 Randomly Selected Correspondents (Bottom View) . . . 74
4.14 Selected Correspondents (Side View) . . . 75
4.15 Correspondent Point Clustering . . . 76
4.16 Correspondent Selection . . . 76
4.17 Spline Deformation . . . 77

5.1 Cerebral Boundary Structure . . . 80
5.2 Cortical Folding . . . 81
5.3 Edge Heuristics . . . 84
5.4 Matching Under the Mahalanobis Criterion . . . 86
5.5 Profile PCA . . . 87
5.6 Poor Boundary Model . . . 88
5.7 Matching with a Dynamic Boundary Model . . . 88
5.8 Simple Translation . . . 90
5.9 Model Elements . . . 91
5.10 Coarse to Fine Matching . . . 92
5.11 Model Decomposition . . . 92
5.12 Coarse Matching . . . 95
5.13 Local Matching . . . 96
5.14 Vertical Fitting . . . 97
5.15 Order of Local Matching . . . 97
5.16 Model Refinement . . . 100
5.17 Validation Profiles . . . 102
5.18 Edge Detection - Example One . . . 105

5.19 Edge Detection - Example Two 106
5.20 Examples of Matched Models 107
5.21 The Need for Coarse Models 108

6.1 Voxel Partial Volume 112
6.2 Prediction of Local Intensities 113
6.3 Profile Prediction and Matching 115
6.4 Synthetic Volumetric Data 116
6.5 Partial Volumes and Edge Location 117
6.6 Matching in the Presence of Noise 118
6.7 Estimates with Model Noise (2-3mm sphere) 119
6.8 Estimates with Model Noise (4–6mm sphere) 120
6.9 Image and Model Noise (4-6mm sphere) 121
6.10 Prediction in Real Data Using a Single Boundary Model 122
6.11 Prediction Using A Good Model 124
6.12 Prediction Using Coarse or Incomplete Models 125

7.1 2D Active Shape Model 128
7.2 Template Initialisation 134
7.3 Computing the Initialisation 135
7.4 Robust Plane Estimation 136
7.5 Poor Boundary Discrimination 137
7.6 Affine (ICP) Segmentation 138
7.7 Closest Point Deformation 139
7.8 Deformation Cusp 140
7.9 ASM Constrained Segmentation 142
7.10 External Force Decay 144
7.11 Simplex Template Segmentation 147
7.12 Cerebral Contours — Example 1 148
7.13 Full Cerebral Surface 149
7.14 Mesh Contours in Sparse Data — Example 2 150
7.15 Mesh Contours in Dense Data — Example 2 151
7.16 Mesh Contours in Sparse Data — Example 3 152
7.17 Mesh Contours in Dense Data — Example 3 153

7.18 Mesh Contours in Sparse Data — Example 4 154
7.19 Sulcal Mismatch 156
7.20 Merged Boundaries 157

8.1 Symmetry Map 162

A.1 Cubic B-spline 165

B.1 Simplex Mesh Triangulation 169
B.2 Simplex Angle 170
B.3 Metric Parameters 171
B.4 Stiffness Function 174

E.1 Cosine Foreshortening 181
E.2 Cosine Foreshortening on the Sphere 183
E.3 Foreshortening Example 185

F.1 Model Database 187

List of Tables

2.1 MRI Contrast-Weighting Schemes 18

F.1 Model Parameters 186

Chapter 1

Introduction

The introduction of *magnetic resonance imaging* (MRI) marked a significant milestone in the development of diagnostic radiology. The excellent soft tissue discrimination afforded by MRI allows for the detection of subtle pathologies which are often invisible or poorly resolved under other imaging modalities. It is also non-invasive[1] and has no demonstrable side effects. These characteristics have ensured that MRI has rapidly emerged as the modality of choice for the investigation of a range of neurological disorders.

An analysis of brain structure necessitates some sort of *segmentation* of the data. For our purposes, segmentation refers to the extraction and description of (specified) anatomical structures within the imaged volume. Unfortunately the tremendous variability of anatomical structures, coupled with poor resolution and inescapable imaging artifacts, renders this task extremely difficult in all but the simplest of cases. These limitations are of particular importance to us, since our MR data is *sparse*. In this context, the term sparse refers to the fact that the spatial resolution of the volumetric image is much coarser along one axis than the other two. A detailed description of the image sequences we employed will be given in Section 2.2. Although current MR technology allows for the acquisition of high resolution image sequences, we cannot simply disregard existing sparse data sequences. Such data sets constitute an important part of valuable long term studies, spanning the last 6–10 years, and reliable techniques must be developed to assist radiologists in processing them. The investigation of *brain asymmetry* in schizophrenics is one such area, and served as the primary motivation for this work. Although this thesis does not address the issue of brain asymmetry directly (though a clinical study), the issues involved in brain segmentation are well illustrated by work in this area, and we shall therefore refer extensively to the relevant literature.

The objective of this work is the development of robust algorithms to segment anatomical structures from sparse MRI sequences. We seek a methodology which will allow us to in-

[1] If no contrast agents are used.

corporate our knowledge of the domain: the successful exploitation of such information may well mean the difference between success or failure.

1.1 Background

The brain is a complex structure with very specific segmentation requirements. The techniques developed here will be tested on sparse brain MR images, since the segmentation of such data provides a challenging application and is a precursor to any subsequent analysis of brain asymmetry. To help appreciate the issues involved, we provide a brief overview of brain anatomy and review a number of segmentation techniques used by other researchers in this area.

1.1.1 The Structure of the Brain

The Brain is composed of *white* and *grey (cortical) matter* : the latter consists of neurons, the fundamental processing units of the brain, whilst the former provides the connections which link these cells together. Grey matter envelopes white and is arranged so as to maximise its surface area in the cramped confines of the skull. The resulting cortical folding, which is characteristic of the largest (uppermost) part of the brain, the *cerebrum*, gives rise to *sulci* and *gyri.* Sulci are winding fissures which work their way across the cerebral surface. These sulci, in turn, delineate the *gyri* — ridges of cortical matter which give the cerebrum its characteristic lumpy look — Figure 1.1. The *cerebellum*, which is attached to the underside of the cerebrum, is a less sophisticated structure, being both smaller and possessing a less convoluted surface geometry. Its primary function is the regulation and maintenance of autonomous processes — those which are considered "instinctive".

The cerebrum is decomposed into a series of *lobes*: frontal, temporal, parietal and occipital — Figure 1.1. Their delineation is somewhat imprecise, since the boundaries are assumed to lie along certain major sulci and these are frequently displaced across patients. The temporal lobe is bounded by the *sylvian fissure*, a prominent sulcus which is often used to define anatomically stable landmarks.

There is a natural bilaterality to the brain, which is most clearly demonstrated by the bifurcation of the cerebrum into the left and right cerebral hemispheres. The inter-hemispheric fissure, or *mid-saggital plane (MSP)*, provides an approximate plane of symmetry, but one which is only defined over part of the brain: the hemispheres fuse together in an interconnect known as the *corpus callosum.* The cerebellum is similarly divided. The interior of the brain contains additional structures. The largest of these is the *ventricular system* (Figure 1.2), a network of interconnected chambers which interpenetrates each hemisphere. This structure

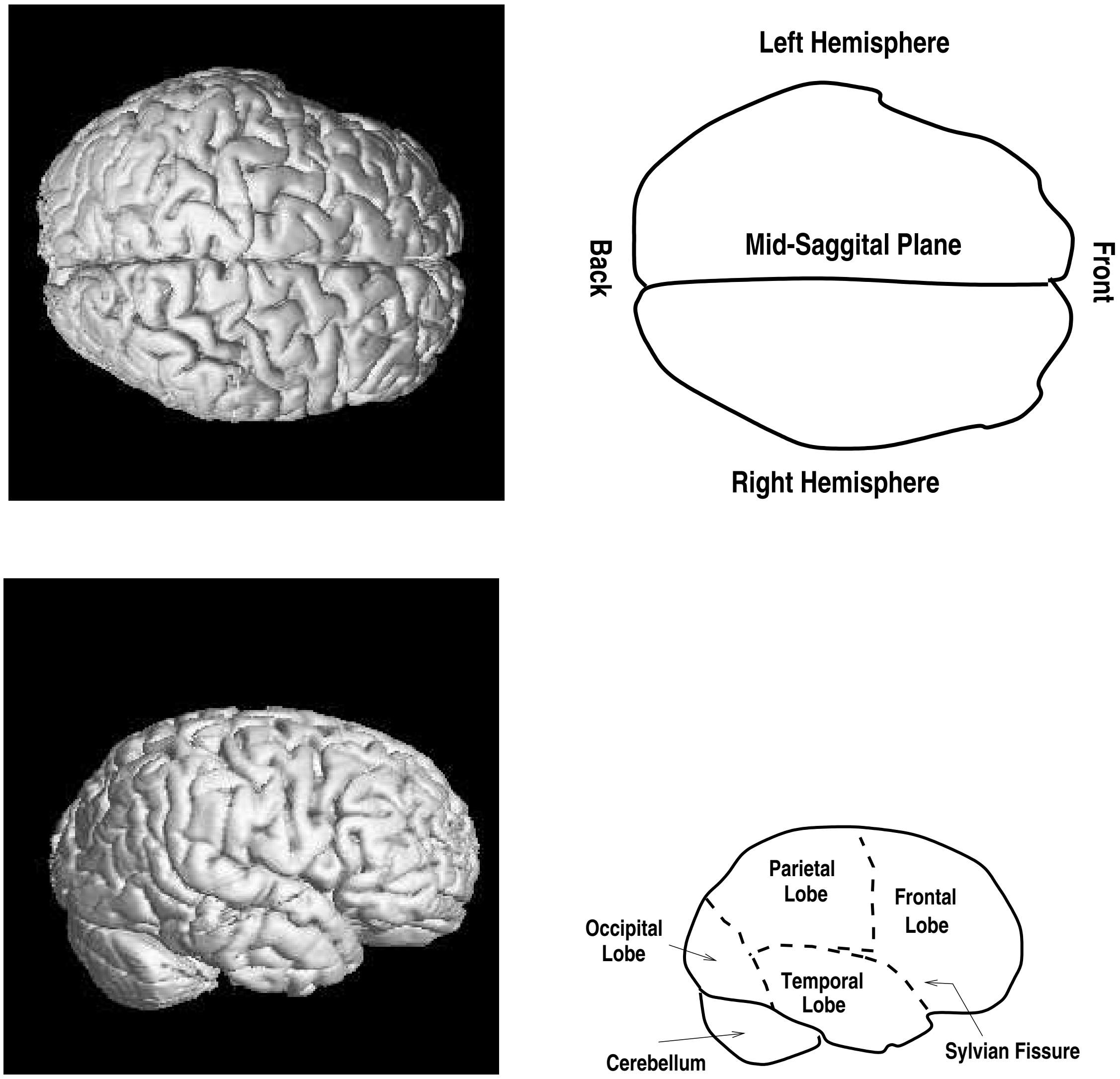
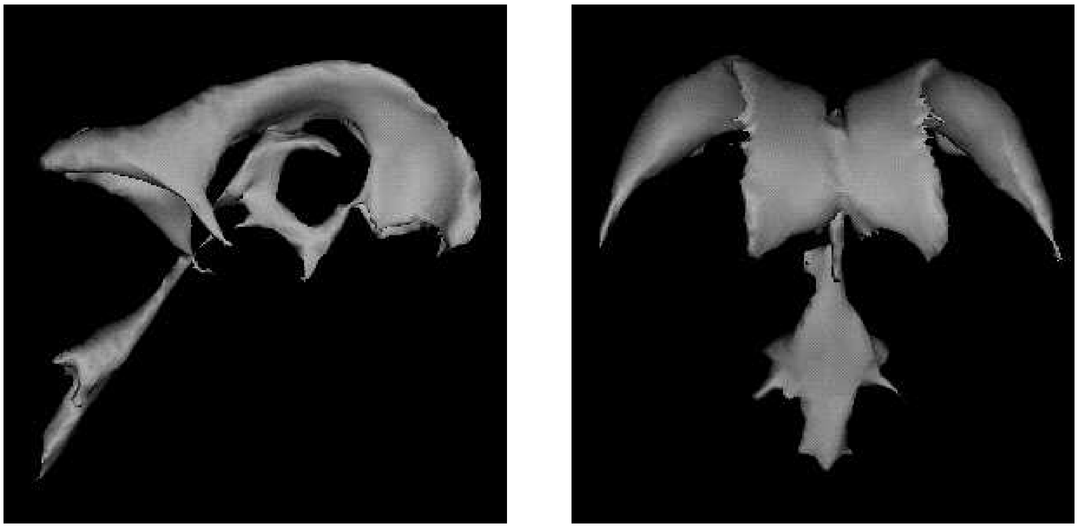


Figure 1.1: Two Views of the Brain The first row shows a top-down axial view, in which the two cerebral hemispheres are clearly delineated by the mid-saggital plane. The second row shows a saggital view — note that the cerebellum is included. The asymmetry between the left and right hemispheres is readily apparent. The principal brain lobes are also indicated (The volume rendered images were supplied by the Biomorph Project).



A B

Figure 1.2: The Ventricular System The ventricular system is a group of connected chambers which interpenetrates each hemisphere. A) lateral view — left and right correspond to the back and front of the brain, resp. B) front view — left and right correspond to the right and left sides, resp., of the brain.

generates the cerebro-spinal fluid (CSF) which nourishes and protects the brain.

One may associate parts of the cortex with specific brain function — this is the purpose of functional neuro-imaging. However, since the aim of our work is to quantify structure rather than function, we shall omit further discussion of this topic.

It is worth pointing out that a standardised nomenclature exists for specifying views of brain anatomy. The three canonical views are along the axis of the body (*axial*), from side to side (*saggital*) and from front to back or vice versa (*coronal*). These terms are frequently used in the context of brain MRI and will appear regularly in subsequent chapters.

1.1.2 Brain Asymmetry and Schizophrenia

Recent research has suggested that schizophrenia is accompanied by structural changes within the brain. In particular, there is evidence of abnormal cerebral symmetry (about the central fissure). The brain usually possesses a natural *asymmetry* i.e. the left and right hemispheres are *not* mirror images of each other. It is thought that the developmental processes responsible for evolving brain asymmetry are interrupted in adolescence, leading to a more symmetric distribution of cerebral structure [17, 27]. Unfortunately, these studies have also shown that this effect can be quite subtle and may consequently elude detection if the spatial resolution of the scans is poor.

The MSP forms the standard reference frame for cerebral *laterality measures* (such as the relative widths of the brain perpendicular to a given point on the plane). When one refers to the symmetry of the brain, one is implicitly assuming a Left-Right (LR) measure of this type — Figure 1.3. Observe that the image presented here has had left and right interchanged: the view point is located beneath the brain. This reflection is a natural consequence of the acquisition protocol and should be borne in mind when examining MRI images. Although

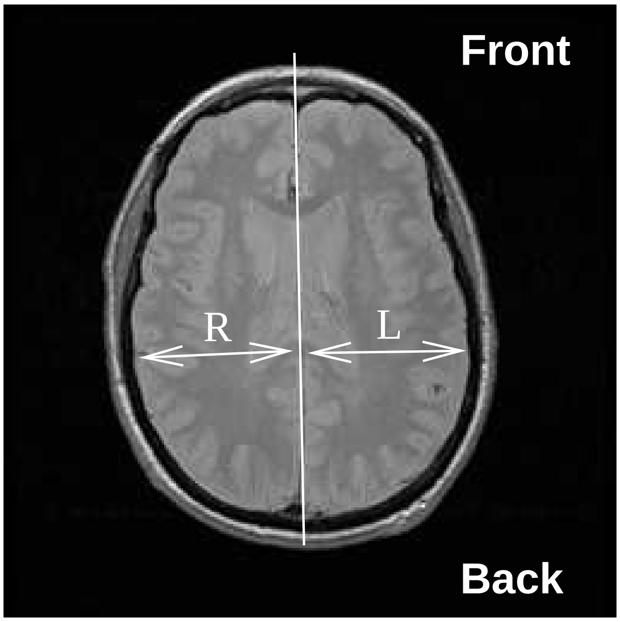


Figure 1.3: Laterality Measures The mid-saggital line is overlaid on this transverse T1-weighted MR image. Observe that left and right are interchanged — this indicates that the scan was acquired from the bottom to the top of the brain. The laterality measure $\frac{L-R}{L+R}$ is often used to quantify asymmetry w.r.t. this line.

a cursory investigation might suggest otherwise, the cerebrum actually possesses a distinct *asymmetry* with respect to the MSP: the brain of a healthy subject displays a significantly larger left hemisphere than right in the posterior part of the brain. As one moves in an anterior direction, this relationship decreases and then undergoes a reversal — see Figure 1.1. This effect is sometimes called *torsion*, since it is believed that there may be a developmental "torque" which produces this skewed tissue distribution.

1.1.3 Clinical Studies of Brain Asymmetry

A number of reports have suggested that this natural asymmetry is significantly reduced in the brains of schizophrenics. Evidence of this phenomenon may be found at several levels within the brain, from gross lateral changes in the temporal lobe to subtler deformations in structures such as the hippocampus.

The techniques used to investigate this process are many and varied. For example, in [17],

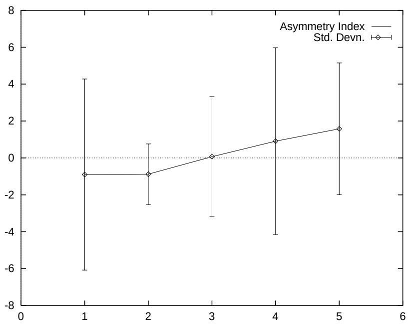


Figure 1.4: The Laterality Index as a means of representing LR asymmetry in the brain. The y-axis measures the laterality index as one moves from the anterior to the posterior of the patient's brain (the numerical values on the x-axis provide ordering information only). Note the great size of the error associated with the results.

the "Radius of Gyration" — essentially a radial variance measure about a point set mean — is used to derive a measure of asymmetry within coronal MRI scans. This metric is applied to the estimated grey/white-matter boundary curves within each slice and the resulting values are used to construct laterality indices. The indices for each slice are then averaged to produce a measure for the entire brain. The authors report a definite lateralised change in asymmetry which seems to be present only in males. However, the suitability of the Radius of Gyration as a metric is questionable. The grey-white matter interface is a complex boundary which may give rise to fragmented curves within individual slices. It is difficult to see how such a simple slice-based measure could encompass this fragmentation or provide a meaningful description of interface complexity.

Rather than adopting such a convoluted scheme, one may choose to directly evaluate asymmetry within each slice [31]. Although conceptually simple, this approach provides the most intuitive interpretation of lateral asymmetry. Unfortunately, slice-based methodologies are particularly vulnerable to patient-machine alignment artifacts (in-plane foreshortening effects) which can introduce spurious results — Appendix E examines this issue in more detail.

In contrast to the above methods, the work of Bilder et.al. [6] is based on the statistical analysis of regional volumetric differences. The regions are delimited manually, using established neuro-anatomical landmarks. Hemispheric sub-volumes are then used to construct a laterality index as a function of axial displacement — Figure 1.4. The results indicate little difference in total or regional hemispheric volumes between patients and controls, but reveal the existence of abnormal lateral hemispheric asymmetries. It can be seen from Figure 1.4 that the error associated with laterality indices (corrected for patient head tilt) is very large, limiting the usefulness of the data. This is probably due to errors in the definition of sub-regions — it is often difficult to identify corresponding regions across patients, particularly when low resolution MRI data is used.

Rather than examining laterality measures, one can opt for a less specific analysis. In [75] the authors segment the brain into several tissue classes using a semi-automatic thresholding procedure. They compute global grey and white matter volumes from this segmentation as well as specific cortical volumes for selected sub-regions. Since there is a well known correlation between grey-matter reduction and age, the data is normalised to account for this effect. Their findings suggest that there are strongly localised grey-matter abnormalities in schizophrenics. In particular, there was a decrease in grey matter in almost all the cortical regions amongst schizophrenic patients. Once again, however, these results must be interpreted with some caution. The data used for these measurements is truncated and sparse which means that corresponding landmarks cannot always be reliably established. An alternative study of localised grey-matter reduction, Shenton et al. [59] examined the volume of grey-matter in the temporal lobe and concluded that this volume was reduced in schizophrenics. However, they also show that the frontal lobe does not exhibit such a reduction, which contradicts the findings of the previous authors.

Although the studies referred to above have all dealt with manifestations of asymmetry, alternative indices of pathology have been investigated.

A study undertaken by Bullmore [16] attempts to quantify the fractal dimension of the grey-white matter interface, with a view to comparing this value for normals and patients. The analysis is slice-based and the fractal dimension is only able to quantify mean structure, which is of limited usefulness. An entirely different approach is taken by Kikinis [45]: the sulcal patterning of the cerebrum is analysed for pathological indicators. The study used 3D volumetric image processing techniques to produce standardised views of the sulcal patterning which were then subjected to qualitative and quantitative analysis. These analyses revealed a distinctive alignment to the sulci of schizophrenic brains, suggesting a neuro-developmental aspect to the disease.

While evidence of anomalous asymmetry can be found in other structures (such as the ventricular system [75]), the discrepancy between the left and right cerebral hemispheres seems to be most frequently cited [6, 31, 75] as representative of this phenomenon.

1.2 A Framework for the Segmentation of Sparse MRI

The main contribution of this thesis is the development of a framework for the segmentation of anatomical structures from sparse MRI. The volumetric images used in this work are composed of a sequence of 2D images, spaced at 2mm intervals, each of which represents a slab of material 5mm thick. The resulting lack of structural coherence across neighbouring slices, suggests that attempts to extract the boundaries of interest using 3D voxel segmentation methods will be unsuccessful. Such schemes operate at the voxel scale, which is unacceptably

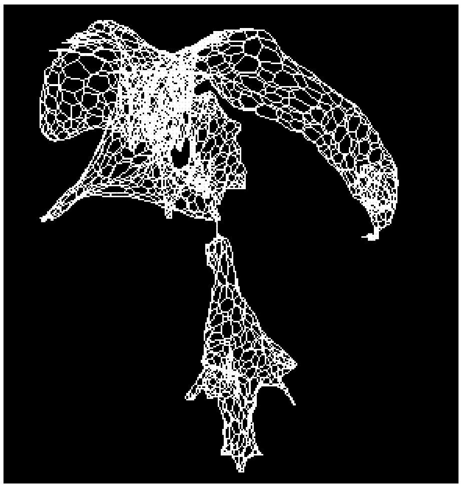


Figure 1.5: Simplex Mesh (ventricular) Model Each vertex in this model is connected to 3 others. This property is useful when constructing algorithms to process surface structure.

coarse in our case.

Given the constraints imposed by the sparsity of our data, we have opted for a model-based segmentation, in which our model encodes the 3D structural variation present within the brains of a normal population (the *training set*). This shape model or template has an associated *boundary model*, which is the active component responsible for the detection of likely object boundaries amidst background clutter. The template deforms towards the detected boundaries, but its deformation is constrained by the shape variability present with the training set.

The template has the following features, which are justified in Chapter 4:

Simplex Mesh The template is represented as a 3D simplex mesh, in which each vertex is connected to three others. Figure 1.5 shows a simplex mesh model of the ventricular system. The simplex mesh possesses a number of useful properties which will assist us in constructing the template.

Discrete PCA The shape constraints are represented as a mean shape with associated modes of variation — Figure 1.6. These are computed from the training set using a Principal Components Analysis on the discrete mesh structure.

The boundary detection scheme contains the following elements:

Line-Based Searches The boundary detection process is based upon the analysis of 1D intensity profiles extracted along 3D lines within the volumetric image.

Plane Constraint For the case of sparse MRI scans, we constrain these search lines to lie within the image plane, since the MRI data is not defined between slices and we wish to avoid the use of spurious interpolated intensity data. Figure 1.7 shows an image with

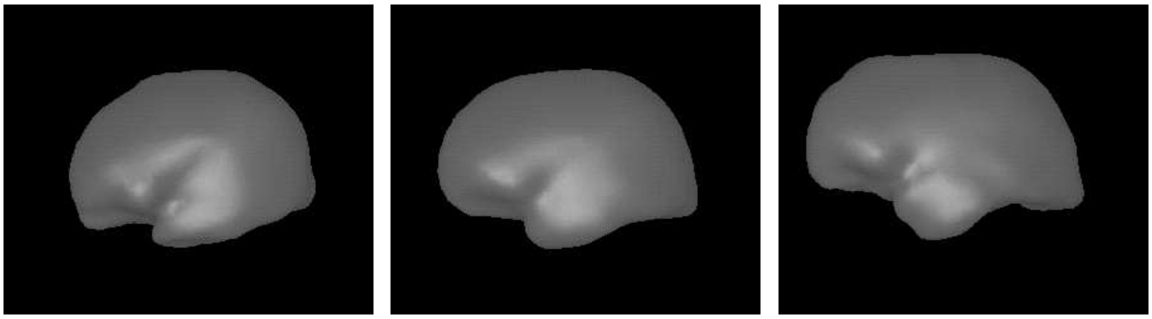


Figure 1.6: Model Constraints The model is defined by a mean shape with a specified number of "modes of variation". These modes allow the model to represent the variability present within the training set. In this figure, the centre image is the mean shape. The left and right images were generated by adding positive and negative multiples (resp.) of the first mode to this shape.

a series of 1D in-plane intensity profiles.

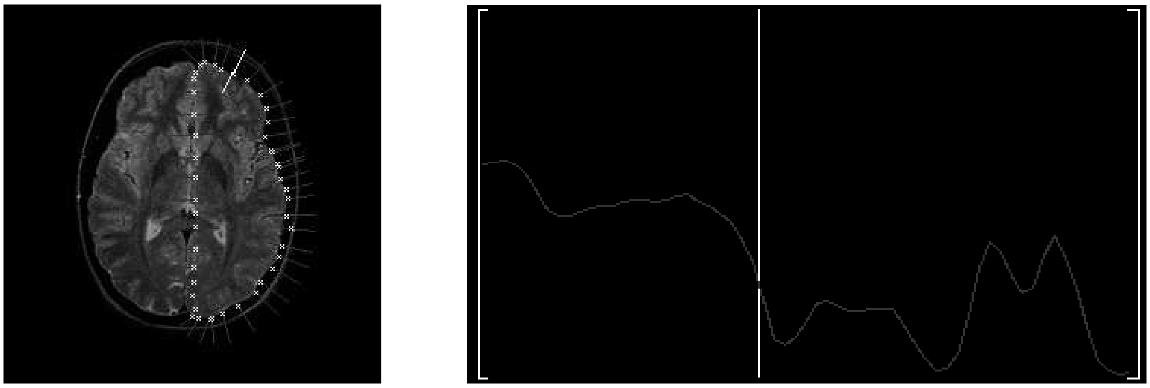


Figure 1.7: In-plane boundary detection The left image shows the edges detected after a few iterations of the segmentation algorithm — the lines represent the data which is examined while the small white crosses are the edge estimates. The right image shows the profile associated with the high-lighted search line on the left. It is worth pointing out that the boundary we seek need not have maximal response to a (step) edge detector — hence the need for a model.

Model of PVE The Partial Volume Effect limits the accuracy of boundary localisation and is particularly severe in sparse images. We have developed a scheme to predict the true voxel intensity in the surface neighbourhood, and coupled this to the boundary detection framework.

The constrained segmentation will recover a surface which is similar to those included in the training set. In other words, atypical structure (which we consider to be pathology) will not be segmented. Since the ultimate goal of this work is the quantification of such anomalous structure, we have implemented a *two-tier segmentation* scheme. The first level of this segmentation is constrained by our template. The second level is initialised with the result of the first stage and the segmentation procedure is repeated. However, in this step, the shape constraints are less restrictive and a more accurate surface can be recovered.

1.3 Thesis Structure

The thesis is presented as follows:

Chapter Two provides an overview of the MR image acquisition process and details the precise nature of the data we wish to process. Next, a review of existing segmentation techniques is undertaken, accompanied by a discussion of their respective shortcomings with respect to our particular requirements.

Chapter Three introduces snake-based segmentation in the context of sparse data. The standard 2D snake framework is extended to include *a priori* information and a segmentation based on propagating planar snakes is developed. The construction of 3D surfaces from this 2D data and the extraction of the Mid-Saggital Plane — required for quantification of asymmetry — are then investigated. The chapter concludes with a discussion of the limitations imposed by sparsity and outlines the criteria which an alternative 3D scheme should satisfy.

Chapter Four introduces the Simplex Mesh formalism. We discuss the deficiencies of the basic mesh-segmentation approach, and show how it can be extended to suit our purposes. The issues involved in the construction of a mesh shape template are then addressed.

Chapter Five develops the boundary detection scheme which will underpin the segmentation.

Chapter Six proposes a scheme to predict the intensity of voxels which are badly affected by PVE. This framework follows naturally from the boundary modeling scheme developed in the previous chapter, and provides the basis for more accurate boundary localisation.

Chapter Seven combines the work of previous chapters to arrive at the full segmentation strategy. The two-tier segmentation scheme is evaluated over a number of examples, and a detailed discussion of the results is presented.

Chapter Eight concludes the thesis. The methodology is reviewed and its shortcomings are discussed, along with proposals for improvement.

Chapter 2

Preliminaries

The construction of a robust segmentation scheme requires a thorough understanding of one's data. This observation is particularly relevant to magnetic resonance imaging, in which the *protocol* determines the image resolution and contrast, and consequently the image features which may be reliably extracted. This chapter provides the background required to understand the issues associated with the segmentation of sparse MRI.

We begin by introducing the basic concepts behind MR imaging — Section 2.1. In Section 2.2, we provide a more rigorous definition of "sparsity" and discuss the problems such data poses for segmentation. A brief review of relevant segmentation schemes is then presented in Section 2.3, and the chapter concluded in section 2.4.

2.1 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a non-invasive volumetric imaging technology. Although there are other volumetric imaging modalities, for example positron emission tomography (PET) and computed tomography (CT), MRI (usually) does not require the introduction of radio-isotopes into the patient's blood stream, nor does it utilise high-energy electro-magnetic radiation. MRI is characterised by exceptional discrimination of soft tissues, which is of great importance for the investigation of tumours and other tissue-based pathologies.

The technique is based on the phenomenon of *nuclear magnetic resonance* (NMR), which has been utilised in chemical analysis since the latter part of the 1940s. The imaging application was first proposed by Paul Lauterbur, then of the State University of New York, in a 1973 Nature article. It was only a decade later, however, that the first MRI machines were realised. Whilst these machines were primitive by current standards, they revolutionised medical diagnosis, providing detailed views of the body's innermost workings.

Of course, MRI did not emerge as a fully evolved technology. There are many issues, particularly those pertaining to imaging artifacts, which are only now being addressed. Since the bulk of our data was acquired some years ago, these problems need to be addressed before an effective analysis can be developed.

2.1.1 Nuclear Magnetic Resonance

The nuclei of elements with odd atomic numbers possess an intrinsic non-zero "spin". This property is the quantum mechanical equivalent of classical angular momentum, which macroscopic rotating bodies possesses. Quantum mechanics predicts that each nucleus with non-zero spin will possess a number of quantised spin states, the precise number depending on the characteristics of the nucleus in question. In the case of Hydrogen, there are two such states. For simplicity, we shall limit our discussion to this element.

Because nuclei have a net positive charge, each has an associated magnetic field and, consequently, a prescribed *magnetic moment*, that is, each nucleus behaves as if it were a tiny magnet rotating about some axis. In an environment with no coherent external magnetic fields, the orientations of the magnetic moments will be random and the net magnetic field arising from these magnetic moments will be approximately zero. The introduction of an external magnetic field $\mathbf{B}$ has two important consequences:

1. It increases the separation between the low and high energy states of the nuclei which, according to quantum mechanics, also increases the relative sizes of the populations inhabiting each of these states — as the field strength grows, the lower energy state becomes more densely populated.

2. It induces *precession* of the magnetic moments of each nucleus about the applied field, Figure 2.1. The precession is at a frequency, ω, specified by the *Larmor Equation*: $\omega = \gamma\ |B|$, where γ is the gyro-magnetic ratio, which differs for each element. In the case of Hydrogen, for example, it has the value 42.58 MHz/Tesla (the Tesla is a measure of magnetic field strength).

The excess of nuclei with aligned magnetic moments in the low energy state manifests itself as a macroscopic magnetisation vector, $\mathbf{B}_0$, oriented along the external field (Figure 2.1). By further stimulating the sample with a radio-frequency (RF) pulse containing the same energy as that separating the two spin states, we induce a *resonance* phenomenon which boosts a proportion of the nuclei into the unstable high energy state, where they remain until the stimulus is removed. To an external observer, the RF pulse, which is at or near the Larmor (resonance) frequency, flips the magnetisation vector into the orthogonal (transverse) plane (for a 90 degree pulse). Within the transverse plane the magnetisation vector is subject to the

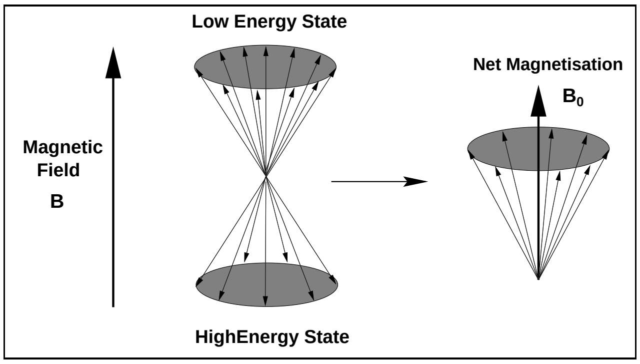


Figure 2.1: Effect of a Magnetic Field An external magnetic field causes the individual nuclear magnetic moments to precess about it. Because of the discrepancy between the two spin states, a net magnetisation is produced on the macroscopic level, oriented along the field.

external field, and proceeds to rotate at the Larmor frequency. Since a changing magnetic field induces a current in a length of wire, the strength of rotating magnetisation vector may be measured by placing a *receiver coil* in close proximity to the sample. These various elements are shown in Figure 2.2.

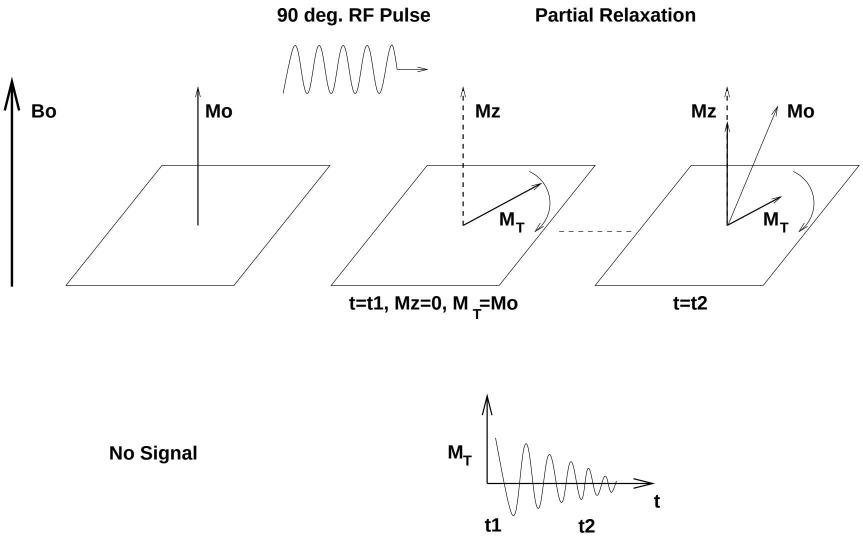


Figure 2.2: RF Resonance and Relaxation A 90^o RF pulse at the resonance frequency flips the magnetisation vector into the transverse plane, where it rotates at the Larmor frequency. The transverse component, M_T, of the magnetisation vector M_0 — which is what our receiver coil detects — is maximal at this point, while the axial component M_Z is usually zero. Over time the nuclei revert to the lower energy state and M_T decays as the magnetisation spirals back to equilibrium. Because of this "relaxation" a decaying alternating current is detected by the receiver coil. This signal is known as the Free Induction Decay (FID).

Over time, nuclei revert to the low energy state, causing a decay in the magnitude of the transverse component of the magnetisation. If one looks at the signal detected in the coil, it appears as a decaying sinusoid. This is known as the *free induction decay (FID)*.

This relaxation is exponential and is characterized by two parameters $T1$ and $T2$. The first governs the growth of longitudinal (axial) magnetisation, M_Z, while the second characterises the decay of the transverse component, M_T. More formally,

$$M_Z(t) = M_0 + (M_Z(0) - M_0)e^{-\frac{t}{T_1}} \tag{2.1}$$

$$M_T(t) = M_T(0)e^{-\frac{t}{T_2}} \tag{2.2}$$

where M_0 is the equilibrium value of magnetisation vector (prior to the pulse) and $M_T(0)$ and $M_Z(0)$ are the initial values after a 90 degree pulse. Observe that $M_Z(0)$ will not be zero if the sample has not been permitted to relax completely prior to the next RF pulse.

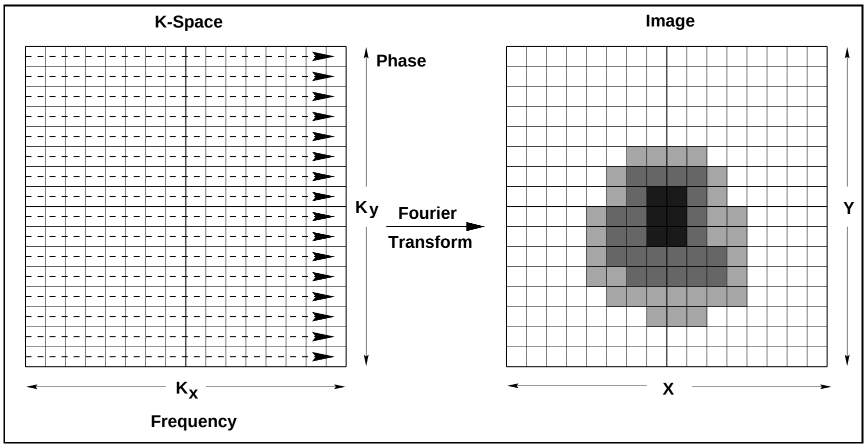
The precise values of $T1$ and $T2$ are affected by, amongst other things, the composition and temperature of the sample, as well as the strength of the external field. These values determine the contrast in an image. Furthermore, one may accentuate one or the other by manipulation of the MR imaging parameters, resulting in so-called $T1$ and $T2$ "weighted" images. Contrast and imaging are discussed in the next section.

In practise the RF pulse will have a finite bandwidth, which means that a range of nuclei will be stimulated. This produces a composite signal which is a superposition of individual FID's. The signal can be decomposed using a Fourier transform, enabling us to recover the individual frequency components and their magnitudes. This is the basis for NMR Spectroscopy, which is widely used in chemical analysis.

To overcome effects such as sample dephasing, in which the magnetisation vector begins to decay as the underlying spin states lose coherence, modified acquisition techniques must be introduced. In *spin-echo* NMR, the sample is subjected to another radio frequency pulse, of equal but opposite polarity. This has the effect of reversing the phase of all of the components which constitute the sample, while leaving the precessional direction unchanged. At a specific time, the spin *echo time (TE)*, all dephasing is reversed and a clear, sharp signal is obtained. By utilising polarised magnetic *gradient fields*, one can induce a functionally identical *gradient echo* in the sample. However, this approach is susceptible to field inhomogeneities and is therefore only used in certain special applications, such as rapid imaging.

2.1.2 Image Formation

Clinical MRI is usually based on Hydrogen (proton) NMR, since this element is present in almost all tissue types and has the added virtue of providing a strong NMR signal. Since different tissues have differing amounts of water, each will generate a unique signature which may then be interpreted as an intensity for image formation purposes. Collecting a single complex signal from a 3D volume is only useful for imaging if we can assign spatial coordinates to the individual frequency components i.e. determine where the signals originated from. Both



A B

Figure 2.3: K-space A) For the case of standard spin-echo imaging, each row of k-space is filled from left to right by sampling the FID for a given phase step. For other imaging techniques, different k-space trajectories may be used. B) The application of a (fast) Fourier transform yields the image.

2D imaging and 3D imaging exist, although the former dominates: the time and processing constraints associated with 3D imaging are still high.

In the case of 2D MRI, we treat the volume as a collection of 2D images, each of which represents a slice through the object in the acquisition direction. The slice to be imaged is selected using a linear *magnetic field gradient*, which is superimposed onto the main field. This has the effect of changing the Larmor frequency across the sample so that it becomes a linear function of displacement along the direction of the field. By applying an RF pulse of the appropriate strength and polarity, we can elicit a signal from the desired slice (technological constraints determine the minimum thickness this slice can be — under 1mm with current MRI machines).

The full imaging process requires the introduction of two additional linear gradient fields in orthogonal directions. The first provides *frequency encoding*: the gradient modifies the Larmor frequency within the slice, allowing one to select a given line or column of material. The second gradient is used for *phase encoding*, in which the phase of the spins vary as a linear function of gradient strength. By using these two complementary encoding methods we are able to encode the 2D position of a voxel within the slice.

In standard spin-echo imaging, we phase encode the y image direction and frequency encode the x image direction. These ideas are formalised by the introduction of *k-space*. In this description, the data matrix we seek to fill, commonly 256×256 samples, is treated as a frequency domain representation: the y direction represents phase, while the x direction represents frequency. See Figure 2.3. To create the image, we first select the appropriate slice and then trigger the phase gradient to select a row within the slice. The frequency gradient is then applied and the signal sampled while the field is maintained. The sampled FID is then copied into a row of the matrix and the phase gradient stepped in preparation for the next RF excitation, which may be initialised after the system has relaxed back to it's equilibrium state. The time, T_R, separating excitations is known as the *repetition time* and plays a fundamental role in the manipulation of contrast. This value is usually in the range 200—3000 milliseconds.

The process of stepping through the phase gradient would have to be repeated 256 times with the above matrix size. Furthermore, to boost SNR, the number of excitations (NEX) per matrix row may be greater than one, typically two or three. The total time required is given by

$$T = \mathrm{NOL} \times \mathrm{NEX} \times \mathrm{T_R}, \tag{2.3}$$

where NOL represents the number of lines (phase steps) in the matrix. It is also possible to acquire several rows within one repetition time by using a modified spin-echo pulse sequence. This allows one to produces several (differently weighted) images simultaneously, or to produce a single image more rapidly. In this case the time required is

$$T = \frac{\mathrm{NOL}}{\mathrm{NR}} \times \mathrm{NEX} \times \mathrm{T_R} \tag{2.4}$$

where NR is the number of echoes generated within T_R.

Once the k-space matrix is assembled, we may apply a discrete Fourier transform to arrive at a spatial representation. The image is prepared by assigning grey-scale intensities to each voxel that are proportional to the NMR signal originating from that spatial region.

In Figure 2.4 we show a typical image from an older MRI sequence. The image matrix size is 256×256 pixels. In this case non-contiguous slices were imaged. The slice width was 3mm, while the gaps between the slices were 2.5mm. The field of view (FOV) — the physical width/height covered by the image — was 24×24cm, giving a material voxel size of $1 \times 1 \times 3\mathrm{mm}^3$.

The MRI acquisition procedure described above is painfully slow, with imaging times ranging from 4 to 20 minutes for a single spin-echo image! Fortunately, newer techniques which are based on alternative pulse sequences, have drastically reduced the time needed to acquire data. Several of these are derived from modified gradient-echo schemes, such as RARE and FLASH [54]. The fastest method to date is echo-planar imaging [35], which is not based on the spin-echo technique. However, this sequence produces a lower SNR, which may limit its usefulness in detailed anatomical studies.

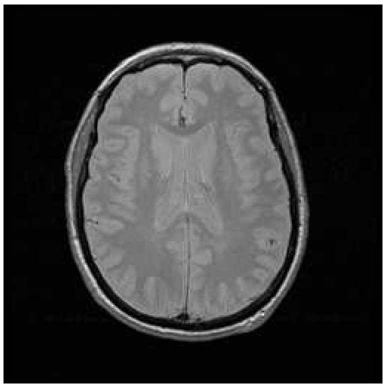


Figure 2.4: An MRI Image An image acquired under an older MRI protocol: the thickness of the slice is 3mm and each image in the sequence is separated by a 2.5mm gap. The FOV is 24×24cm.

2.1.3 MRI Slice Resolution

During image formation, a gradient magnetic field is used to select the volume of material from which the MR signal is to be acquired. The signal itself is generated by the application of an RF pulse at the appropriate Larmor frequency. Since the pulse has finite duration, its Fourier domain representation will have an associated frequency bandwidth. Consequently, the gradient field will not select an idealised slice in 3D, but a slab of material, the thickness and location of which are determined by the field strength and RF bandwidth. Furthermore, since the RF pulse is usually a Gaussian or truncated sinc function, the frequency response will not be uniform over the frequency range. Rather, the response will decay rapidly as one nears the limits of the frequency range. This manifests as a blurring of the selected volume boundary, since signal information will be captured from adjacent material. To reduce the overlap between neighbouring slices, consecutive image slabs may be acquired with an intervening gap. In this case, the gap is made wide enough to ensure that the no signal contribution arises from adjacent slices. The sparse images used in this thesis are examples of such acquisitions.

2.1.4 MRI Contrast

By adjusting TR, the duration of the RF pulses and the strength of the magnetic fields, the relative contrast of tissues within the imaged volume may be modified. Such contrast modifications can provide a significant aid to diagnosis. One can also use *contrast agents*, such as Gadolinium DTPA, to enhance specific pathology such as tumours. However, by judicious manipulation of imaging parameters, one can often form a conclusive diagnosis without the need for invasive indicators.

There are several standard image contrast protocols, each named for the aspect of the NMR

Intensity	PD	T1	T2
	long TR, short TE	short TR, short TE	long TR, long TE
High		Fat	CSF
	Grey, White Matter	Bone Marrow	
	Fat	Grey, White Matter	Grey, White Matter
	CSF		Fat
		CSF	
	Cortical Bone	Cortical Bone	Cortical Bone
Low	Flowing Blood	Flowing Blood	Flowing Blood

Table 2.1: MRI Contrast-Weighting Schemes This table shows the relative brightness of various tissue types for given MRI contrast weighting schemes. For spin-echo imaging, the repetition time (TR) and the echo time (TE) are the parameters which select the particular image weighting; the relative sizes of these parameters are indicated in the first row of the table.

signal they most emphasize. For example, in a *T1-weighted (spin-echo) image*, the RF pulse repetition time is too fast to allow significant relaxation of the magnetisation vector. As a consequence, tissues with long T1 relaxation times will lose contrast relative to those with shorter times, since the magnetisation will be tilted back into the transverse plane before having recovered its full strength. Two other popular weighting schemes are T2, the complement of T1, and Proton Density (PD), which gives an indication of the Hydrogen density across the image. See Table 2.1 for details (reproduced from [54]).

The spin-echo signal intensity is given by

$$\mathrm{SI} = \mathrm{K}\rho(1 - \mathrm{e}^{\frac{-(\mathrm{T_R}-\mathrm{T_E})}{\mathrm{T1}}}\mathrm{e}^{\frac{-\mathrm{T_E}}{\mathrm{T2}}}) \tag{2.5}$$

where K represents the influence of various environmental parameters and ρ is the proton density. This equation describes the effect of RF pulse timing on the image contrast, and hence whether the signal is T1 or T2-weighted.

There is no easy way to determine visually the particular weighting given to an image, since there is a degree of overlap between each protocol. Certain heuristics may, however, allow one to make an educated guess. For example, a T2-weighted scan will have a high signal (bright intensity) for water-based substances, such as CSF, while CSF will appear dark on a T1-weighted scan. Proton-density scans are less easily differentiated. In the interests of clarity, we shall explicitly state the weighting scheme whenever we present MRI data.

2.1.5 MRI Artifacts

The MR imaging process is susceptible to a number of artifacts which can complicate image interpretation.

Inhomogeneities in the MR receiver coil and the magnetic field can result in a low frequency

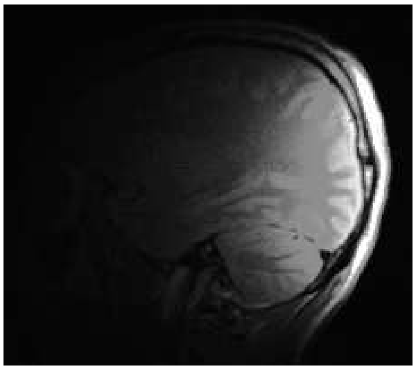
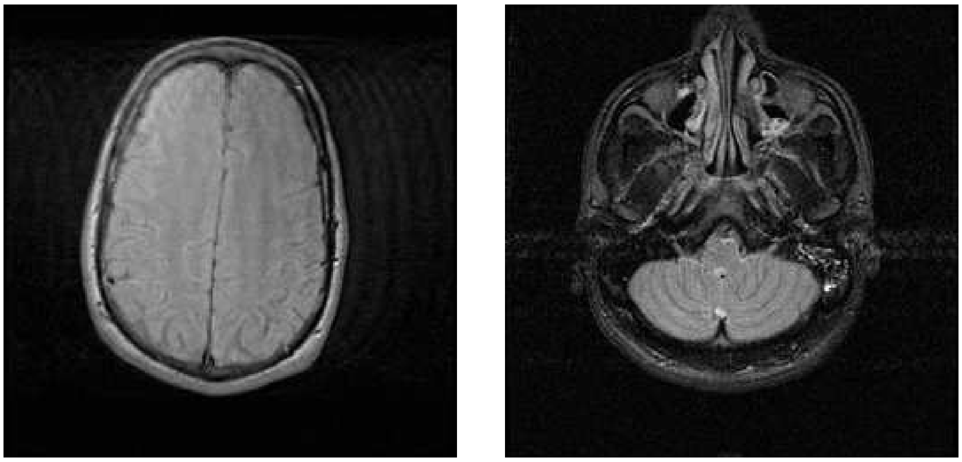


Figure 2.5: Bias Field This image exhibits a bias field artifact, which is caused by inhomogeneities in the MR receiver coil. The contrast in this image has been manipulated to accentuate the artifact. (Image courtesy of W. M. Wells, Harvard Medical School)

multiplicative signal, the *bias field*, corrupting the image. An extreme example of this effect is shown in Figure 2.5.

There are also more destructive artifacts, particularly those associated with motion. Patients are seldom still: each twitch is translated into ghostly blurring or worse [35, 54]. Examples of such images are shown in Figure 2.6. Furthermore, the NMR characteristics of a region may be affected by blood flow and local field inhomogeneities, complicating intensity based analysis further.



A B

Figure 2.6: Imaging Artifacts A) PD-weighted MRI scan with severe motion "ghosting". B) Horizontal bands of noise reveal the presence of rapidly flowing arterial blood in this T1-weighted image.

If the object being imaged is under-sampled (when the FID is sampled too coarsely) one may observe aliasing effects in the reconstructed data: structures beyond the field of view are erroneously mapped or "folded back" into the image. This artifact can be eliminated during image acquisition by oversampling or increasing the time over which the MR signal is collected.

There are a multitude of other artifacts, but the ones discussed above represent those most commonly encountered in the data used for this thesis. A more detailed discussion can be found in Rinck [54].

2.2 Characterisation of Sparse MRI Data

The MRI data we wish to segment was acquired under a long term study of anomalous brain structure in schizophrenics. Data from such extended clinical studies is rare, and special segmentation techniques are required to cope with many of the older scans in the series — the bulk of the data was imaged using old spin-echo protocols which have poor resolution perpendicular to the slice plane. Although there are a number of existing techniques for the segmentation of high resolution data, these methods would not work well on such coarse data, hence the need for alternative sparse segmentation techniques. In this section we describe our data in more detail, as well as describing the problems it poses for segmentation.

2.2.1 Sparse MR Images

The term "sparse MR" was coined to describe the class of MR images which have a high sampling rate within each image slice, but a comparatively low sampling rate in the direction perpendicular to the image plane. Since the image is formed by sampling the FID within the Fourier domain, the dimensions of the k-space matrix directly determine the scale (spatial frequency) of the structures which can be discerned when the image formation DFT is applied. The k-space matrix represents a uniform sampling of the Field Of View in the x and y directions. For example, a 128×128 matrix will yield a poorer (more blurred) image than a 256×256 matrix for a fixed size FOV. In sparse images, the sampling rate is sufficiently high to enable structure on the scale of about 1mm to be represented within each image. However, due to the effective sampling distance *between* each slice, the structure is degraded in a very specific way, described in Section 2.2.3 below, as one moves through the stack of images. Furthermore, with the large inter-slice sampling distance used in this work, there is no possibility of detecting similar fine-scale structure in the perpendicular direction, even when the in-plane degradation is minimal.

2.2.2 Sparse MR Image Database

The sparse images used in this work are 17 or 20 slice spin-echo images. The image slices are 5mm thick and separated by a gap of 2mm, with a typical FOV of 24×24cm. The image matrix size is 256×256, and the number of excitations (NEX) is variable. All the standard contrast-weighting schemes (T1, PD, T2) are present, but the particular choice was left to the discretion of the radiologist. In certain cases, interleaved PD and T2 images were acquired. The magnet strength used 1.5T. Both axial and coronal scans are present, with the former being more numerous. Subjects were imaged at a number of consecutive time-points. The period of the study spanned several years, and high resolution (coronal) scans were introduced near the end of this period. Such temporal MR sequences provide a record of the progression of the disease.

The age distribution of the subjects ranged from 20—60 with the bulk of the subjects lying within the lower half of this range. The sex ratio was approximately even, but favoured males. Handedness was predominantly left.

2.2.3 Segmentation Issues

There are a number of issues which are of particular relevance to the segmentation of sparse data:

Partial Volume Effects Due to the large size of the voxels, sparse data is particularly susceptible to the *partial volume effect (PVE)*: the averaging of signal intensities that occurs when a voxel is occupied by more than one tissue type. As a consequence, it may be difficult to discriminate one voxel from another and boundary information may be lost. For the elongated voxels ($1 \times 1 \times 5\text{mm}^3$) arising from the sparse protocol, the PVE is maximal in the parts of the brain where the tissue boundary surface normal is parallel to the direction of the longest voxel axis (near the brain poles for axial scans). In this part of the brain, voxels near tissue boundaries are more likely to contain multiple tissue types (because of voxel elongation). In these regions, the signal

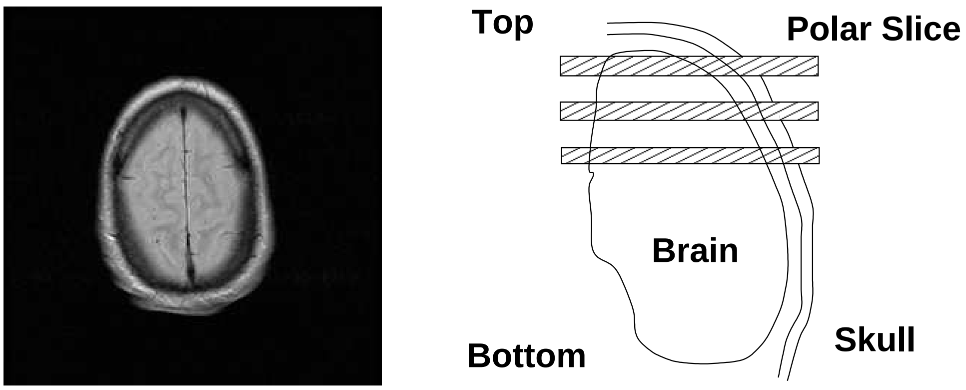


Figure 2.7: Partial Volume Effects The PVE found in sparse data are particularly severe near the poles of the brain. In this region the tissue surface normal is approximately aligned with the direction of the longest voxel axis. In the vicinity of a tissue boundary this results in a higher proportion of voxels containing more than one tissue type.

intensity is smeared over a spatial region and structural information is obscured — Figure 2.7. Since segmentation involves the detection of differing tissue types, and these are indicated by different signal intensities, the PVE must be tackled before we can proceed.

Data Truncation Sparse scans are frequently *truncated.* This simply means that parts of the brain are not imaged. The combination of thick slices and a non-zero slice gap ensure that some peripheral structure will always be missed. As a consequence, the top and bottommost slices contain structures which terminate abruptly, as shown in Figure 2.8. If we wish to reconstruct an approximation to the brain surface, in order to

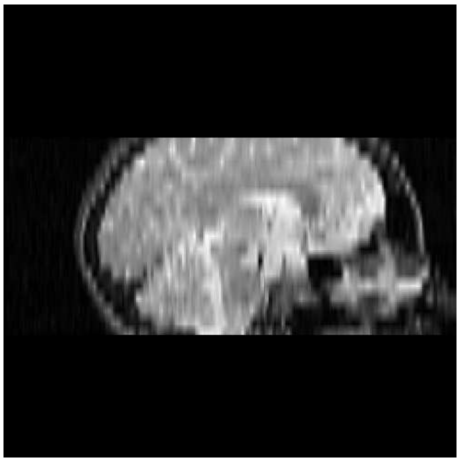


Figure 2.8: Data Truncation In this image, a resampled saggital cross-section has been generated from a truncated sparse T2 axial scan. If we wish to compute meaningful comparative (volumetric) measures across subject, we must have a mechanism to interpolate missing data.

quantify volumes, for example, we need a meaningful way of interpolating this missing data. Unfortunately, the issue is complicated by the arbitrariness of this truncation: patients' heads may be displaced relative to one another within the machine, giving rise to a significant variation in the location of the truncation.

Image Artifacts Although the sparse protocol dates from a time when MRI technology was less sophisticated, the data we have seems relatively free of serious artifacts. While there is some evidence of ghosting and blood-flow effects, as well as occasional aliasing, the majority of the images are fairly clean. Although the bias-field artifact is not readily apparent, one can apply bias-correction to ensure the best possible input for the segmentation algorithms. It should be noted, however, that our segmentation, which is based on the detection of boundaries, is less susceptible to bias-field anomalies than a scheme which seeks a voxel classification. This is because the bias field is multiplicative, and will not shift tissue boundaries. Naturally, there are pathological cases in which the edge might be obliterated, but there is no evidence of such scans within the database used in this thesis. We do not consider PVE to be an imaging artifact. The MR scanner is recording precisely what it was intended to: the tissue content of a voxel. Of course, a mechanism must still be found to cope with PVE, and we shall suggest such a scheme later in this thesis.

Image Type The particular contrast scheme (T1, T2, PD) and the acquisition direction (axial, coronal, saggital) will both have an effect on the algorithms needed to segment the data. The effect of contrast is obvious: the tissue boundary model is directly dependent on the image intensities. The role of the acquisition direction is less clear. The cerebral surface in sparse axial scans appears significantly smoother when compared to the corresponding coronal view: in the latter case the sulci are clearly visible, revealing highly convoluted boundary structure. Given the sparsity of the data, the tracking of individual sulci is not feasible, as there is little continuity from slice to slice. In the case of axial and (to an extent) saggital scans, we may utilise a "cling-wrap" type surface which will ignore small declivities. Application of a similar scheme to coronal scans could lead to a surface representation which contains features such as cusps, inter-penetrations.

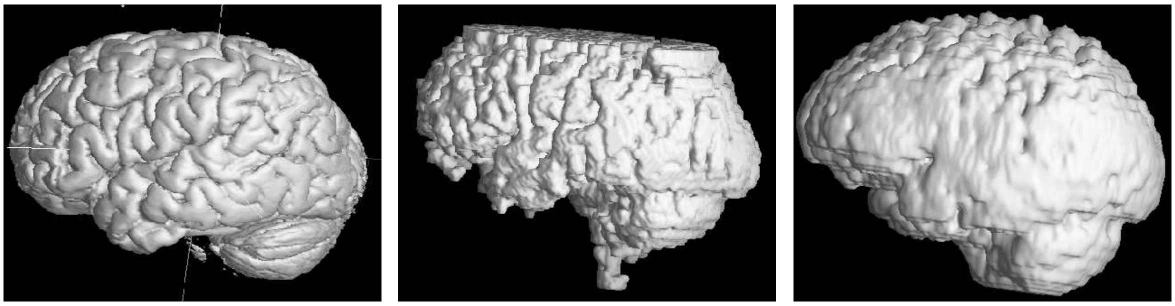
We now examine a number of existing schemes which have been utilised for the task of segmentation.

2.3 Overview of Segmentation Methods

There are a number of generic algorithms for tackling the problem of volumetric image segmentation, the complexity and robustness of which vary considerably. At the lower end of the complexity scale are what we have termed *voxel-classification schemes* which utilise simple intensity models to segment the data. An example of such a scheme would be voxel-intensity thresholding. To improve the robustness of the segmentation scheme, one requires models of greater sophistication. In certain cases, intensity-based schemes may result in gross mis-segmentations, such as the disconnected regions for a solid object, unless the algorithm is augmented by some sort of shape model. We shall call such schemes *shape model based*, to emphasize their added functionality. It should be noted that this classification is intended to provide a context for our own segmentation scheme and may differ from the "taxonomy" presented by others. More detailed discussions of relevant work will be presented as required.

2.3.1 Voxel-Classification Schemes

In many volumetric MRI studies, the image is segmented using a *voxel-classification scheme*, in which each voxel is allocated to one of a specified number of tissue classes. Once this classification has been accomplished, an estimate of tissue volume may be obtained by simply counting the appropriate voxels and multiplying by the voxel dimensions.



A B C

Figure 2.9: Voxel Segmentation Surface renderings of voxel segmentations for several different MR resolutions. Observe that the classification has retained the cerebellum and brain stem: these structures cannot be easily excluded on the basis of intensity information alone. A) High-resolution ($0.8 \times 0.8 \times 1.5\text{mm}^3$) postmortem brain. The gyrification of the brain surface is clearly visible. B) Low resolution ($0.9 \times 0.9 \times 5.5\text{mm}^3$) axial scan, with no inter-slice gap. Note the truncation at the top of the brain, and the roughness of the iso-surface (which has already been smoothed). There is also evidence of incorrectly segmented structure near the eye. C) Low resolution with an additional gap (2mm). The slice-wise coherence of the sulci is very badly affected (Images courtesy of Guido Gerig, ETH, Switzerland).

Any scheme based on intensity thresholding alone falls within this category. In many instances, a spatially varying intensity threshold is required for acceptable results. For example, in [72], voxels corresponding to the cerebral vessel tree were extracted from Magnetic Resonance Angiography (MRA) data using a scheme based on multi-resolution adaptive thresholding. In work undertaken by Zijdenbos [74], the intra-cranial contour (IC) is extracted on a slice-by-slice basis using a method which computes a spatially varying intensity threshold. Application of the threshold produces a binary image which may be processed by simple morphological operations to yield the IC. Unfortunately this method only works on transverse MR images. Furthermore, the radial sampling used to extract the IC make very specific assumptions about the nature of the binary image and the properties of the boundary to be extracted. Such limitations are unacceptable in a general MR segmentation framework.

Region growing techniques (see, for example, [56]), in which neighbouring voxels are accumulated based on their relative intensities, are also a form of voxel classification. An example of such a scheme is provided by the "intelligent paint brush" [48, 49] developed for the segmentation of MR images. A "watershed algorithm" is used to create a hierarchical region-based representation of the image. When the user selects a ROI within the image, the hierarchy is traversed by a selection algorithm and the regions which most closely match the desired ROI grey levels are determined and a boundary computed. Although such a scheme is useful in the context of general interactive image analysis, there is no high level facility to enforce boundary shape constraints, and the method — as demonstrated — is only suitable for the semi-automated processing of 2D images.

An interesting approach to voxel-based segmentation, which attempts to compensate for the bias field artifact, was introduced by Wells and Grimson [70], and later extended by Guillemaud and Brady [39] and Brechbuhler et al [14]. This method uses an Expectation-Maximisation algorithm to iteratively generate a bias-corrected voxel classification for a given number of tissue classes. Unfortunately, it is not always clear *a priori* how many tissue classes are required, or how to deal with tissue classes for which the intensity distributions overlap to a significant degree.

When multi-modal image data is available (such as PD and T2), one may use data from each protocol to produce a more robust classification [46, 42]. This approach assumes that the images are registered, which need not be the case: patient movement cannot be ruled out over the scanning period.

The "hierarchical segmentation" introduced by Griffin et al [37] uses intensity homogeneity and "edge strength" to build a region-based image decomposition. Unlike the other schemes outlined above, this scheme allows one to enforce rudimentary constraints on the segmentation process. The constraints are posed in terms of pixel membership of overlapping regions, and are to be supplied by some "higher-level" logic. For medical image segmentation, at least, such a constraint methodology is of limited usefulness. We require structural constraints which are not easily codified on a pixel level, and the means by which such constraints are generated must, of necessity, be specified too.

An *iso-surface*, constructed from the bounding voxels of a given class segmentation, may be used to estimate other useful geometric quantities, such as surface area and curvature. Measures computed from this surface representation are fundamentally limited by the voxel size.

In its most basic form, MRI voxel-classification assumes:

1. **Continuity** The data are assumed to be contiguous i.e. each voxel is connected along each face to neighbouring voxels, except along the boundary of the imaged volume, and

2. **Voxel Class Membership** Each voxel may be classified as belonging to a *single* tissue class.

For sparse MRI the assumption of continuity is violated since the volumetric images are sampled so coarsely in the acquisition direction. Furthermore, "corrective" voxel interpolation exacerbates the segmentation problem, since we both increase the effective voxel size and lose intensity information. A little thought will reveal that the second assumption, voxel class membership, is also violated: sparse data voxels frequently contain significant tissue contributions from several classes, resulting in an averaged signal. The impact of PVE on the sparse segmentation problem can be profound. A probabilistic voxel classification method, in this framework, can do no better than to choose the most likely tissue candidate and classify the voxel accordingly, disregarding the contributions from other classes in the process. Within the sparse framework, any iso-surface approximation based on a voxel level segmentation is likely to provide a poor estimation of the underlying surface geometry. Voxel segmentations for high resolution and sparse images are shown in Figure 2.9. Observe that the segmentations include both the cerebellum and the brain stem, since these are also composed of cortical matter. The structure of the sulci is badly compromised when the MR data includes an inter-slice gap.

An alternative approach, which deviates from the basic paradigm outlined above, is to utilise a *probabilistic hyperstack* [68]. This multi-scale representation sub-divides voxels to a specified level and attempts to predict which higher level structures each sub-voxel belongs to. While this is an interesting idea, the poor resolution of sparse data would make prediction of structural relationships at a sub-voxel scale implausible.

In practice, one needs some sort of underlying model for all but the most trivial segmentations. This model may, for example, represent the spatial variation of intensity across an object or serve to delimit permissible variation of a shape. A fundamental limitation of most voxel-based schemes is the lack of structural constraints: given that voxel mis-classification is such a strong possibility, we need a mechanism to reject spurious segmentations.

2.3.2 Constraining Shape

The image data we wish to segment may contain missing data or significant image degradation. In these cases we may improve the robustness of the segmentation by employing a *deformable model* which embodies *a priori* topological and/or geometric constraints. This model is usually a surface (or contour) with an associated algorithm for extracting shapes of interest.

Arguably the most popular type of deformable models are active contours or *snakes* [44]. These are energy minimising curves (or surfaces in 3D) which evolve towards features of interest in an image (or volume). In [44], the features of interest were edges, and the energy which governed the evolution of the curves was a combination of an internal energy, which constrained curve flexure, and a image-wide gradient energy, which was maximal in the vicinity of edges. By choosing an appropriately defined energy term, one can select alternative features of interest, such as textures [43]. Modifications to the basic snake methodology abound. For example, to tackle the problem of missing data, an *expansion term* may be introduced [21, 22], which adds an expansive force in the normal direction along the curve. These ideas may be generalised to 3D, although the computational overheads increase and the additional degree of freedom can complicate or even frustrate implementation. In the case of 3D segmentation, the introduction of *a priori* knowledge may significantly ease these problems. For example, in [63] the authors use a snake surface with an embedded axial symmetry constraint to segment structures with cylindrical topology. Unfortunately the brain has a complicated geometry which is not easily modeled.

There are many permutations of the above schemes. For example, Blake et al [8] developed a model which uses a snake formalism complemented by a shape template constraint: only shapes related to the template by an affine transformation are permitted. While the affine constraint is appropriate for the particular context within which the system was proposed, it would be too restrictive for use in our work, since brain variability has a significant non-linear component. This method also has certain other interesting features. For example, the global "image potential", which moves the snake towards features of interest, is replaced by local feature searches and a modified update scheme developed. This reduces computational overheads and allows one tighter control over local snake evolution.

Parametric surfaces such as superquadrics and their generalisation, hyperquadrics [20, 64, 41], have been widely used as a means of enforcing simple shape constraints on the surface model. Unfortunately, the complexity of the shapes these models can represent tend to be rather limited. An interesting approach which provides increased shape discrimination is the use of *free-form deformations* [3] after a rough segmentation has been achieved with a parametric surface (a superquadric in this case). However, the number of additional parameters required to accurately represent a shape grows rapidly as data becomes more complex, limiting its usefulness for the segmentation of convoluted structures.

The *deformable templates* introduced in [47] employ parametric curves/surfaces to represent the shapes of interest. The model fitting is accomplished by a global energy minimisation over the set of shape parameters. This approach is not suitable for the complex geometry of the brain: obtaining a parametric representation of such a complex object is a rather difficult task. Furthermore, such a representation, assuming it could be generated, would require a large number of parameters resulting in a search space of extremely high dimension.

The deformable Fourier Surfaces introduced by Staib and Duncan [61] provide an interesting blend of segmentation and analysis functionality. A discrete mesh of simple topology is decomposed on a Fourier basis, and a vertex update scheme developed which allows a constrained mesh deformation in the direction of the data points. These ideas were taken further in the work of Szekely et al [62], who applied this deformable surface to the segmentation and analysis of structures within the brain (the corpus callosum and deep grey matter surfaces). In their work the Fourier descriptors of the final surface representation were used for the investigation of pathology — essentially they use a Euclidean metric to compare vectors of descriptors from different patients. Unfortunately the Fourier representation requires that a surface parameterisation be established. Although a method for obtaining a surface parameterisation was suggested by Brechbuhler [13], only closed surfaces with no holes may be processed. This means that the ventricular system, for example, cannot be decomposed onto a Fourier basis using this method. We desire a scheme which can represent complex objects with (potentially) many holes.

The lack of a general surface parameterisation scheme for arbitrary topologies means that the techniques discussed above are fundamentally limited. A number of "non-parametric" methods have been developed to address this issue. For example, the "implicit deformable model" introduced by Whitaker [71] is based on a discrete surface model which evolves under a series of differential equations. In a somewhat different vein, the topological snakes of McInerney et al [51] are defined on a "simplicial grid" of fixed resolution. By decomposing the polygonal snake curve onto the simplicial grid, the evolving snake contours can be made to merge or split under the action of image forces. While such a simplicial grid may be generalised to 3D, as the authors have shown, there is no easy mechanism to enforce shape constraints: the curves or surfaces will evolve to fit the most likely edge candidates (those selected by the image potential). If there is a high level of images noise, or edge degradation, a poor segmentation is likely.

The most common non-parametric schemes involve the direct evolution of a vertex set representing points on an underlying surface for which no parameterisation (implicit or otherwise) is assumed. There are a number of schemes based on surface triangulations under which the mesh vertices evolve in a snake-like fashion towards feature points in the volume. The vertices are usually assumed to be connected by a series of perfect springs, and the appropriate Newtonian force model is invoked (see, for example, [67]). A closely related approach developed by Nastar and Ayache [53] adds "volumetric springs" (vertex connections through the volume) to increse stability and better model the object properties. They also utilise a *modal analysis* (a multi-scale basis decomposition, similar to Fourier decomposition) to obtain a compact set of shape descriptors which may be used for subsequent analysis. While this approach is very general, there is no mechanism to enforce shape constraints (a shape model), and because of the non-locality of the volumetric springs, local perturbations may affect distant parts of the mesh.

The most serious shortcoming of the segmentation schemes discussed above is the lack of an explicit shape constraint. This constraint (or model) should encode shape geometry and topology for surfaces with complex structures, whilst being sufficiently general to represent the variability present within the shape class.

An elegant approach encompassing shape model construction and segmentation is the *active shape model* (ASM) introduced by Cootes et al [24, 23, 25, 26]. By examining a set of normalised training shapes one can build a point distribution model (PDM) which encodes the shape class variability. The model is a discrete representation in which every point has an associated mean and variance that constrain its displacement. The PDM is the result of performing a principal components analysis on the set of corresponding points. This yields the dominant "shape modes" which span a reduced linear subspace within which learned shapes may be represented with only a few parameters (basis or modal coefficients). This modal representation provides a simple framework to compare shapes within the class, via their modal coefficients.

A more recent technique uses a *simplex mesh* [30], a discrete surface model composed of vertices with 3-connectivity, to represent the shape model. This model deforms under the action of image and internal forces. If data is not available the affected region tends to relax towards the shape defined by the template. There is also a well defined set of local mesh transformations which enable one to dynamically modify the topology and node-density of the template. The 3-connectivity property ensures that one can derive simple and symmetric algorithms to process the mesh structure. Like some of the other mesh-based segmentation schemes presented above, the simplex mesh is governed by a Newtonian force model. However, the internal forces acting on the model are local and can be forced to respect an overall shape constraint [52].

The simplex mesh framework provides a very general infrastructure for shape representation which includes local surface updates and a complex shape model. However, it does not address the issue of shape model variability or the detection of boundary data in a degraded image (such as sparse MRI). The ASM addresses the issue of shape variability through a training process, but does not specify any particular mesh representation, and likewise assumes that boundary detection is (near) trivial. The ASM also assumes that only shapes within (or very similar to) the training set are valid. This is not necessarily the case if pathological structures are to be segmented. By combing the different shape model capabilities of both schemes, and addressing the issues involved in boundary detection within sparse images, we believe that a fairly general, robust sparse segmentation scheme can be developed. The active shape model and simplex mesh will be examined in considerably more detail in Chapter 4.

2.4 Conclusion

Magnetic resonance imaging permits a wide choice of acquisition protocols. These protocols determine scan attributes such as contrast and resolution and have a profound impact on the strategies one can utilise to segment them. In the case of sparse data the combination of truncation artifacts, slice gaps and large voxel size confound generic segmentation schemes. However, a synthesis of existing schemes augmented by domain specific knowledge can allow us to overcome these limitations. The key aspect of this new methodology is the development of robust shape models that can provide a meaningful extrapolation of missing object boundary data. The modelling process must be flexible enough to represent the extensive variability present in biological structures, whilst maintaining a level of detail which is appropriate to the sparseness of the data. We now turn our attention to the development of such a framework, starting with the evaluation of a 2D snake-based segmentation strategy.

Chapter 3

Snakes

An MR image is composed of a stack of 2D images, each of which represents a cross-section through the object of interest as one steps along the imaging axis. If the distance separating these images is sufficiently large it no longer makes sense to view the scan as a volumetric entity: it is a composite of 2D cross-sections. Since the segmentation of a 2D image is often computationally cheaper and more tractable than that of a volumetric image, this alternative interpretation is appealing. Of course, these images are not entirely independent, and it would be foolish to ignore this *a priori* knowledge.

Inspection of high resolution MR data reveals that 1) scanned objects produce closed intersection curves in the imaging plane and 2) these contours are geometrically similar over adjacent slices, in the sense that their gross shape and location exhibit only small changes. This structural coherence still exists, although somewhat tenuously, in the case of sparse MR images. Our segmentation strategy should capture both these elements. If we were able to reliably determine boundary pixels, we could connect them to produce the curve we desire. Unfortunately, image noise and artifacts such as PVE make such a scheme infeasible. Furthermore, such pixel/voxel based schemes are not amenable to the introduction of shape constraints which limits their usefulness. However, if we utilise a so-called *snake* contour model we may simultaneously address the issue of boundary detection and representation: the snake contour will gravitate towards features of interest in the image (the boundary, in this case). Additionally, the snake possesses intrinsic regularising properties which enable it to provide a smooth approximation to the boundary in the presence of image noise.

Given these considerations we decided to implement a snake segmentation scheme, augmented to include our adjacency constraint. This constraint is implicitly utilised through a curve propagation strategy, under which snakes representing the boundaries of interest in a given image are used to initialise the search in the following image. The slice contours extracted using this approach may be used to generate a 3D surface representation suitable for calculating symmetry metrics.

This chapter describes the implementation and evaluation of such a scheme.

3.1 The Snake Framework

A snake is an elastic model for shape which can be coupled to image features. Snakes were originally developed [44] as "active contours" which move under a combination of "internal elastic forces" and "external forces" arising from image features, such as edges. A number of modified snake schemes have since been devised which are not only more efficient but also have more sophisticated methods of incorporating *a priori* information [8, 11, 47, 64].

In the original formulation, the snake was interpreted as the solution to a global (variational) minimisation problem. This approach requires the definition of an "image potential" to act as an attractor for the snake. Such a scheme provides only rudimentary shape constraints, through implicit continuity conditions. An alternative scheme which addresses *both* computational complexity and the addition of a shape model was proposed in [8]. In this framework the snake is explicitly represented as an *approximating B-spline* (cf. Appendix A.1) which has the following features:

Template A shape model or *template* may be incorporated into the update process. This model constrains snake deformation in a prescribed way; in the original work only affine deformations of the template were permitted.

Local Computations Rather than using a "potential" image to represent features of interest, a local feature search is introduced. For the detection of edges, a line of pixels normal to the spline curve is extracted at specified points along its length. More specifically, the polynomial "spans" which constitute the snake are uniformly sampled and these points used as the origins for the profile extraction. A 1D edge search is then performed within each intensity profile and the location of any edge candidates within these lines are noted. If a sufficient number of edge candidates are deemed "good" over a given span, that portion of the snake is considered to be **locked** onto the data and these points will participate in the update process. The update is implemented by re-fitting the spline curve to the set of locked points, subject to the model constraints. As a consequence of this locality, the update process reduces to a sparse matrix system which may be solved rapidly.

Closed form expressions The use of such approximating spline snakes allows the use of analytic expressions, for example based on Green's theorem, for computing geometric quantities such as length, curvature and area — Appendix A.2. These quantities can be obtained efficiently and directly from expressions involving only the control points [19]. Although these expressions are also available for interpolating snakes, they are not as readily obtained or as easy to manipulate.

For our purposes such a scheme is preferable to the canonical snake approach, since the explicit curve representation provides a natural framework for the inclusion of *a priori* constraints.

3.1.1 Modified Framework

Although we wish to include explicit shape constraints in our snake, the template approach suggested in [8] was not appropriate for our needs. In the original context, a *single* shape was to be segmented in each image, and each shape instance was essentially related to the reference shape by an affine transformation. In our work, the snakes are required to represent the intersection of a 3D shape with a series of planes. Since the sampling is coarse, the resultant intersection curve(s) may change substantially between slices; this change may be topological i.e. the curves may merge or split as we step through the image stack. Consequently, the construction of a 2D shape model was deemed infeasible — it would have to encompass both geometric and topological variability whilst remaining invariant to patient alignment. Instead, we opted for a small set of constraints which encode low-level knowledge of our problem.

More specifically, we replaced the template with the following:

Matched Boundary Searches Different biological structures have different edge characteristics, and the edge search is tuned to these characteristics. For example, a valley/ridge detector is used to isolate the inter-hemispheric fissure, whilst a contrast sensitive step detector is used to isolate the external cerebral boundary. Furthermore, the features that the snake searches for are allowed to vary along the curve, permitting *a priori* information about boundary structure to be incorporated into the segmentation process. The boundary constraints are propagated with the snake and remain static.

Interactive Modification At any stage in the update process a user can manipulate the snake, moving it around and adding or deleting control points if necessary. This level of control is necessary for medical segmentation, since no scheme is guaranteed to give perfect results.

Prediction of Snake Evolution The snake is represented by its control points. These trace a trajectory through space as we step through the sequence of slices. We can use a linear prediction scheme to help us guess where the snake is going to go on successive slices — this gives us a way of reasonably propagating the snake so that it has a good chance of "locking on" to the boundary in the next slice. Unfortunately, the number of control points may change as the snake evolves (we may need to add some to improve flexibility for example). Rather than predicting the motion of each control point, we predict the motion of the *centroid* of the control points as well as the relative expansion of the snakes (as determined by the ratio of their areas). This is an unsophisticated global measure, but if we wish to decouple the prediction step from the actual representation such a compromise seems inevitable.

3.2 Segmentation

The modified snake framework outlined above served as the basis for a segmentation and measurement system which is currently in clinical use. The various components of this system are described below.

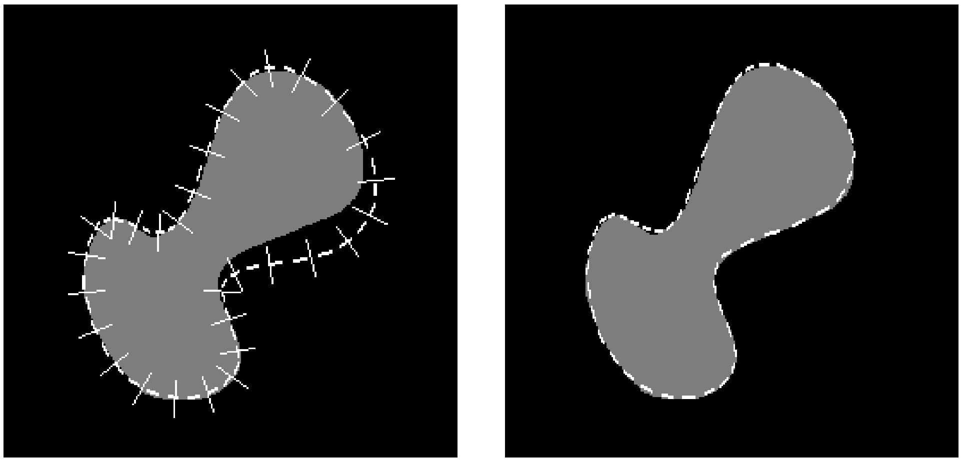
3.2.1 Initialisation Issues

The snake is initialised on the first slice of interest and allowed to propagate across slices, with possible intervention by the user. There are several issues which need to be addressed before the process can be initialised.

Control point density The continuity conditions imposed on the B-spline snake implicitly constrain its shape. When laying down the curve, we must be sure to place sufficient control points to allow the snake the flexibility it requires to represent the boundaries of interest. If we have too many control points we essentially lose all regularisation (each control point adds two degrees of freedom), so some sort of balance must be struck. In the scheme we have developed, the operator delineated the initial boundary and control points were spaced regularly along its length. The default spacing could be manipulated if required. Alternatively, the snake could be defined by directly placing control points on the image and moving them about. This approach was preferred by the clinicians who used the software.

Sample line density The number of sampling lines per span was determined empirically and was static once selected. The number was set to be three or four. The control point density also affects the efficacy of the edge searches: if we have too many edge searches in a region where the control point density is high, then we increase computational time in return for little improvement.

Snake labelling For the boundary-specific edge searches to be used, the operator must highlight sections of the curve and indicate which boundary type they correspond to. This must be done for each intersecting contour, but is only required when a new snake is created.



A B

Figure 3.1: Snake Feature Search A) A closed cubic B-spline snake, showing the one dimensional search lines. The snake is "locked" onto the top of the grey region, but not on the part to the lower right where the grey region is currently outside the search range. After a small number of iterations the unlocked part of the snake is pulled onto the feature boundary by the rest of the snake. B) After 11 iterations the snake has reached equilibrium and area can be measured.

Snake number and Topology The number of snakes required depends on the number of intersecting contours the object makes with the image plane. This number will in general vary from slice to slice. Because of the sparse nature of the data, there is no clear way of predicting such transitions based on slice data alone. Consequently, the user is required to delete or add snakes when such events occur.

3.2.2 1D Feature Detection

Image features are located by searching along lines emanating from the current position of the snake — Figure 3.1. These *search lines* are normal to the curve and spaced at regular intervals along its length. The search occurs at a specified scale, which is defined by a *search length* on each line, and selects the point of maximum contrast on each line. If there is no contrast above a specified threshold, the search line is marked as *un-locked*, and may not contribute to the update of the snake parameters (see Section 3.2.3). Both the position and signed contrast of the edge point are recorded. Retention of the sign is useful for feature disambiguation: we may safely reject edge candidates with the wrong contrast sign, even if they have a high contrast value. The contrast threshold used was empirically determined and has remained the same throughout our experiments. Although the number of search lines per span can be made variable, we found that three per span was adequate for our purposes. The set of these point positions, measured over the entire snake, is used to update the snake's parameters.

3.2.3 Position Update

The snake is deformed so that it fits as closely as possible to the set of detected feature points. It is possible that some span searches may not have detected any edge points (i.e. they have not established a lock). Even when this occurs, they can still move, since the locked portions of the snake drag them as they shift to fit the new contour.

Algebraically, we wish to find new positions for the control points so that the distance between the spline (at the sample point) and measured feature is minimised. This is achieved by minimising the following cost function over the control point positions:

$$\min_{\underline{\mathbf{X}}} \mathcal{E} = \|\mathbf{v_x} - \Gamma\underline{\mathbf{X}}\| \tag{3.1}$$

where $\mathbf{v_x} \equiv (x_{1,1}, \cdots, x_{1,p}, \cdots, x_{N,1}, \cdots, x_{N,p})^T$, is the vector of x target data. More precisely, $x_{j,k}$ is the x coordinate data for the kth sample on the jth span. For convenience, we assume that each span is sampled p times; in the experiments reported below, $p = 3$. The data for the spans (p items each) are listed in turn, giving a total of pN data. The control point vector is arranged similarly: $\underline{\mathbf{X}} \equiv (X_1, \cdots, X_l)^T$.

The matrix Γ is $((\gamma_1^x)^T, \cdots, (\gamma_N^x)^T, (\gamma_1^y)^T)^T$ with (see Equation A.3):

$$\gamma_k^x = \begin{pmatrix} 1, s_1, \cdots, (s_1)^d \\ \vdots \\ 1, s_p, \cdots, (s_p)^d \end{pmatrix} \mathbf{B}_k \mathbf{S}_k. \tag{3.2}$$

where the parameter s is evaluated at uniformly spaced intervals along the span and the matrix $\mathbf{S}_k$ is a selection matrix of size $d \times l$, which selects the control points for the appropriate coordinate on the kth span.

A similar expression is minimized to determine $\underline{\mathbf{Y}}$.

Dealing with partial lock If no lock is established for a particular search, then we have two options. We may ignore that failed search entirely i.e. use a reduced matrix with no entry for that sample, or we may choose to constrain the cost function somewhat by ensuring that the old sample is still considered. In the latter case, we just retain the old value i.e., the target value is set to the old sample value. The effect of this is to add stability to the solution of the minimization at the cost of slowing or possibly stalling the motion of the unlocked part of the snake. Currently, both modes of operation are implemented. In practice, the constrained mode is often a better choice, since it tends to converge (albeit more slowly, on occasion — see below) to the contour, whereas the unconstrained mode permits the definition of an ill-conditioned linear system (when the number of control points or samples is small).

The cost function $\mathcal{E}$ in Equation 3.1 can be minimized using a pseudo-inverse:

$$\underline{\mathbf{X}} = (\Gamma^\top \Gamma)^{-1} \Gamma^\top \mathbf{v_x} \tag{3.3}$$

In this case, minimization of the expression for $\underline{\mathbf{Y}}$ involves application of the same matrix, so it need not be recomputed.

However, such a "one jump" solution may not be optimal, since part of the snake may not have locked onto a target. It is often better to iterate towards the feature (the snake pulls other regions of itself closer to the edge, allowing them to establish lock) — see Figure 3.1. Such a scheme may be formalised by the following recursion relation:

$$\underline{\mathbf{X}}^{m+1} = \underline{\mathbf{X}}^m + \omega(\Delta\underline{\mathbf{X}})^m \tag{3.4}$$

where $0 < \omega \leq 1$ and $(\Delta\underline{\mathbf{X}})^m = \underline{\mathbf{X}}_p - \underline{\mathbf{X}}^m$. The predicted solution, $\underline{\mathbf{X}}_p$, is obtained by evaluating Equation 3.3 with the updated target edge data. The parameter ω determines the rate at which the snake converges towards the solution. We used a value in the range 0.5 to 1.

Rather than solving the equation directly (which requires computation of a large inverse), we employed Gauss-Seidel iteration to approximate the solution to $(\Gamma^\top\Gamma)\underline{\mathbf{X}} = \Gamma^\top \mathbf{v_x}$. Since $\Gamma^\top\Gamma$ is sparse (block circulant), the matrix operations may be implemented very cheaply.

3.2.4 Correspondence Over Slices

In moving from one slice to another, a natural initial position for the snake is the converged position from the previous layer. However, the overall size and location of the contour of interest may change significantly between layers, and our search scale may be too small encompass this displacement. There is then an increased likelihood that the snake will lock onto a spurious boundary. By predicting the position of the snake on the next slice one can provide a reasonable initialisation which reduces the chance of a false lock.

We used a linear recursive filter [2] to update two parameters which controlled the subsequent positioning of the snake on the next slice. These parameters were the *centroid* $(\overline{x}, \overline{y})$ of the contour and the relative *scaling* between layers (measured by the area ratio) about the centroid. The predicted curve position on the next slice was obtained by scaling the snake about the current centroid and then translating it to the predicted centroid location. Once the snake had established lock, the updated parameters were fed back into the update and a new prediction made for the next slice. Initial model parameters were not learned — they will vary significantly depending on the slice from which the segmentation was initialised. Instead, user intervention was required to guide the initial segmentation. Once a sufficient number of slices had been segmented, the prediction was activated and one iteration of the segmentation algorithm was run every time the user stepped to the next slice. If the prediction was sufficiently good, no further intervention was required by the user, since the snake would immediately lock onto the relevant boundary.

3.2.5 Results

Figure 3.2 shows the the final snake positions after a single snake was initialised and allowed to propagate across the set of slices. The measured centroid and scaling as the snake tracks through the sequence, attracted to the right hemisphere, is shown in Figure 3.3. These measurements are for a sequence where no prediction is used, and the final position of the snake is checked and corrected interactively.

Figure 3.4(a) shows the measured mean distance of the target contour measured from the initial position of the snake inherited from the previous layer. The starting position is slice 10. Because the brain is roughly hemispherical, the initial distances are small at the "equatorial" sections, and increase at the highly curved upper region. The measured centroid and scaling functions are used to set the initial parameters of a linear recursive filter. With the predictive filter the measured mean distance is substantially reduced (Figure 3.4(b)). A small search scale can be used and matching between layers is automatic. (Note: the scale on the second figure was halved to show more detail on the graph).

3.3 Surface Construction and Mid-Saggital Plane Estimation

Although the segmentation provides a representation of the 3D object, this representation needs to undergo additional processing before we may properly use it. As we have stated elsewhere, the patient's brain need not be aligned with the imaging axes of the MR machine. In order to produce laterality estimates which are free from foreshortening effects we must compute our measures relative to the 3D MSP. This involves two steps:

1. MSP estimation
2. Surface construction.

Further discussion of these tasks may be found in [50, 40].

3.3.1 MSP Estimation

The MSP is most readily identified over the upper part of the cerebrum, where it is defined as the centre of the (inter-hemispheric) fissure separating the hemispheres (before they fuse lower down). Although the hemispheric divide is not properly planar over its entire extent, a plane serves as a good approximation.

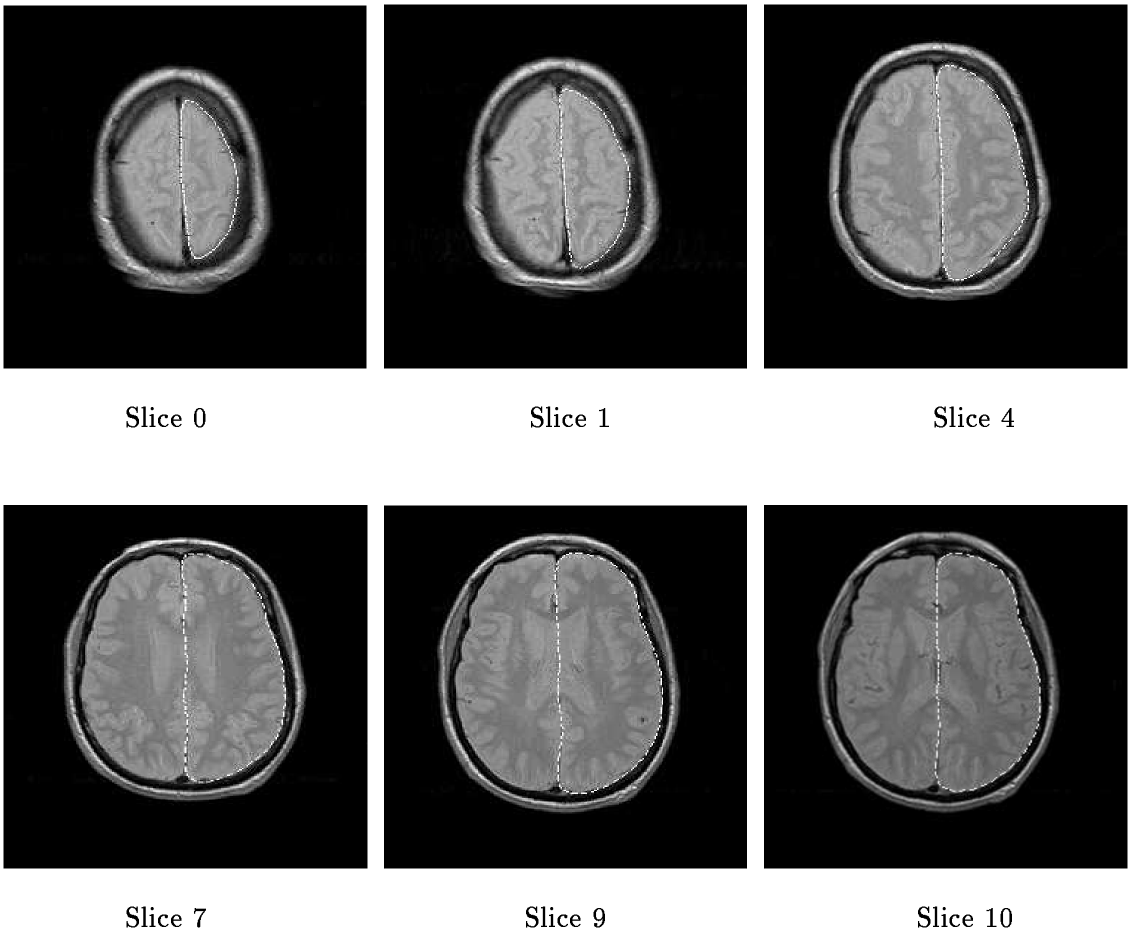


Figure 3.2: Snake segmentation A snake was initialised on slice 10 and allowed to propagate across the upper part of the brain. The final positions of the curves required some minor editing, particularly near the top of the brain where PVE obscures edges.

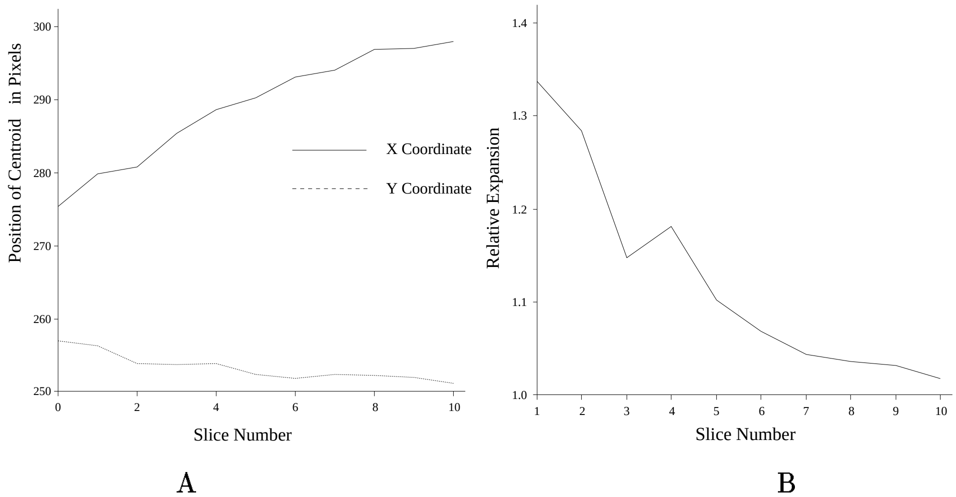
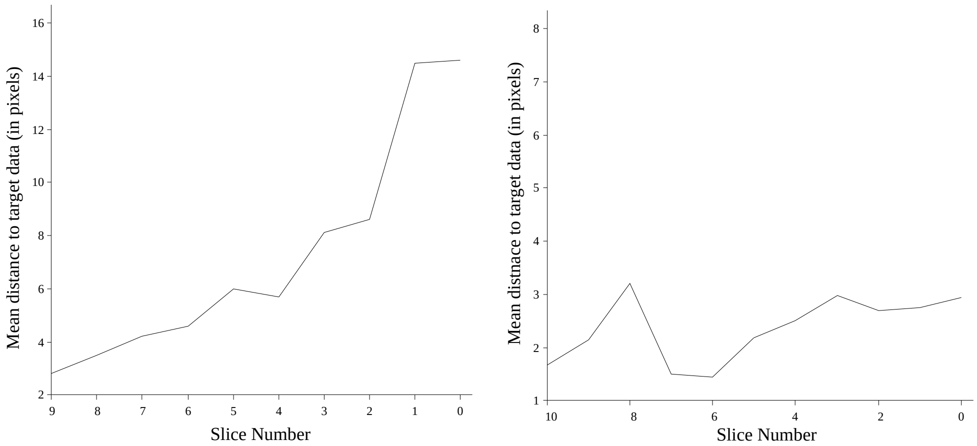


Figure 3.3: Predicted quantities A) Measured snake centroid and B) relative expansion between slices as a function of slice number for the right hemisphere sequence of Figure 3.2. The expansion is about the centroid of the snake. X is horizontal measured from the image left, and Y vertical measured down from the image top. As expected, as the sectioned hemisphere enlarges (increasing slice number), there is a consistent drift of the centroid to the right, and the relative expansion between layers decreases.

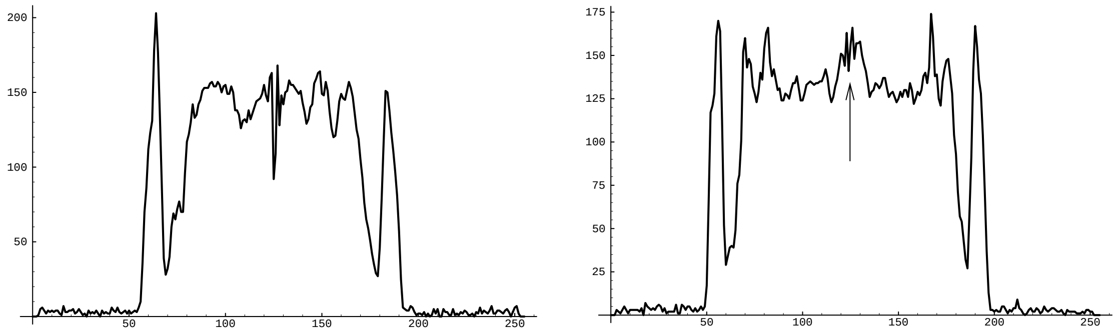


A B

Figure 3.4: Motion prediction The mean distance between the initial position of the snake in each layer and the target contour (the right hemisphere), as a function of slice number for the sequence of Figure 3.2. A) No motion prediction — the initial position of the snake is the converged position of the previous layer. B) With prediction - distances are substantially reduced and consequently a small search scale (7 pixels) can be used.

An open-ended snake curve, defined by a small number of control points, is drawn along the fissure and allowed to propagate across adjacent slices. The snake is designed to look for the feature corresponds to the hemispheric divide. Visually, this feature corresponds to a region which is darker or brighter than tissue to either side i.e. a valley or ridge in the intensity profile perpendicular to the plane.

The scan-weighting (T1/T2 etc) determines the polarity. Unfortunately, there is a large amount of clutter about this line and 1D cross-sections sometimes reveal no trace of its existence — Figure 3.5.



A B

Figure 3.5: Mid-saggital line cross-sections A) The intensity profile through a T1 image. The mid-saggital line corresponds to the trough in the centre of the profile. B) Another cross-section on the same slice — the mid-saggital line is not clearly visible; its position is indicated by the arrow.

We attempted to solve this problem by extracting a small set of additional profiles parallel to the test profile and averaging them, in the hope of boosting the SNR. The feature of interest should be preserved, since it occupies essentially the same location within each neighbouring profile. This was only partially successful, since the line representing the intersection tends to fade in and out. T1 and T2 scans generally provided a more clearly defined mid-saggital line than PD scans.

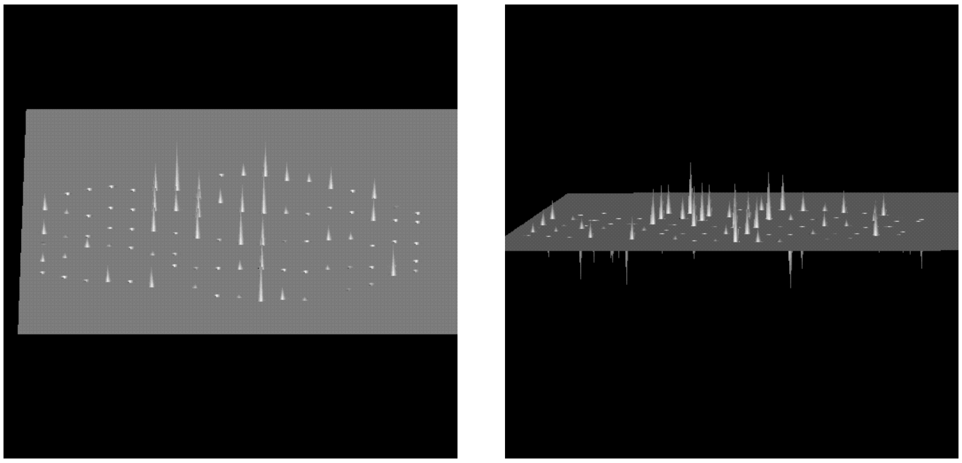


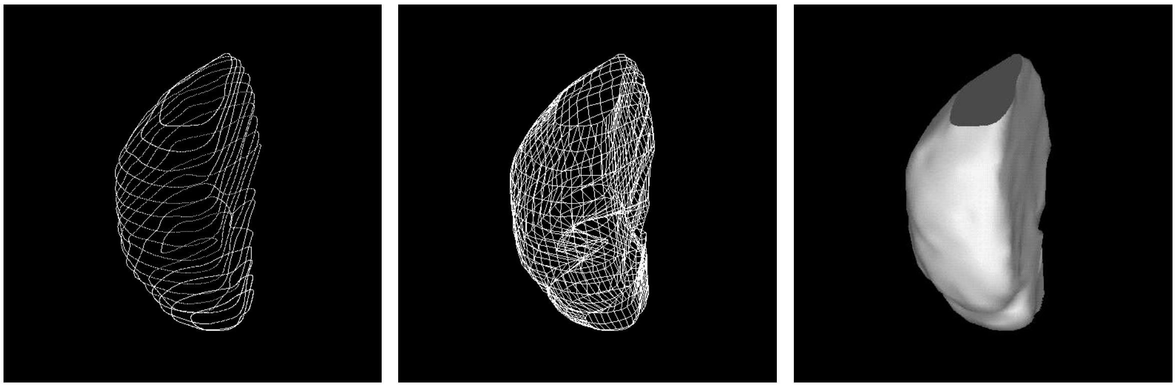
Figure 3.6: Mid-saggital estimation Orthogonal regression is used to fit a plane to the 3D data points. The peaks on the surface represent the distance to points in the data set (the scale has been exaggerated for visualisation purposes). The average error between the plane estimate and the point set is about 1mm.

Because the feature points required for the snake update are sometimes absent, the snake requires manual editing on each slice. The set of snake curves are then sampled to provide a 3D point set, and a plane is estimated using orthogonal regression — Figure 3.6. This may be implemented by computing a principal component analysis of the extracted point set and using the two eigenvectors with the largest eigenvalues to define the plane. The point set mean defines the origin of this plane.

It should be noted that the MSP approximation is based on points near the middle of each slice since the fissure opens up near the periphery of the brain and it becomes more difficult to estimate the requisite points.

3.3.2 Surface Construction

The laterality calculations require the identification of points on the surface of the brain which may be joined by a line perpendicular to the MSP. The contours extracted during the segmentation stage are not adequate for this purpose: we need a surface approximation. The method chosen was approximation via quadrilateral patches. This choice was determined by the nature of our surface data (serial cross-sections) and the decision to avoid generic surface fitting schemes, which we considered unsuitable for such sparse data with its potential topological complexity. By resampling the curves and "connecting the dots" we can generate



A B C

Figure 3.7: Steps in Surface Construction A) The segmentation yields a series of planar snakes which represent the object. B) These are connected to form a surface mesh. C) The mesh may be shaded for visualisation.

a surface composed of quadrilateral patches.

The mesh generation algorithm presented here is only appropriate for the case where there is one snake per slice. Because of this limitation, the snake surface is represented as a series of disjoint open ended surfaces (topologically cylindrical). This decomposition arises naturally from the segmentation process: the addition of a new snake curve signals the creation of a snake "sub-object". As the new curve propagates across slices, it implicitly builds up the desired sub-object. The explicit algorithm to construct the complete surface representation from the set of these "sub-object sequences" is as follows:

Algorithm 3.1 (Surface Construction)

1. For each sub-object sequence
   1.1 Find slice with first curve, C_0, in the sequence
   1.2 Compute any plane, P, which passes through the control point centroid and has its normal lying within the image plane.
   1.3 Select the closest intersection point of P with C_0.
   1.4 Resample the curve from this point to produce a set of V_n evenly spaced vertices.
   1.5 For each remaining snake $C_N, N \geq 1$, in the sequence:
      1.5.1 Compute the intersection of P with the curve.
      1.5.2 Resample the curve from this point to produce a set of V_n evenly spaced vertices.
      1.5.3 Join each vertex to its correspondent on C_{N-1} to form a quadrilateral representation.
2. Connect the first and last sample point on each curve to close the quadrilateral mesh.

The various stages in the production of the surface are shown in Figure 3.7. The number of vertices, V_n, was usually set at 100 — the intrinsically coarse nature of sparse data did not necessitate a dense mesh representation.

The procedure used to determine an origin for the reparameterisation produces quadrilaterals which may be sheared in an alternating pattern across the surface. This is due to the difference in length between adjacent curves and the choice of a common origin: correspondents on each curve match well near this point, but can have very different relative spacing as one moved away from the origin. When two pairs of these rotated correspondents are connected a sheared quadrilateral results. In our work, the plane always intersected the sub-object curves. If the curves had large relative translations, this method would have been inappropriate.

Perhaps the most serious objection to this scheme is the production of a disjoint surface representation. This is a consequence of topological transformations within the set of propagating snakes. An alternative scheme, based on an "active surface" is suggested in Section 3.4.2.

3.4 Discussion

The methodology outlined was an instructive first attempt; it revealed several flaws in our thinking but also provided some important ideas for our subsequent work. This section addresses the problems we encountered.

3.4.1 Edge Model Robustness

The mechanism for extracting edge-points was based on a contrast sensitive edge detector. Our "model" of the grey-matter/CSF boundary was simply contrast sensitive detection of neighbouring pixel intensity differences. For slices in which PVE was minimal, this approach was quite successful at extracting the desired boundary in most cases. Unfortunately, the increasing effect of PVE and the lack of neighbourhood information meant that we obtained poor results as we neared the poles of the brain.

The extraction of the MSP proved to be quite problematic. The simple valley/ridge detector was unable to reliably extract the desired points — in many instances the line was simply not visible. If we averaged a series of neighbouring profiles (boosting SNR), we improved matters somewhat, but there were still many cases in which the scheme failed. The model assumes that the background (the region of the profile outside the fissure/MSP region) is essentially flat. Unfortunately this is most definitely not the case: there may be large undulations in intensity caused by transitions from white to grey matter or CSF. In many cases, these events are associated with large contrast changes, and may form localised valleys/ridges of their own when viewed on a large scale. In fact, the symmetry of the MSP detection (the model looks to find a central ridge/valley) is not only satisfied for the MSP, but also for any small enough window about a secondary valley/ridge. In many cases, these features are present in several of the neighbouring profiles as well, which means that profile averaging may actually accentuate them. Based on these observations, we must conclude that a localised feature search is completely inappropriate for the detection of the MSP. Furthermore, the application of a symmetry constraint is problematic within 2D images because of patient-machine alignment artifacts: the widths of the hemispheres may be distorted by length foreshortening.

3.4.2 Coping with Topology and Geometry

A fundamental limitation of the above scheme is its inability to deal with topological transformations of the snake contours. Attempts could be made to predict how and where such transitions are likely to occur, but given the sparsity of the data this seems implausible. Although we know the topology of the object beforehand (from anatomical knowledge) we are unable to easily mobilize this knowledge in the contour-based segmentation process. For example, we know that the topology of a brain hemisphere is (at the scale we are interested in) spherical. Of course, simple topology may be associated with fiendish geometric complexity — as the brain illustrates.

The algorithm used to produce a surface from the set of snake curves is crude and places very specific constraints on the type of surface we can construct. Part of the reason for these limitations arises from our inability to interpolate the missing data: serial sections of a 3D object do not provide a unique representation of that object. This problem has been studied elsewhere (for example [69]), with some success. However, one has no way of knowing whether the resulting interpolation will be physiologically meaningful, which is of paramount importance to us. Furthermore, since our data sets are truncated, the resulting surface representation will be artificially distorted.

If we were to utilise a 3D surface for our segmentation we would no longer be required to implement a separate "fitting" step. Better yet, the surface can be made to possess the correctly topology at the outset, thus overcoming the problems associated with evolving contour topology. Of course, data truncation would still distort our shape representation, unless we were able to interpolate the missing surface regions. Such an interpolation would only be meaningful if it produced a result consistent with brain anatomy. This requires the inclusion of an anatomical shape constraint in the segmentation process. The use of implicit shape constraints (such as those associated with a B-spline or hyperquadric) would be insufficient, since these would not necessarily reflect the true underlying physiology.

3.5 Conclusion

The use of a 2D snake-based segmentation scheme for sparse data was investigated. The segmentation was based upon B-spline snakes which provide a good framework for the inclusion of constraints; these constraints are intended to improve robustness by including *a priori* knowledge of our problem.

During the course of our investigation it became clear that such a 2D segmentation scheme was fundamentally limited. While there is certainly room for improvement, particularly in the boundary detection stage, such modifications would ultimately fall short of our desire to quantify complex 3D geometric structure. These observations prompted our research into a 3D model-based methodology. This new approach incorporates aspects of the old (line-based, region-specific boundary searches etc) but is based upon a discrete 3D model known as a Simplex Mesh. The bulk of this thesis concerns the development of this new framework and its validation.

Chapter 4

Constructing a Shape Template

The selection of an appropriate shape representation is strongly influenced by the structures it is supposed to represent, as well as its intended purpose. For our work, the representation must be sufficiently general to encompass complex geometry and topology, and flexible enough to serve as a "shape template". This template will enable us to seek out similar shape instances within a volumetric image, despite the presence of noise, imaging artifacts and significant shape variability.

Following from the review and discussion presented in Chapter 2, we have rejected parametric models, with their limited shape resolution, in favour of a dense, discrete mesh representation. Although more powerful continuous parametric descriptions (such as Fourier surfaces) are available, they are subject to certain fundamental limitations, such as the inability to represent complex topologies. A model capable of representing objects with several "holes" is essential for the segmentation of structures such as the ventricular system. Although the primary focus of this work is the segmentation and description of objects with spherical topology (cerebral hemispheres), there is sufficient interest in structures such as the ventricular system to motivate the development of a more generic segmentation scheme.

As the model complexity increases, the number of parameters required to represent it may become excessive. This is particularly true for discrete surfaces, in which each vertex represents 3 additional degrees of freedom. Fortunately, there are methods which can allow one to reduce the dimensionality of the model. More specifically, by examining a set of training examples (segmented surfaces in our case), it should be possible to construct a representative model of lower dimensionality by minimising the intra-object redundancy. The reduction process we utilise — called Principal Components Analysis (PCA) — also provides a means of constraining the permissible shapes the model may assume. This, in turn, stabilises the segmentation process and provides a framework for the classification of pathological structures.

The discrete surface representation we have chosen is the *Simplex Mesh* [30]. Within this framework, it is possible to represent arbitrary topologies, transform from one topology to another and refine the mesh vertex density adaptively.

We begin this chapter with a brief description of the simplex mesh and its properties. The original mesh segmentation strategy is discussed in Section B.3, followed by a review of possible shape template schemes and an overview of our proposed methodology in Section 4.2. We then examine the issues which must be addressed if this approach is to be executed successfully — Section 4.3 and Section 4.5. After presenting the results of our investigations in Section 4.6, we discuss several issues associated with our particular template scheme — Section 4.7 — before concluding the chapter.

4.1 The Simplex Mesh

The simplex mesh [30] is a discrete surface representation in which each vertex is connected to 3 others. Each vertex is associated with a local curvature-like measure called the "simplex angle", and may be expressed in terms of its 3 neighbouring vertices. The mesh formalism contains a number of strategies to refine and update the surface, including a mechanism to enforce "shape constraints". The reader is referred to Appendix B for additional details.

In its original conception, the simplex mesh was intended as a 3D shape description rather than an active surface for general segmentation. In particular, it was assumed that object boundaries could be easily and unambiguously identified. Such an assumption was justified for the 3D range data and simple volumetric images used to represent the boundaries of objects. When one deals with complex medical images, in which the boundary may be difficult to isolate, problems emerge. The use of 3D gradient and/or edge (thresholded gradient) images allows the detection of obvious boundaries but fails to locate those that are hidden by surrounding gradient maxima. Furthermore, the gradient operator is computed in 3D, which may yield poor results for sparse volumetric data.

The segmentation process also requires a high level of interaction to guide the surface towards its target. This usually involves the modification of mesh parameters and continuity constraints, prior to the intialisation of a surface fitting step. The best surface fit of the available data would be followed by another change of parameters and surface update step. Repetition of this process eventually leads to a segmentation (object description) of the object from the 3D data set.

In order to overcome some of these problems, one may utilise a reference template [52], defined through a "shape constraint" — Appendix B.1. Unfortunately, such a model might not adequately represent the variability which exists within the given class of shapes. This objection is particularly relevant for biological structures. Although the model will constrain the mesh to a "reasonable" shape this may not be entirely desirable since we have no way of knowing whether

1. the data which is being detected by our feature search is "good" (i.e. the actual boundary we want)
2. whether the template provides an appropriate representation for this instance of the shape.

These objections seem to be somewhat justified by the results presented in [52], in which a simplex template of the liver (generated through an undisclosed process) was used to segment that organ from abdominal MRI data. There are significant discrepancies between the model and the actual data. In certain instances it is not clear whether the model has provided a reasonable approximation to the organ boundary.

4.2 Learning Shape Variation

The use of a shape template [8, 24, 52] for segmentation provides additional robustness when we have truncated or noisy data. Missing or noisy boundaries may be interpolated by the model to provide a compromise solution which should reflect the true nature of the underlying shape. If we have a single shape this may be used to define our template, since we have no interest in representing any other shape instances (these would be considered spurious). Alternatively, we may allow the template to deform in a specified manner so as to capture some element of the shape instance which we know *a priori*.

We require a shape model which can represent a class of objects with high shape variability. This precludes the use of a shape constraint such as that presented in [52], which assumes that one shape is a suitable representative of an entire class. Furthermore, such an approach does not place bounds on the permissible shape variation — this facility is useful for classifying putative abnormalities (out of class variation)

One mechanism for the encapsulation of shape variability may be found in Principal Components Analysis. Within the PCA framework, the variability across a set of vectors (which might represent, for example, vectors of shape descriptors), is made explicit by the computation of a set of "modes of variation". These modes provide a reduced set of basis vectors on which similar data may be approximated. The PCA process builds this description by examining a representative "training set" of vectors — see Section 4.2.1 for details. Furthermore, by noting the extent to which each mode contributes for a given data instance, it is possible to reject data which are not in good correspondence with the learned variability.

For shape processing, the PCA is performed on sets of vectors which provide a description of the shapes in the training set. If the shapes are represented as Fourier Surfaces, for example, the coordinates of each shape on the Fourier basis could be used. In general, however, the shape data is only available through discrete digital sampling and a process has to be devised to convert this data into a form suitable for analysis. While it may be possible to fit the discrete mesh data with a continuous parametric form such as a Fourier Surface or superquadric, this may fail to give a satisfactory result since,

1. the parametric description may be dependent on the spatial orientation of the data, and
2. the minimisation algorithm utilised by the fitting procedure may be trapped by local minima, and
3. the associated surface parameterisation may be inconsistent across the training set, and
4. the parametric form may be limited in topology it can represent. In most cases, only surfaces with spherical or toroidal topology can be described.

The consistency referred to above concerns the extent to which the surface parameterisation identifies "the same point" across the training set. This is important for us, since we wish to gather information about surface structure at corresponding points on each mesh. This topic will be discussed in Chapter 5.

Rather than attempting to resolve these difficulties, we have devised a shape modelling process based on the direct computation of correspondents. Our definition of "correspondent" is as follows: Given two surfaces from a particular class of shapes, which have been aligned as best as possible, a given point on one surface is considered to be the correspondent of the point on the other surface which is closest to it, under some sensible distance metric. The nature of the distance metric, and the manner in which surface alignment can be achieved are addressed below.

If we are able to associate a point on a given mesh surface with its counterpart on each of the remaining surfaces, we will have effectively parameterised the training set. This "implicit parameterisation" will not provide a means of deducing additional corresponding pairs[1], since there is no underlying continuous surface parameterisation. However, it provides sufficient infrastructure to support the construction of a shape template.

We propose the following approach for the construction of our model:

Registration The registration step will enable us to identify the surfaces as closely as possible (within the chosen class of transformations). This is a necessary precursor to cor-

[1] Unless the surface is very smooth, in which case we can use interpolation to estimate additional points.

respondent extraction and serves to eliminate "uninteresting" inter-patient differences, such as location of the patient in 3D space. We shall consider 4 classes of registration transformations, drawn from methods based on 1) a bounding box description of each shape, and 2) the Iterated Closest Point (ICP) algorithm.

Correspondent Extraction By applying our correspondent selection scheme to the registered mesh surfaces, we aim to produce an implicit reparameterisation of each surface. That is, given a set of points on any one surface, we wish to be able to identify their counterparts on the remaining surfaces. It should be noted that unlike an explicit surface parameterisation, in which there is the notion of an underlying continuous (usually orthogonal) parametric system, this scheme provides only a discrete set of point correspondents. Of course, one may choose as many points as desired.

Construction of PDM The resampled surface mesh representation, may be viewed as a vector of 3D spatial coordinates. The Point Distribution Model [24] provides a framework for the construction of a discrete shape template from a set of such corresponding shape vectors.

These issues are now addressed in some detail. We begin with an overview of the theory necessary to understand the Point Distribution Model.

4.2.1 Principal Components Analysis

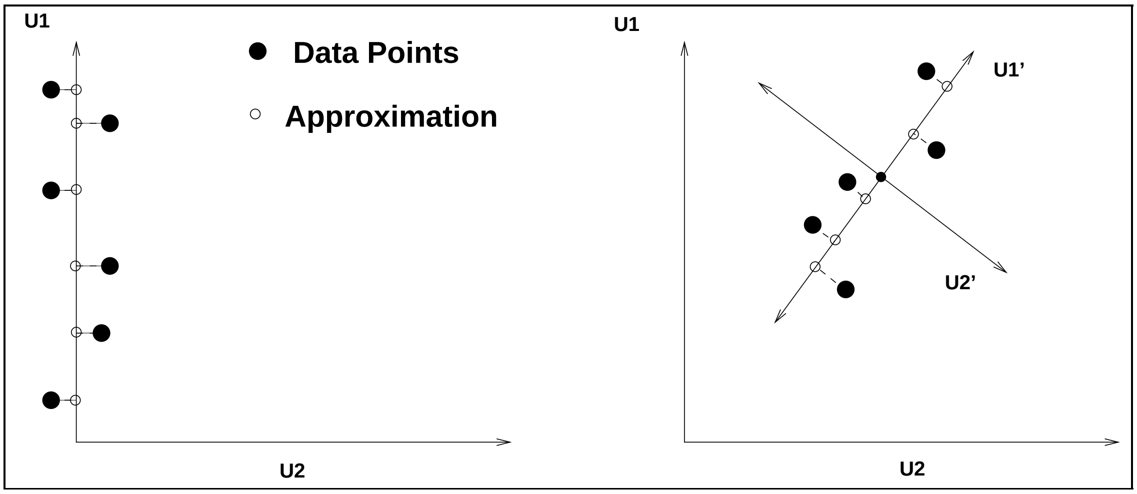
Given a set $\{\mathbf{x}_i\}_{i=1}^{M}$ of N dimensional points, we seek a means of exhibiting structure within the data set. This is useful for analysis and, in certain cases, may allow us to approximate the elements of the point set using less than N parameters. Any point, $\mathbf{x} \in \mathbb{R}^N$, may be expressed within an orthonormal basis $\{\mathbf{u}_j\}$

$$\mathbf{x} = \sum_{j=1}^{N} \alpha_j \mathbf{u}_j, \tag{4.1}$$

where $\alpha_i = < \mathbf{x}, \mathbf{u}_i >$. If the point under consideration lies on or near a coordinate axis, its projection onto each of the remaining axes will be negligible. Thus, by examining the basis coefficients, it is possible to establish how well any point is approximated by its projection onto a given coordinate vector, say $\mathbf{u}_p$. If the set of points $\{\mathbf{x}_i\}_{i=1}^{M}$ is clustered along $\mathbf{u}_p$, then the total contribution of each point's projection onto the given axis will far outweigh their projection contribution on the remaining axes. Consequently, we could approximate each point $\mathbf{x}_i$ using only the pth basis vector — Figure 4.1A:

$$\hat{\mathbf{x}}_i = < \mathbf{x_i}, \mathbf{u}_p > \mathbf{u}_p. \tag{4.2}$$

It may be that the projection of the point onto the remaining axes is not zero; in this case we could improve our approximation by adding on a term for additional basis vectors. If we



A B

Figure 4.1: Principal Components For the point clustering shown in A), each point may be approximated using only 1 parameter, assuming U1 and U2 are known. This parameter is the projection of the point onto the "principal axis", U1 in this case. B) For an arbitrary arrangement of points, the PCA process determines the optimal basis which may be used to approximate the point set. If the points are clustered as shown, then U1' is the principal axis. U1' and U2' constitute a orthogonal basis.

associate a cost function

$$C_k = \sum_{i=1}^{M} \| < \mathbf{x}_i, \mathbf{u}_k > \|^2 \tag{4.3}$$

with the kth coordinate vector, we may refine our approximation to the point set by systematically adding the basis vector with the highest score, since this reflects, on average, the next most important contributing term in our orthogonal expansion.

We observe two things:

1. We may successively improve our approximation to the point set by adding basis vectors according to our cost function, and

2. if we only use L basis vectors, where $L < N$, we may represent each point using only L parameters (assuming that the basis vectors are known). In other words we have reduced the dimensionality of the point set from N to L!

Unfortunately, it is unlikely that the point set will be so conveniently arranged. If we blindly apply this algorithm, we may obtain a very bad estimate unless we retain all the coordinate vectors, thus gaining nothing. This does not mean that there is no redundancy in the data, just that it is not obviously exhibited. What we really wish to find is a set of orthogonal unit vectors in our N dimensional space which are such that our cost function C_k is as small as it can be for each axis.

Principal components analysis provides a way of computing these vectors — Figure 4.1B. The formulae quoted here, along with a more extensive discussion, may be found in [7]. For reasons of economy we shall present only a brief description.

We may express any point, $\mathbf{x}_i$, within our N dimensional space on an arbitrary orthonormal basis, $\{\mathbf{u}_j\}$, as

$$\mathbf{x}_i = \sum_{j=1}^{N} \alpha_{i,j}\mathbf{u}_j. \tag{4.4}$$

To facilitate the derivation below, each point may be alternately represented by $\hat{\mathbf{x}}_i$

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{L} \alpha_{i,j}\mathbf{u}_j + \sum_{j=L+1}^{N} \beta_j\mathbf{u}_j. \tag{4.5}$$

Of course the two representations are initially equivalent. The $\{\beta_i\}$ are the basis coefficients for terms we wish to discard, and are taken to be constant i.e. $\alpha_{i,j} = \alpha_{k,j} = \beta_j,\ L < j \leq N$. We must determine the basis which minimises the residual

$$E_L = \frac{1}{2}\sum_{n=1}^{M} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2 = \frac{1}{2}\sum_{n=1}^{M}\sum_{j=L+1}^{N} (\alpha_{n,j} - \beta_j)^2. \tag{4.6}$$

Minimising w.r.t. β_j we obtain

$$\beta_j = \frac{1}{M}\sum_{n=1}^{M} \alpha_{n,j}. \tag{4.7}$$

If we define the mean $\overline{\mathbf{x}}$ of the point set as

$$\overline{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i, \tag{4.8}$$

we may rewrite the residual as

$$\begin{aligned} E_L &= \frac{1}{2}\sum_{n=1}^{M}\sum_{j=1+1}^{N} \{\mathbf{u}_j^T(\mathbf{x}_n - \overline{\mathbf{x}})\}^2 \\ &= \frac{1}{2}\sum_{j=L+1}^{N} \mathbf{u}_j^T \boldsymbol{\Sigma} \mathbf{u}_j, \end{aligned} \tag{4.9}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the set of vectors, given by

$$\boldsymbol{\Sigma} = \sum_n (\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^T. \tag{4.10}$$

One may show (see, for example, [7]) that the minimum of Equation 4.2.1 occurs when

$$\boldsymbol{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i, \tag{4.11}$$

where the $\mathbf{u}_i$ and λ_i are the eigenvectors and eigenvalues of the covariance matrix.

The residual reduces to $E_L = \frac{1}{2}\sum_{j=L+1}^{N} \lambda_j$. By discarding the (basis) eigenvectors associated with the smallest $N - L$ eigenvalues, we obtain the optimum description for a specified dimensionality.

4.2.2 The Point Distribution Model

A Point Distribution Model [24] is a discrete representation of shape which is generated from a PCA. We may represent a shape in a N-dimensional space by a discrete sampling at M points along its boundary. The manner in which those points are selected will be addressed presently, but for the moment we will assume that the shapes are in the space $N = 2$ i.e. a 2D contour. Such a sampling will yield a NM dimensional vector, which we may consider to be a point in a space of the same dimension. We wish to examine a set of such discretely sampled shapes, with a view to describing their variability and (hopefully) reducing the dimensionality of the parameter space.

As we did in the preceding section, we wish to find a new basis in our shape space which provides the principal components of our training set. This may be achieved by performing a PCA on the set of NM dimensional points. By discarding the p basis vectors with the smallest eigenvalues, any model, $\mathbf{X}_k$ may approximated as

$$\hat{\mathbf{X}}_k = \overline{\mathbf{X}} + \sum_{i=i}^{N-p} \alpha_{k,i} \mathbf{U}_i, \tag{4.12}$$

where $\{\mathbf{U}_i\}$ are the NM dimensional basis vectors we have retained and $\alpha_{k,i} =< \mathbf{X}_k, \mathbf{U}_i >$. The mean has to added to the expansion because the data is mean centred.

The PCA process produces a set of eigenvectors, arranged in order of decreasing eigenvalue. The variability explained by a given basis vector is proportional to the corresponding eigenvalue. In fact

$$\alpha_{k,i} = \lambda_i = \sigma_i^2. \tag{4.13}$$

We wish to retain the smallest set of basis vectors which describes a given proportion of the variance observed in our training set. To this end, we may require that the sum of variances associated with the retained modes explain a specified proportion of the observed variability, say 95%:

$$\lambda_T = \frac{\sum_{n=1}^{d} \lambda_n}{\sum_{n=1}^{N} \lambda_n} > 0.95, \tag{4.14}$$

We would then simply find the maximum value of d for which this sum is below our threshold and retain the first d modes. We may also generate shapes which are similar to those in our training set by perturbing the $\alpha_{k,i}$ (typically with the range $\pm 3\sqrt{\sigma_k^2} \equiv 3\sqrt{\lambda_k}$; i.e. 3 standard deviations from the mean).

In this work, the discrete shape models are embedded in a space of dimension $N = 3$ and a large number of 3D points on the mesh surface are chosen to represent each model. This means that our shape space has a ridiculously high dimension. Fortunately, there is a vast amount of redundancy in this representation and the PDM provides a means of encapsulating this variability in a compact manner.

An important issue — from an implementational point of view — is the computational cost required to solve the eigensystem. The covariance matrix Σ is of dimension $NM \times NM$. Since the complexity of such algorithms is typically $O(N^3)$, the construction of a PDM is likely to prove rather time consuming. Fortunately, inspection of the matrices involved shows that we may solve the eigensystem using a two-stage process in which the dimension of the transformed covariance matrix is $N_t \times N_t$, where N_t is the number of training examples. Details are provided in Appendix C.3. Since $N_t << NM$ for our work, this reduces the basic complexity of the problem dramatically.

4.3 Generation of the Mesh Training Set

The construction of the template requires a set of mesh surfaces which represent the boundary of interest. The precise manner in which these surfaces are obtained does not impact on the training process — provided they are representative of the desired structures. To ensure that this goal is achievable, high-resolution data should be utilised in the segmentation process. For the experiments presented in this chapter, the underlying MRI data consisted of T2 coronal scans with contiguous slices and voxel dimensions of $0.9375 \times 0.9375 \times 1.5\text{mm}^3$. The cerebral hemispheres of 20 controls (of both sexes) were segmented, as described below. The MR data set used to build the templates is described in Section 7.5.1.

The segmentation was based on the technique discussed in Chapter 3. Each hemisphere was delineated using a spline snake and the curves were then propagated across each slice until the entire object had been processed. A fair degree of user intervention was required to rectify misidentified boundary structures. The snakes were then densely sampled — Figure 4.2A — to provide a set of points which could be used as input to the default simplex mesh segmentation process, referred to in Section 4.1.

A simplex mesh with a large number of vertices — 4500 in this case — was manipulated to fit the sampled contour data. This required interactive placement of the initial spherical mesh and adjustment of parameters controlling aspects such as surface rigidity and data point attraction. The simplex refinement process was also enabled, allowing more vertices to concentrate in regions of high curvature. Details of this process may be found in [30]. The end result of this fitting process, was a series of simplex mesh surfaces representing the desired anatomical boundary in each control subject — Figure 4.2B. An alternative to this approach becomes feasible if the data has been segmented by other means, and can be represented by a triangulation. In this case, the simplex mesh may be simply constructed

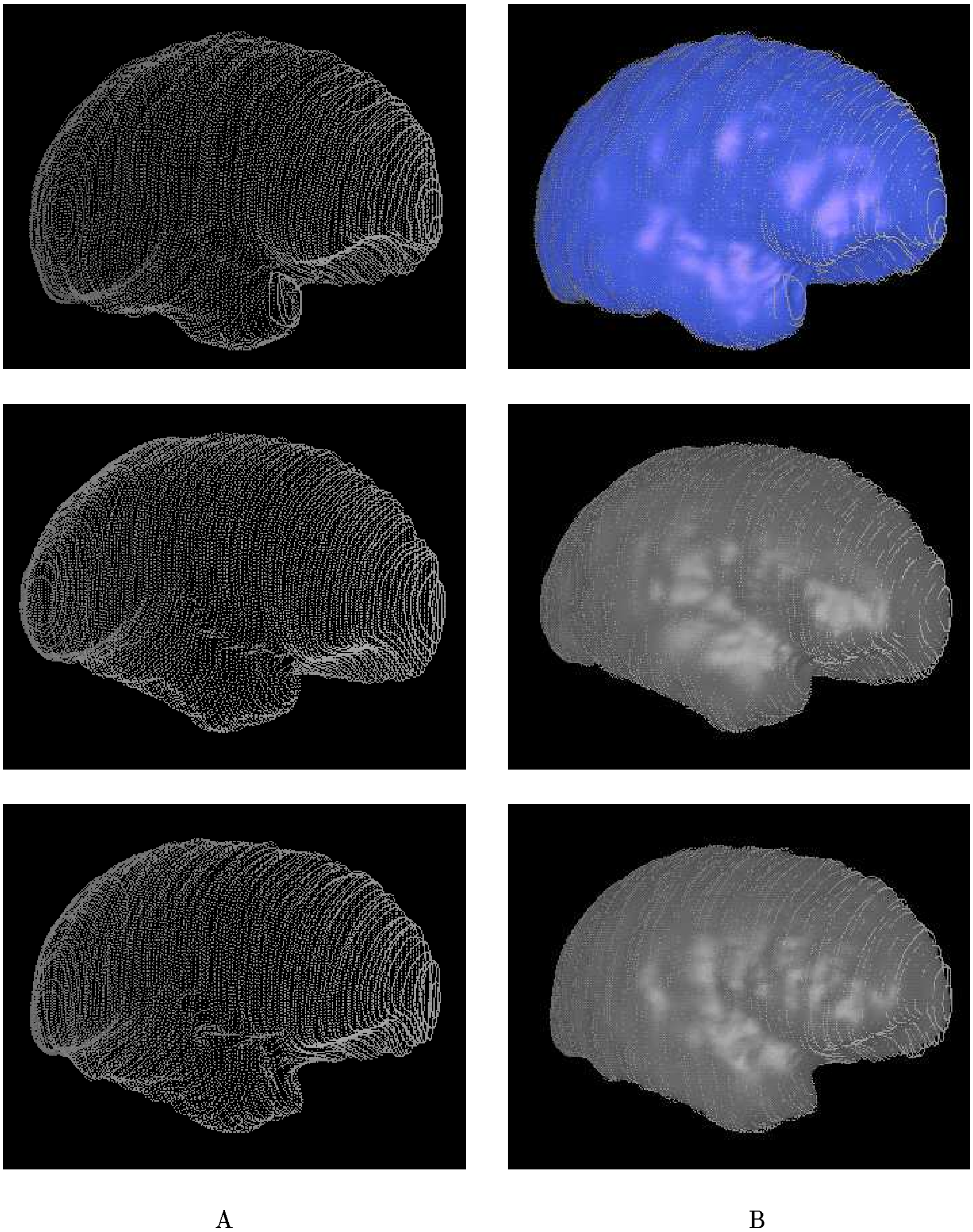


Figure 4.2: Snake Segmentation. A) The full snake segmentation of 3 dense coronal scans. The curves have been resampled to provide a point-set for the simplex mesh segmentation. B) The simplex surfaces generated from the snake data.

from the triangulation (recall that they are dual representations). Unfortunately, in this instance, pre-segmented data was not available and existing tools had to be used.

The number of vertices in the simplex mesh was chosen to reflect the nature of the data we wish to represent. On the one hand, we require a sufficiently large number of vertices to represent the necessary structure, while on the other, it seems sensible to reduce the complexity of the mesh where possible, since it will ultimately serve as a template for extracting a smoothed approximation to the cerebral surface — which is the best we can hope to do in the face of data sparsity. Furthermore, since the computational complexity of the active surface segmentation scheme scales with the number of vertices, it would be unwise to select an excessively high number. This choice was found to be satisfactory for our requirements —Figure 4.3.

To ensure the correctness of the segmentations, the snake delineation process was discussed with neuro-anatomists and a consistent procedure developed for the snake segmentation. The series of images in Figure 4.3 illustrates a number of slices in which these decisions are reflected.

Since the hemispheres fuse at the corpus callosum, an artificial line had to be extracted in order to separate the two. The plane of symmetry which cuts the corpus callosum was used to define this line — Figure 4.3C,F. When the ventricular spaces were encountered by the snake, the curve was forced to trace this boundary — Figure 4.3C. However, the lateral ventricles were not traced, but bisected by the extension of the mid-line used to sever the corpus callosum — Figure 4.3D,E.

Those parts of the ventricular system which did not intersect the propagating curves were ignored — Figure 4.3B,C. When delineating the temporal lobe, the deepest parts of the sylvian fissure were ignored. Because the data to be segmented is sparse, a decision was made to treat the *arachnoid* as the brain surface: this "cling wrap" membrane does not trace deep sulci and therefore our treatment of the sylvian fissure and other complex sulcal penetrations was consistent. The brain stem was excluded, as was the cerebellum — Figure 4.3E,F,G.

4.4 Establishing Point Correspondences

In order to construct the PDM, we need to establish corresponding surface points across our training set. Establishing such relationships is a difficult task: there is no general algorithm to parameterise an arbitrary surface, although it is now possible to parameterise specific kinds of surfaces with spherical topology [13]. However, even with advent of such methods, additional information is required to pin down the coordinate origin. Furthermore, there is no guarantee that points which have the same parametric description across patients are

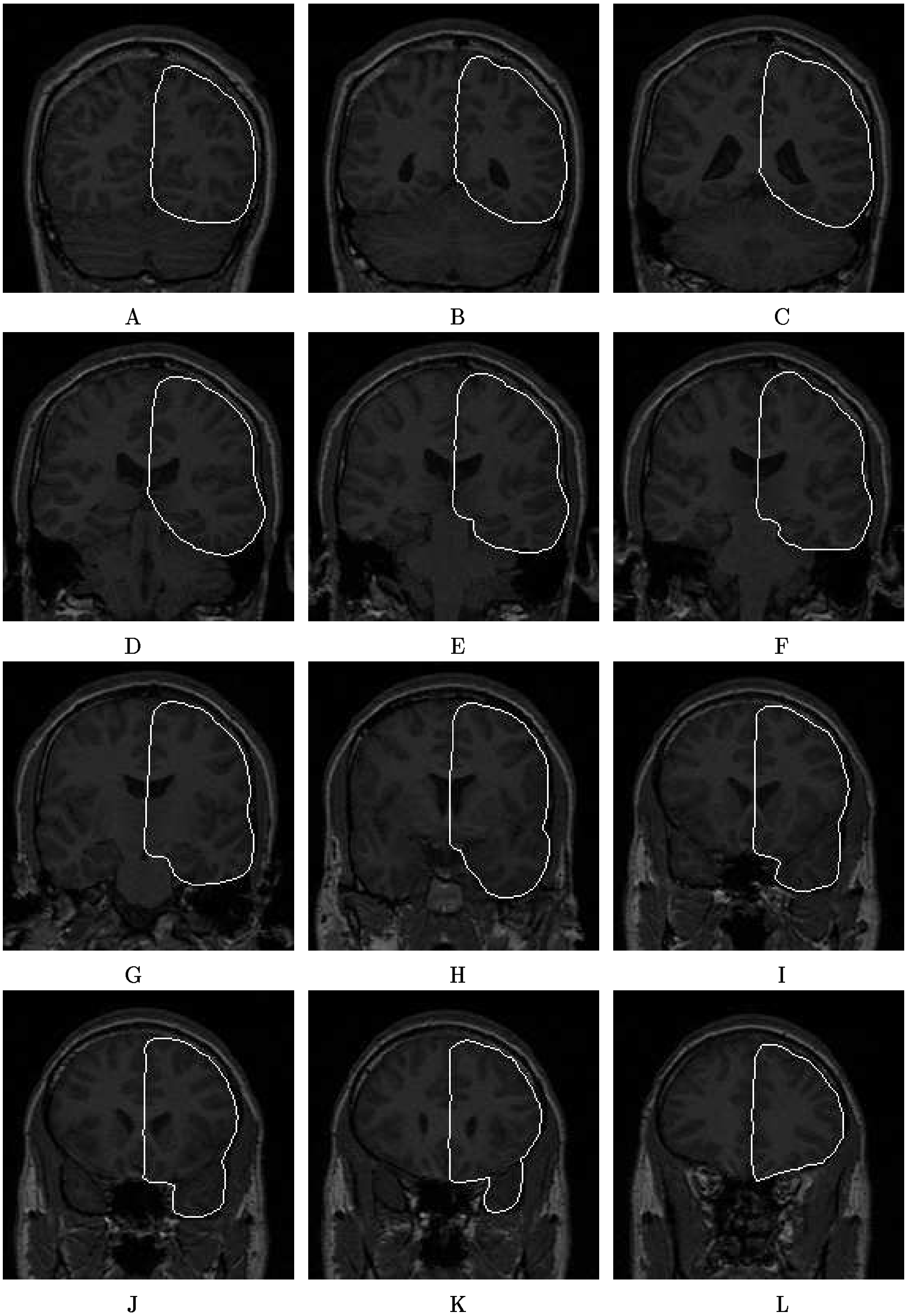


Figure 4.3: Mesh Surface Representation A number of coronal sections are shown, from the back to the front of the head, which illustrate some of the segmentation decisions referred to in Section 4.3. The curves shown here represent the intersection of the image plane with the simplex mesh resulting from the fitting process.

truly the ones we desire.

Rather than attempting to resolve the various problems associated with the determination of a suitable mesh parameterisation, it seems sensible to adopt an approach which is sufficient for our requirements. Our primary goal is to establish surface correspondents for both structural and local grey-scale analysis (addressed in the next chapter). If one makes certain assumptions about the geometric variability across patients, the desired correspondences may be computed without the need for an explicit parameterisation.

More specifically, the following assumptions are made:

- Brains are structurally similar across (at least) the same sex.
- Registration of the data set using global (affine) transformation will leave significant inter-patient differences intact.
- We can compute the mid-saggital plane (MSP).
- We can compute the intersection of a ray with the simplex mesh: in practice this means computing the intersection with a triangular mesh generated from the simplex mesh (not necessarily the simplex dual, since this loses vertex segmentation information).
- We have *complete* brain data i.e. no truncations. This is important if we are to construct a consistent and physiologically meaningful model.

The choice of the registration transformation will have an important impact on the correspondent selection scheme. The core idea is simple: if the transformation brings the meshes into alignment, then choosing the *closest* point on the target mesh to a given point on the reference mesh should provide a satisfactory correspondent. Alternatively, the closest point along the surface normal may be selected. Both of these methods were implemented — the results are discussed in Section 4.5.

4.4.1 Correspondent Extraction

The registration step selects an arbitrary "reference" mesh from the training set and attempts to register each remaining mesh to this surface. Once this has been accomplished a correspondent must be established for each point of interest on the reference mesh. In the interests of simplicity, only mesh vertices were considered. For a specified vertex on the reference mesh, the correspondent is obtained using:

Normal Sampling A ray from the reference vertex, oriented along the surface normal, is intersected with the registered target mesh, or,

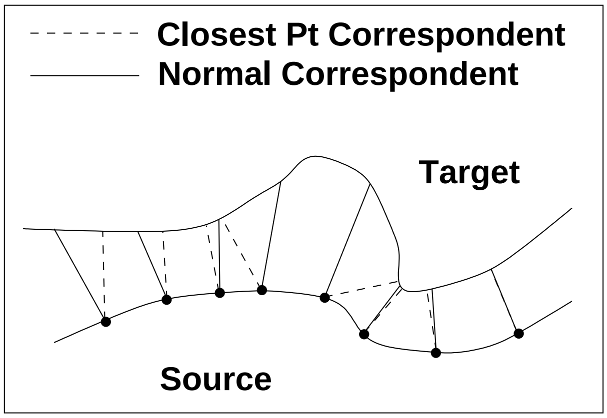


Figure 4.4: Correspondent Selection The correspondents selected by each scheme are shown. Observe that, in general, the selected points may be quite different, unless the surfaces are parallel. In this case the same correspondents will be selected under both schemes. The order of the extraction is also important: if the Source and Target surfaces are exchanged, a very different point distribution may arise, unless they are parallel.

Closest Point Sampling For a given reference vertex, the closest mesh point on the registered target mesh is selected, without regard for geometry.

Figure 4.4 shows how the choice of selection scheme affects the returned correspondent set. For parallel surfaces (or regions) the two schemes return identical point sets. However, if the surfaces are very different, there may be a significant difference between the two schemes. Furthermore, if non-parallel surfaces are exchanged, the correspondents may be very different. These issues will be examined more fully in Section 4.7.

The target mesh is triangulated to provide a surface composed of planar facets — this is essential for the closest point and ray intersection calculations. The closest point on this triangulated surface may be found applying the following algorithm:

Algorithm 4.1 (Closest Point on Triangulation)

1. For each vertex, V, on the reference mesh
   - 1.1 For each triangle, T_i, on the target mesh
     - 1.1.1 Project the V onto the plane spanned by T_i
     - 1.1.2 If the point, P_i, lies within this triangle, $\mathrm{CLPT} = \mathrm{P_i}$ and we are done; else continue.
     - 1.1.3 Compute the closest point, P_i, on the triangle boundary.
   - 1.2 $\mathrm{CLPT} = \mathrm{argmin}_{\mathrm{P_j}} ||\mathrm{V} - \mathrm{P_j}||$.

This calculation may be accelerated by using a spatial indexing scheme to locate a small set of feasible triangles, and constraining the required calculations to this subset. For surfaces with extremely complex geometry, such schemes may fail. Fortunately, for the (arachnoid) surfaces we have considered, such an indexing scheme could be usefully employed.

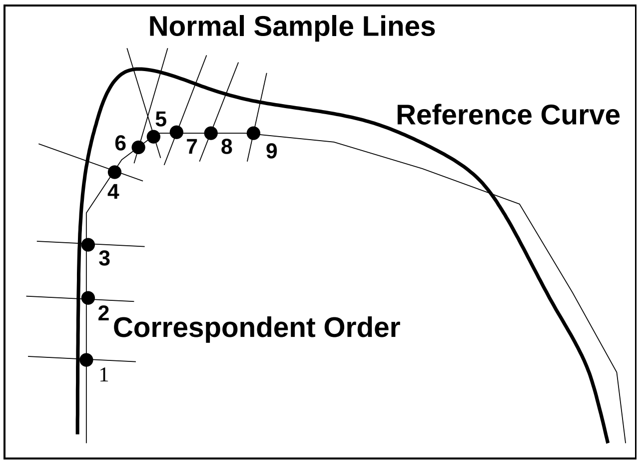


Figure 4.5: Correspondent Inversions. The bold line represents the reference curve. Sample lines are fired off at points along its length and the intersection computed with the target curve. The numbers indicate the sequence order of the computed intersections i.e. the sequence in which they should be connected to form a new curve. Clearly points 5 and 6 are inverted. The combination of relative displacement and differences in local geometry combine to create this inversion.

The result of this process is a set of resampled meshes, with the same connectivity relationships as the reference mesh. Each mesh may be represented as a vector of ordered vertices — the ordering is important — and submitted to the PDM algorithm. Section 4.5 examines the issues associated with each correspondent selection scheme.

4.4.2 Dealing with Correspondence Errors

If the geometry of the surfaces is complex, or the registration too coarse, the association of points across the two surfaces may yield erroneous matches. These mismatches will manifest as *correspondent inversions*, which violate the mesh connectivity ordering imposed by the resampling strategy. More precisely, if we connect each pair of correspondents with a straight line, these lines should not cross each other. If such crossings exist, they imply that the vertex ordering of the resampled mesh has been compromised. Figure 4.5 illustrates this problem for the case of two plane curves. In this example the resampled polygonal curve would fold back on itself if the new vertices were connected in the correct order. The detection of such inversions is trivial in the simplex framework: mesh points with negative *metric parameters*, $\{\epsilon_1, \epsilon_2, \epsilon_3\}$, are designated as inverted. The metric parameters provide barycentric coordinates for the projection of the point onto the plane passing through its three neighbours (see Appendix B for details). A negative parameter indicates that this point has a projection which falls outside of the triangle described by its three neighbours — a violation of the regularity conditions imposed on the mesh (see Appendix B).

For each point on the resampled mesh which violates the metric constraint, the centroid of its 3 neighbours is computed and the closest point to this centroid on the registered target mesh

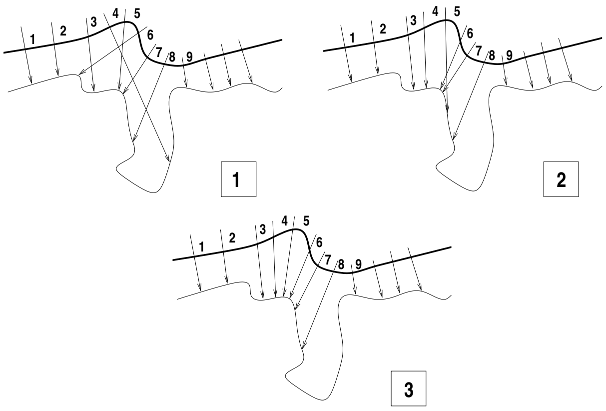


Figure 4.6: Example of Vertex Redistribution for a Curve. We start off by looking for correspondences on the target curve. Diagram 1 shows the initial correspondent approximation (these were based on normal resampling). After computing the centroid of each inverted triple and recomputing the closest point, we get Diagram 2. After one more iteration we have no more inversions (Diagram 3) and each point satisfies the metric constraint for a 2-simplex i.e. every point has a projection onto the line between its 2 neighbours which lies between those neighbours.

becomes the new approximation for that correspondent. Figure 4.6 illustrates the idea for a contour — in the 2D case the centroid is computed using the 2 neighbours of the point under consideration. The rationale behind this is as follows: because the two surfaces are close to one another, and similar, most of the correspondences will be correct[2]. The problematic points will be surrounded by the patches of good data; we can interpolate this data, subject to the points lying on the surface and having positive metric constraint, by applying this "centroid-projection" scheme iteratively. This has the effect of unknotting fold-overs and providing a best compromise mapping for the correspondents which were initially "snarled".

Algorithm 4.2 (Vertex Redistribution)

1. Repeat until metric constraint is satisfied for all points or until iteration count is exceeded:
   1.1 For each point
      1.1.1 Evaluate $\{\epsilon_1, \epsilon_2, \epsilon_3\}$
      1.1.2 If metric constraint violated
      1.1.2.1 Compute centroid of 3 neighbours, C
      1.1.2.2 Find closest point to C on the target mesh
   1.2 Swap newly computed points with old points.
2. If there are still points which violate the constraint,
   2.1 stabilise the mesh by repeating the above step, but without searching for the closest point any longer.

If the registered meshes are poorly aligned, or the intrinsic geometry too complex, the number of points which violate the metric constraint may be non-zero after a large number of iterations. This means that the action of projecting onto the target mesh leads to a violation

[2] At least approximately — it is difficult to assert that any correspondence is ever truly correct.

of the metric constraint (prior to the projection, the point — being the centroid of its 3 neighbours — has equal, positive metric parameters). This, in turn, means that the closest surface point is very different from its neighbours, all of which lie on the target mesh surface. In this case, we just interpolate the good data, by running the algorithm without requiring that the problematic points lie on the surface. The full algorithm is presented above.

It should be noted that the application of this algorithm does not guarantee that the final correspondent set will be optimal — this is an almost impossible task without the specification of additional *a priori* information. The next section examines the effects of the registration transformation on the selection of the correspondent set.

4.5 Analysis of Registration Schemes

Before examining the efficacy of the registration step, the purpose of this process must be clarified. We seek a way of normalising the data so that the only variability contained within the transformed training set represents the true variability of the class. Since the template (mean shape) involves spatial averaging, maximal alignment of the data is crucial for the creation of a representative object. The template will serve as the initial shape instance approximation. Since there is no *a priori* landmark information, we cannot use a registration scheme based on such discriminants: the normalisation is intended to allow us to extract these points!

We wish to learn the structure of the brain, in order to quantify deviation from "normality". To this end, we are prepared to learn redundant information not eliminated by our normalisation — as long as it does not impact on subsequent segmentation efforts.

Two classes of transformations have been considered

Bounding Box These global transformations uses the MSP to construct an object centred coordinate frame — represented as a bounding box — for each mesh. The registration is performed by mapping the source bounding box onto the reference bounding box, subject to the specified global mapping (rigid or affine).

Iterated Closest Points The original ICP formalism [5, 73] allowed for the global registration of two point sets. The extensions introduced in [33] permit non-rigid registration, using a volumetric spline technique [29]. The ICP approach has the advantage of requiring no additional information beyond two point sets.

We are not free to choose an arbitrary mapping. The choice of transformation will be constrained by the assumptions detailed in Section 4.4. In particular, we require that the surfaces be as close to one another as possible after registration, since this will minimise the incidence of spurious matches.

In an effort to quantify the effects of the transformation on the selection of the correspondent set, the following measures have been used:

Inversion Error As the surface registration becomes more accurate, one would expect the number of correspondent inversions to decrease. In the (unlikely) case of perfect registration, the correspondents are trivially identified as the same point on each registered surface (by means of either correspondent selection strategy — they are identical in this pathological case). The initial proportion of correspondents which are inverted, and the rate at which the inversion correction procedure converges, are taken as indicators of the efficacy of the normalisation and the intrinsic correctness of the correspondent set.

Resampling Error Closely allied to the first measure is the idea of quantifying the mesh "resampling error". If one disregards discretisation errors, then a perfectly registered surface would experience no change of volume or surface area (cf. Appendix C) when resampled in the manner described previously. However, if the correspondent selection process was largely unsuccessful, the geometry of the resampled surface would vary — perhaps very markedly — from its original form. By quantifying the difference between the prototype and its resampled version, through volume and surface area changes, as well as surface separation statistics, the effects of normalisation mappings may be compared.

Projection Error By projection error we mean the discrepancy between a surface and its best approximation under the retained PDM basis. The error is represented through surface separation statistics (mean and standard deviation over the two surfaces). The purpose of correspondent selection is to provide the input vectors necessary for the construction of the PDM. Intuitively, if correct correspondences have been selected, one would expect the PDM to better encapsulate the true variability within the registered training set. Given the enormous number of vertices, and the small number of training examples, a reduction in the number of modes required to represent the chosen level of variability is unlikely[3]. However, a more representative PDM should enable one to approximate shape instances within the class to higher accuracy using the set of retained modes. Consequently, by comparing the original mesh surface and its best approximation under the PDM, it should be possible to deduce something about the usefulness of the correspondent selection scheme. Of course, the extent to which registration aligns the training set will also influence these numbers. However, since the selection of the best registration scheme is part of what we wish to establish, this

[3]The number of modes cannot exceed the number of training examples. The variability is consequently forced into these modes and unless there is low variation over the training set, the maximal number of modes will be required to achieve the best representation.

is appropriate. It is worth emphasizing that the difference metric must be evaluated after the surfaces have been registered. The transformation should be the same as that used to register the training set; the "global" information removed by this transformation will not be reflected in the PDM.

Qualitative Evaluation The qualitative assessment of the selected correspondents is provided by several graphical renderings. In the first, the reference surface and the surface to be registered are presented from similar views and the selected correspondents are indicated on each. The second rendering shows the localisation of correspondents across the training set, by overlaying a random subset of the selected points. One would prefer that the spatial clustering of the points across patients — for each correspondent — be as centralised as possible, since this implies that the associated PDM shape space will contain less variability. This is desirable since the PDM segmentation is less likely to admit spurious shapes.

4.5.1 Type I: Bounding Box Registration

The definition of an object-centred description is central to this approach. It is assumed that the MSP has been extracted. This may be accomplished by selecting points lying along the mid-saggital line in a number of slices, and fitting a plane to them using orthogonal regression — See Chapter 3. The MSP normal provides one coordinate axis for the bounding box. By projecting the mesh points onto the MSP and computing the two principal axes of this point distribution, a consistent set of coordinate frame may be constructed.

The principal axes of such a point distribution are influenced by the arrangement of the projected points. Clearly this is undesirable as it implies that models which have the same shape, but which are sampled unevenly, will yield different coordinate frames. We require (at least approximate) invariance to point density (at a given resolution — we will use a dense mesh to minimise discretisation errors).

To overcome this limitation, we compute the *convex hull* of the planar point set [58] and uniformly sample the returned polygonal contour. This representation, which is invariant to the internal distribution of the projected points (modulo mesh discretisation errors), is then subjected to a PCA — Figure 4.7 — to obtain the requisite coordinate axes.

Once the reference mesh has been selected, the associated plane-constrained bounding box is computed and mapped onto the origin, such that the (right-handed) coordinate frame is aligned with that of global coordinate system. For each remaining training example, the bounding box is computed and the affine transformation which maps this bounding box onto that of the reference is applied to register the source mesh with the reference. The correspondent selection scheme is then applied, as discussed above.

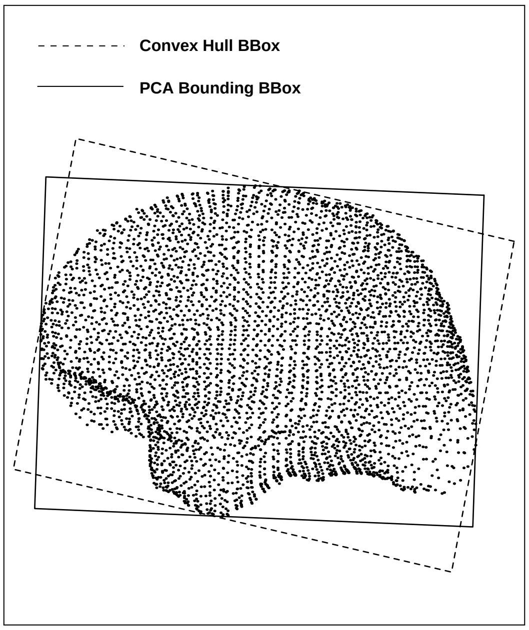


Figure 4.7: Comparison of PCA Methods A different bounding box results if the convex hull of the point set, rather than the point set itself, is used to compute the PCA. The convex hull method ensures invariance of the PCA to the internal arrangement of points.

4.5.2 Type II: ICP Registration

The ICP algorithm [5, 73] proceeds by iteratively refining the transformation which best aligns two sets of points. In the standard framework, a simple Euclidean distance metric is used to determine a set of likely "corresponding" pairs between the two points sets. A minimisation is then performed to find the (global) transformation which best associates these putative matches, prior to the extraction of our correspondent set. It should be noted that the "correspondences" utilised by the ICP algorithm are not, in general, the correspondences we seek. The ICP pairwise matches are based on a closest point selection strategy and can only identify points within the supplied point sets.

Given two point sets, S_1 and S_2, representing the vertices of the two mesh surfaces we wish to register, we seek the transformation T which minimises the criterion

$$D(T) = \sum_{P_i \in S_1} \|T(P_i) - \mathrm{clpt}_{S_2}(T(P_i))\|^2, \tag{4.15}$$

where $\mathrm{clpt}_{\mathrm{S}_2}(\mathrm{x})$ returns the closest point in the point set S_2 to $\boldsymbol{x}$.

The two point sets are filtered to eliminate suspected mismatches in order to improve the robustness of the registration. In [29], two methods are identified:

1. A fixed distance threshold can be used to eliminate patently spurious matches (based on *a priori* experience).

2. A threshold based on a χ^2 statistics is proposed to reject unlikely matches.

The ICP implementation used here is based directly on the work of Feldmar and DeClerck [33, 29]. This work introduced notable improvements over the original ICP algorithm, such as the modification of the "closest point" metric to include differential information (when available). The most important modification — for our work — was the introduction of a non-rigid registration algorithm. By utilising a space-deforming spline, it is possible to map one set of points onto another in a smooth, local fashion. For the work presented here, differential information was not utilised.

The Affine ICP registration is initialised using the best rigid transformation, computed using the ICP procedure. This is done to ensure stability — the affine parameter space is of sufficiently high dimension that local minima can have a marked impact on the solution. By bootstrapping the parameter search in this way, a better solution is more likely to emerge.

If spline registration is required, the best affine transformation serves as an initialisation. The volumetric spline is defined by a regular grid of 3D control points, which directly affect the magnitude of permissible warping.

The spline deformation vector $(U, V, W)^T$ is defined as [29]:

$$U(x, y, z) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} u_{ijk} B_{i,K}^x(x) B_{j,K}^y(y) B_{k,K}^z(z) \tag{4.16}$$

$$V(x, y, z) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} v_{ijk} B_{i,K}^x(x) B_{j,K}^y(y) B_{k,K}^z(z) \tag{4.17}$$

$$W(x, y, z) = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} \sum_{k=0}^{n_z-1} w_{ijk} B_{i,K}^x(x) B_{j,K}^y(y) B_{k,K}^z(z), \tag{4.18}$$

where $n_{x,y,z}$ specifies the number of control points for each dimension, $B_{i,K}^{x,y,z}$ are the K-th order B-spline basis functions and $(u, v, w)_{ijk}^T$ are the control points at index (i, j, k). For this work, $K = 3$, since cubic B-splines are employed. The knots are regularly spaced throughout the volume. A weighted "Tikhonov stabiliser" (regularising term) given by

$$E(U) = \int_{\mathbb{R}^3} \left[\frac{\partial^2 U}{\partial x^2}^2 + \frac{\partial^2 U}{\partial y^2}^2 + \frac{\partial^2 U}{\partial z^2}^2 + 2\frac{\partial^2 U}{\partial x \partial y}^2 + 2\frac{\partial^2 U}{\partial x \partial z}^2 + 2\frac{\partial^2 U}{\partial y \partial z}^2 \right], \tag{4.19}$$

is added to the distance criterion, Equation 4.15, with similar expressions being defined for $E(V)$ and $E(W)$.

Once the surfaces have been registered, and correspondents extracted, an "inverse" spline mapping is performed which warps the resampled mesh back on to the result of the best affine registration. This is done since, firstly, the computation of a local spline map is too costly to implement during segmentation (when the model is deployed) and secondly, the spline transformation has a large number of DOF. This increases the likelihood of running foul of local minima and reduces the utility of the PDM, which is intended primarily as a means of constraining the segmentation.

By utilising the best affine map to register the meshes, and the best spline map to compute correspondents, the end result should be better.

4.6 Results

The results presented in this section are based on the construction of a PDM using 20 training examples (11 males and 9 females). Since only these examples were available, a single template was generated rather than two gender-specific templates. Although there is no easy criterion to select the number of training examples, quick experiments showed that 20 training examples provided a satisfactory variability coverage for the smoothed brain surfaces we wished to extract. Since the largest sex-specific difference is one of size, and the affine registration used prior to template construction controls for this aspect, the constructed template is still useful. Furthermore, the primary purpose of this chapter is to evaluate the template construction procedure, rather than the generation of the best template for a immediate clinical application.

The qualitative visualisations are presented for the case of the "worst" mesh i.e. the one which was most poorly represented under all our measures of "goodness".

4.6.1 Vertex Redistribution

The initial incidence of correspondent inversions varied significantly depending on the choice of registration scheme — as we had anticipated. Figure 4.8 shows the rate at which the vertex redistribution corrected the initial inverted correspondents when Normal Sampling was utilised. This should be compared with Figure 4.9, which shows the details for the case of Closest point Sampling. Recall that the total number of correspondent pairs is 4500.

It is immediately obvious that the Closest Point selection performs significantly better than the Normal Sampling scheme. There are fewer initial inversions, and the rate at which the vertex redistribution removes inversions is higher.

In some instances, a small set of points remain which must be stabilised using the process discussed in Section 4.4.2. This is a consequence of mesh discretisation error: the approximating triangulation cannot adequately represent the underlying surface — implicitly defined

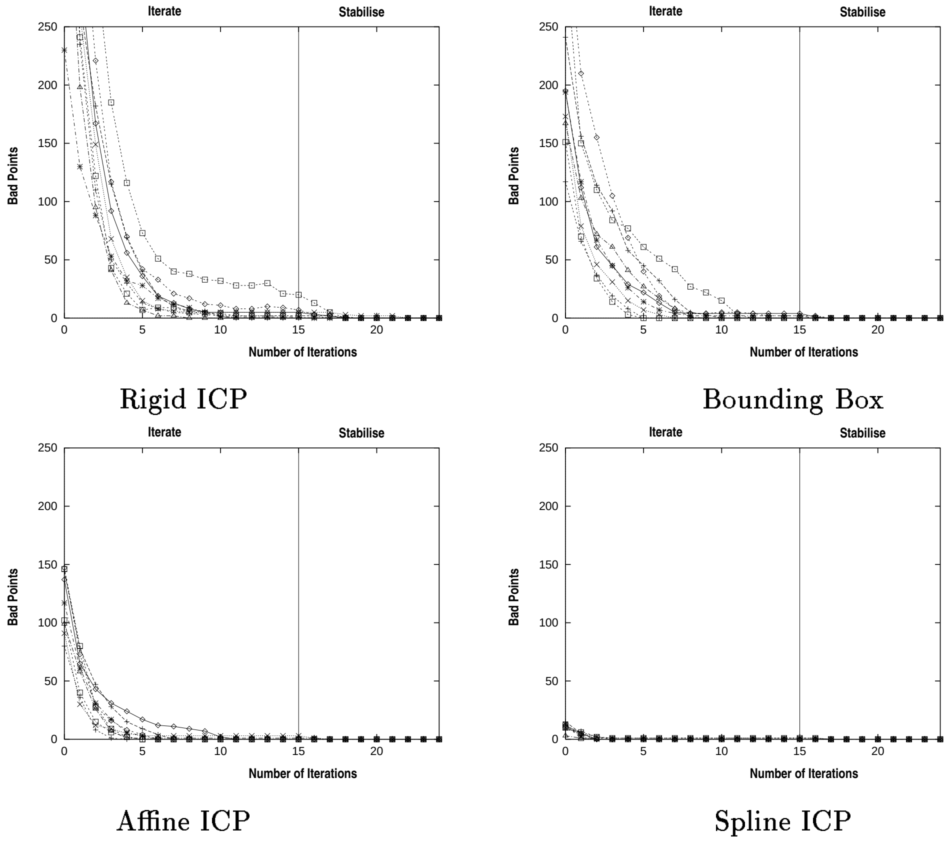


Figure 4.8: Normal Selection Vertex Redistribution The graphs show the initial incidence of correspondent inversions induced by the Normal Sampling correspondent selection scheme, as well as the rate at which the vertex redistribution converges. In general, the scheme converges exponentially, with one or two irreducible metric errors. These points have to be "stabilised" as discussed in Section 4.4.2.

by the simplex mesh — over every region. However, since the number of points affected is negligible (1 to 8 out of 4500, for Closest Point sampling) in all our experiments, we did not consider this a suitable motivation for increasing the density of the meshes.

Of the registration schemes discussed previously, the spline correspondent selection resulted in the fewest inversions, with the affine scheme coming a close second. Figure 4.10 shows the registration of the surfaces prior to the computation of the correspondent set. The bounding-box affine map produces a less satisfactory result than the ICP affine matching scheme. This is a consequence of the constraints implied by the transformation: only 9 parameters are available, compared to the 12 of a full affine transformation.

4.6.2 Resampling Errors

The data we collected for the resampling process showed that the choice of registration method had only a marginal impact on the distribution of resampling errors. In general, the relative change in volume was below 1%, while the relative change in surface area was

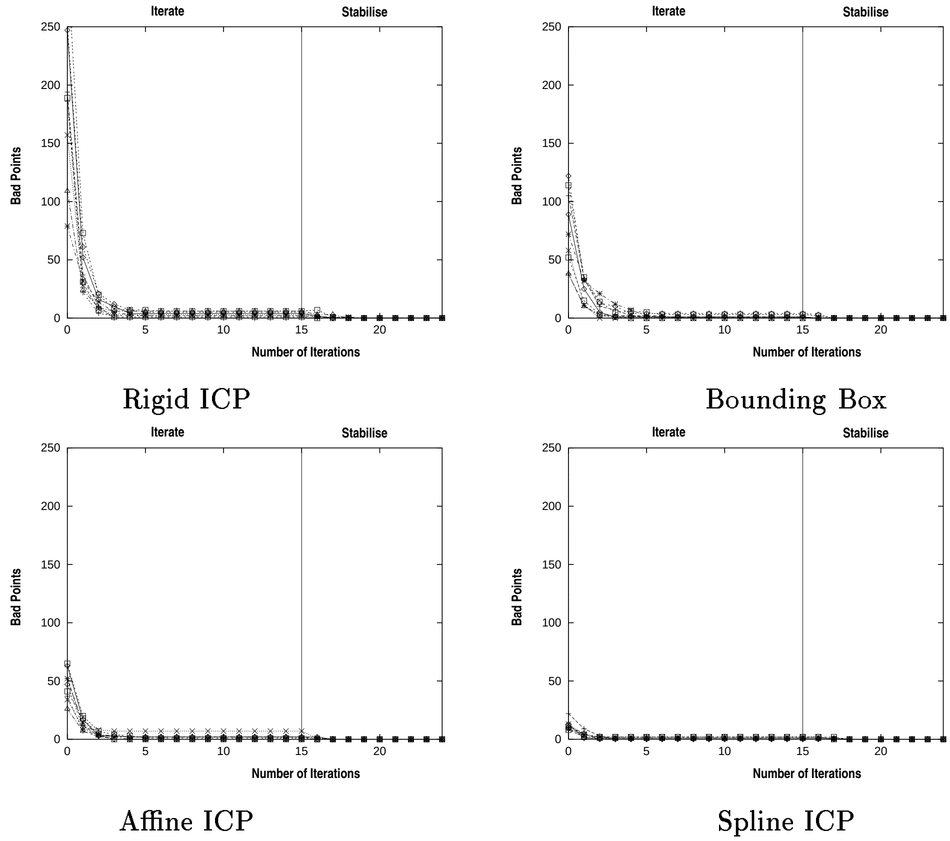


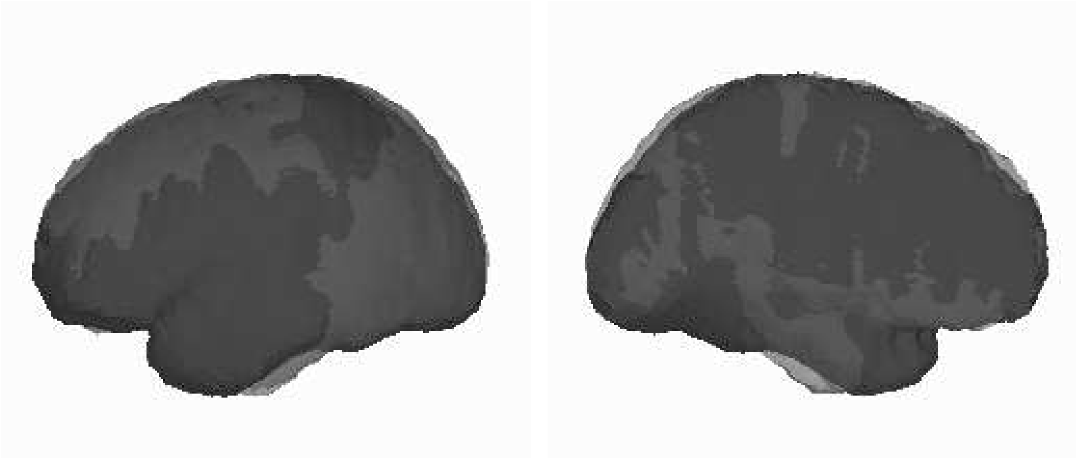
Figure 4.9: Closest Point Vertex Redistribution The graphs show the initial incidence of correspondent inversions induced by the Closest point Sampling correspondent selection scheme, as well as the rate at which the vertex redistribution converges. The vertex redistribution converges more smoothly than for the Normal Sampling case, and exhibits fewer initial inversions.

below 2%. The particular correspondent selection scheme was also largely irrelevant. These observations show that the metric normalisation scheme is quite robust: inversions will be rectified and as a consequence of this process, a fairly uniform and representative discrete surface will result. It is important to emphasize that the volume and surface area changes also reflect the discretisation scale of the resampled mesh.

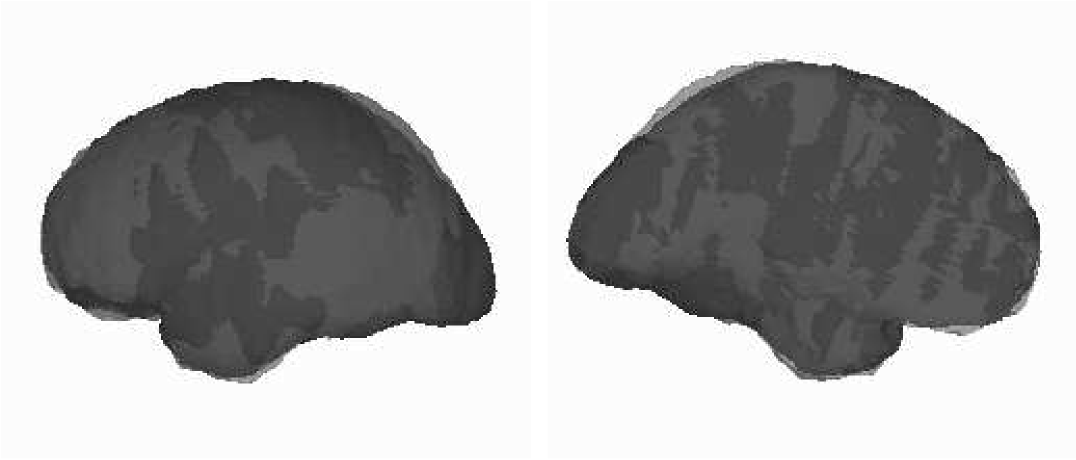
4.6.3 Projection Error

The PDMs associated with the correspondent selection schemes required from 14 and 17 modes (out of a possible 20 — the covariance matrix has rank 20) to obtain 95% coverage of variability. This level of coverage is considered to provide an adequate degree of shape encapsulation [24]; additional modes did not noticeably improve shape approximation (for our data). Figure 4.11 shows the mean and the effect of first mode of variation for the Affine and Rigid PDMs. For the Rigid case, the first mode is clearly a global scaling, while in the Affine case it a more complex deformation.

A



B



C

Figure 4.10: Global Registration The effects of the different global registration mappings are shown above, for the case of the left cerebral hemisphere. In order to show the accuracy of the registration, the reference mesh has been rendered using translucency (pale shade), whilst the source mesh remains opaque (dark shade). Each row shows two views of the registered surface pair. A) shows the Rigid ICP map, B) shows the Affine Bounding-Box map and C) shows the Affine ICP map. The Affine ICP transformation yields significantly better matching over the temporal lobe than either of the other two schemes; the rigid transformation produces a poor result almost everywhere (as one would expect).

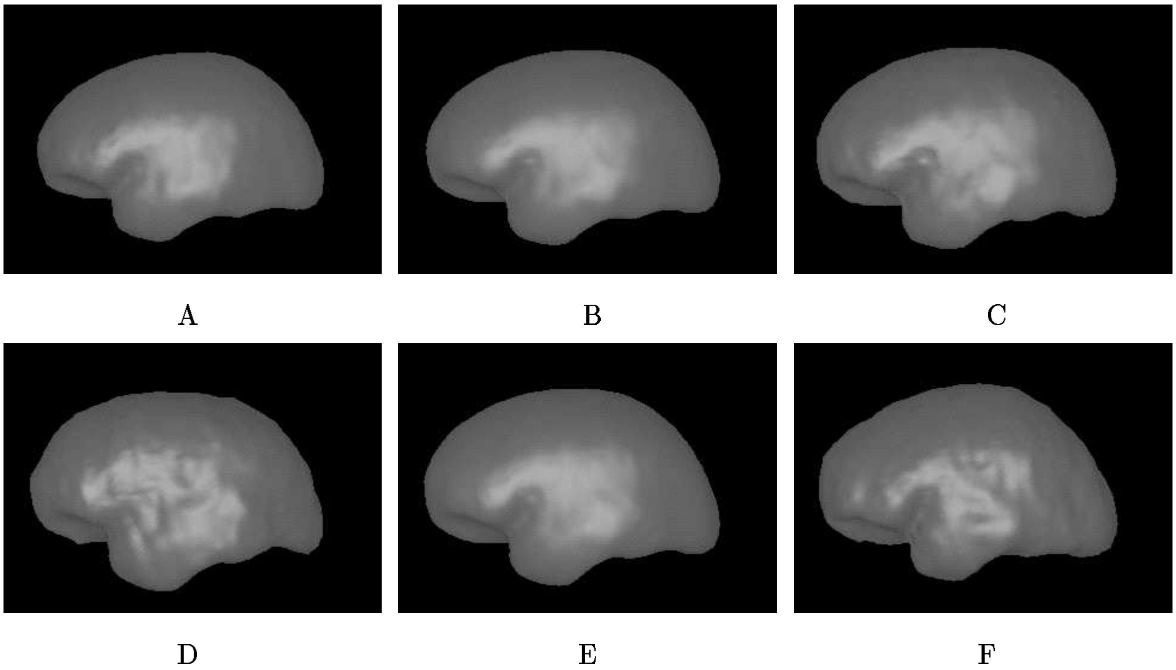
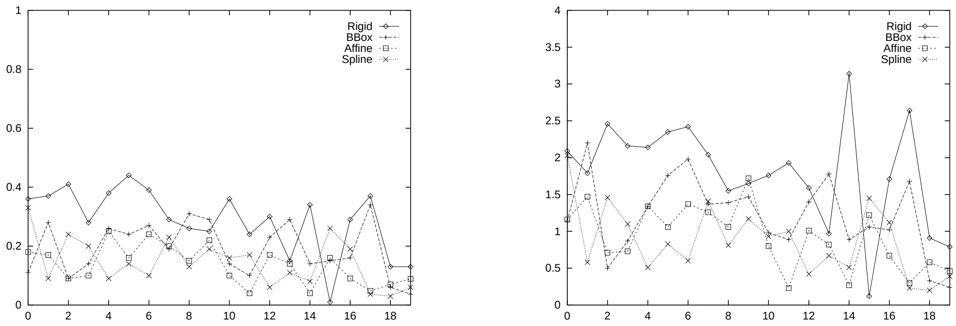


Figure 4.11's panel labels: A, B, C (top row) and D, E, F (bottom row).

Figure 4.11: PDM Deformation Modes The mean and first mode of the PDMs associated with the extracted correspondent set for A—C) Rigid ICP registration and D—F) Affine ICP registration. In each case, the mean shape, M (B,E) is shown as well as $M \pm 2\sqrt{\lambda_1}M_1$, where M_1 is the first (most important) deformation mode and λ_1 is the eigenvalue associated with that mode (which measures the proportion of variability it captures). The first mode of template constructed using Rigid ICP seems to correspond to global scaling. For the Affine ICP template, the first mode is not as easily interpreted. It should be recalled that both M and M_i are vectors of dimensions $3N$, where N is the number of points in the PDM. This kind of visualisation shows some of the shapes the template is capable of representing.



A B

Figure 4.12: Projection Errors A training set of 20 mesh surfaces was used to generate a PDM. Fourteen modes explained 95% of the variability within this set of shapes. A) shows the average surface separation (in mm) after approximating each surface in the training set using the retained modes. B) shows the associated maximum separation for each surface.

The projection error will have a lower bound induced by (potentially) poor variability coverage. To minimise this effect, we have evaluated the projection error only for those mesh surfaces which formed part of the training set. These measures are aimed at evaluating the effects of the registration transformation and choice of selection scheme, rather than the degree to which the PDM is able to represent arbitrary shape instances; that issue will be addressed in a later chapter. With the exception of one example generated from the Rigid PDM, the relative change in both surface area and volume was well below 1%. As Figure 4.12A shows, the average surface separation was less than 0.5mm in all cases. The standard deviation of this measure was also within $\pm$0.5mm. The maximum separation — Figure 4.12B — is the greatest for the Rigid scheme, between 1—3.5mm. The remaining schemes produce very similar results, with maximum separations of 0.5—2mm. It is clear from our data that the Rigid scheme provides the least satisfactory PDM — as one would expect.

4.6.4 Qualitative Assessment of Correspondences

The qualitative evaluation of the correspondent set is accomplished by means of a number of visualisations, which show the selected points on the two surfaces we have chosen. As we have stated before, no *a priori* information was utilised to obtain these correspondents. The results presented here are based on the closest point selection scheme, since this is the method we intend to use for future correspondent extractions

A randomly selected set of mesh points were extracted on the reference mesh, and the correspondents computed on the test surface. A number of views of the computed correspondent sets are shown in Figure 4.13 and Figure 4.14. It is immediately apparent that a consistent set of correspondents is extracted for all the proposed transformations, even for the rigid case. However, the points identified following the Affine ICP registration appear to provide the most satisfactory results: the relative placement of points is qualitatively correct. The addition of the Spline deformation does not appear to have improved matters noticeably — a possible explanation is given in Section 4.7.

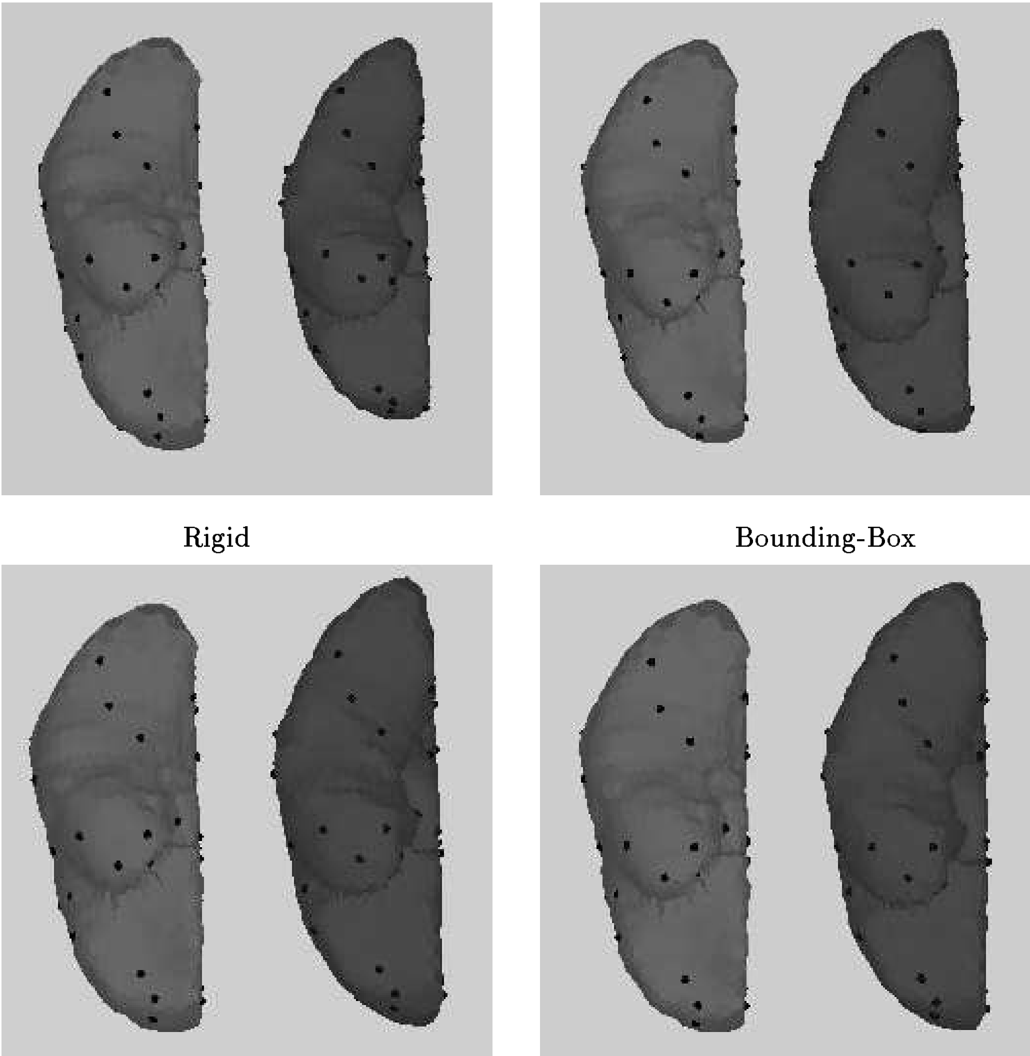
Figure 4.15 shows the point clustering of the correspondents over the training set. The effect of the registration is clearly evident: the correspondents extracted after rigid alignment are more widely scattered, while those extracted after Affine ICP alignment are comparatively dense. Interestingly, the Spline correspondent scheme produced a somewhat more diffuse arrangement than the plain Affine case. This may be due to regularisation problems, as suggested in Section 4.7.

4.7 Discussion

On the basis of the quantitative results presented above, the correspondent selection scheme should be based on the Closest Point sampling strategy, since this ensures a lower number of correspondent inversions and more rapid convergence of the vertex redistribution procedure. Intuitively, a lower number of inversions suggests that the manner in which the correspondents have been chosen is closer to the underlying reality. Some caution is necessary when considering this issue. One question in particular requires some thought: is the choice of the closest point a natural way of selecting a correspondent? If we consider two surfaces such as those shown in Figure 4.16, the choice of selection method is not so clear cut. When the source mesh "grows" into the target Figure 4.16A), the Normally selected correspondents produced a more natural correspondent selection. Conversely, if the source mesh "shrinks" towards the target — Figure 4.16B — the Closest Point selection strategy results in a better correspondent set. Unfortunately, because of its susceptibility to local geometry, the Normal Selection method is prone to excessive correspondent inversions which must then be rectified. Although the vertex redistribution procedure should regularise the distribution of selected points, this regularisation depends on the spatial location of the "good" i.e. non-inverted correspondents. Clearly, if there is doubt about the validity of these, confidence in the final correspondent set will be lower.

These troublesome questions can be evaded altogether if the registration succeeds in matching the surfaces to a sufficient degree. In the ideal case, the surfaces would be superimposed ensuring that no ambiguity in the choice of correspondents were possible. Of course, this is unlikely to occur in practice!

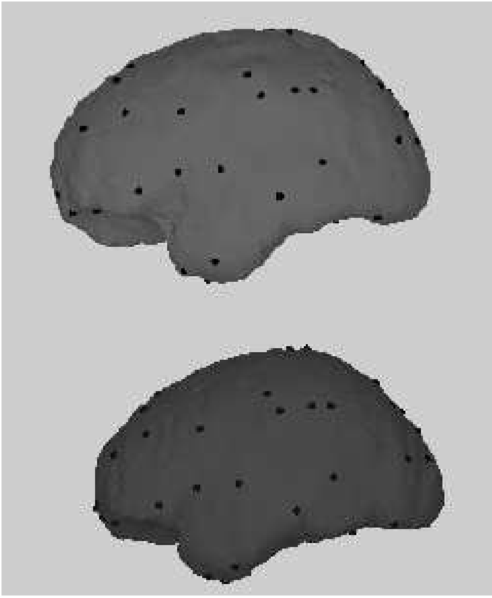
By examining both the quantitative and qualitative results presented above, it would seem that the Affine ICP mapping provides the most satisfactory global transformation for the registration step. In this case, the incidence of correspondent inversions is very low (less than



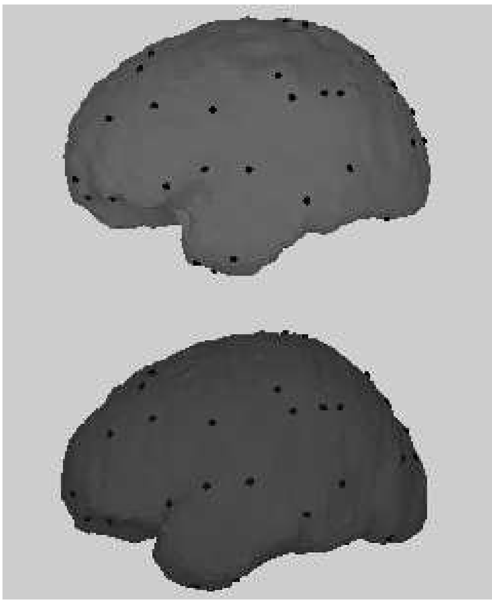
Rigid Bounding-Box

Affine ICP Spline ICP

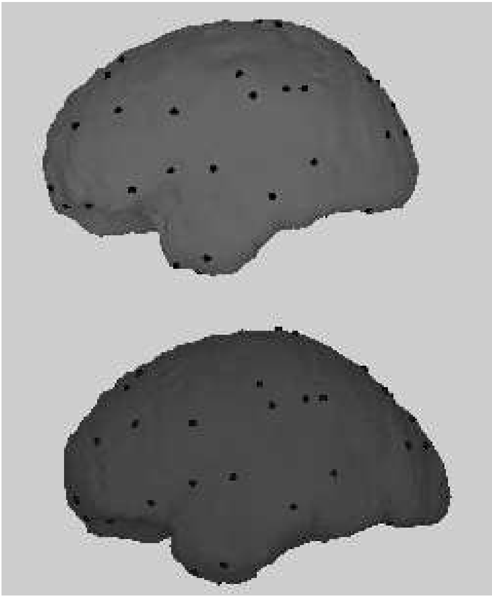
Figure 4.13: Randomly Selected Correspondents (Bottom View) A comparative sequence of views showing an arbitrary subset of correspondents for each of the proposed schemes. The reference mesh is on the left (lighter colour). Each mesh has been aligned using the best global transformation (affine or rigid) and separated by a shift to allow better appreciation of the correspondent choices. The Affine ICP method seems to have delivered the best set of correspondences. However, given the crudity of the transformation, and the high incidence of correspondent inversions, even the Rigid ICP scheme produced reasonable results.



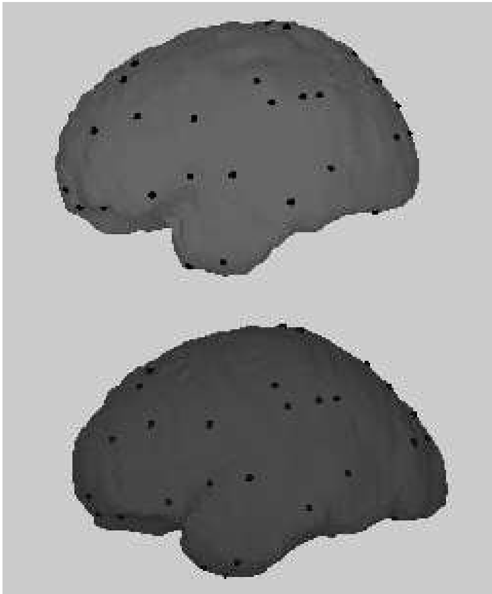
Rigid ICP



Bounding-Box



Affine ICP



Spline ICP

Figure 4.14: Selected Correspondents (Side View) The reference mesh is the lighter of the two, and the associated correspondents are indicated on each surface. The correspondents are the same as those displayed in Figure 4.13. From this view, the schemes are qualitatively similar.

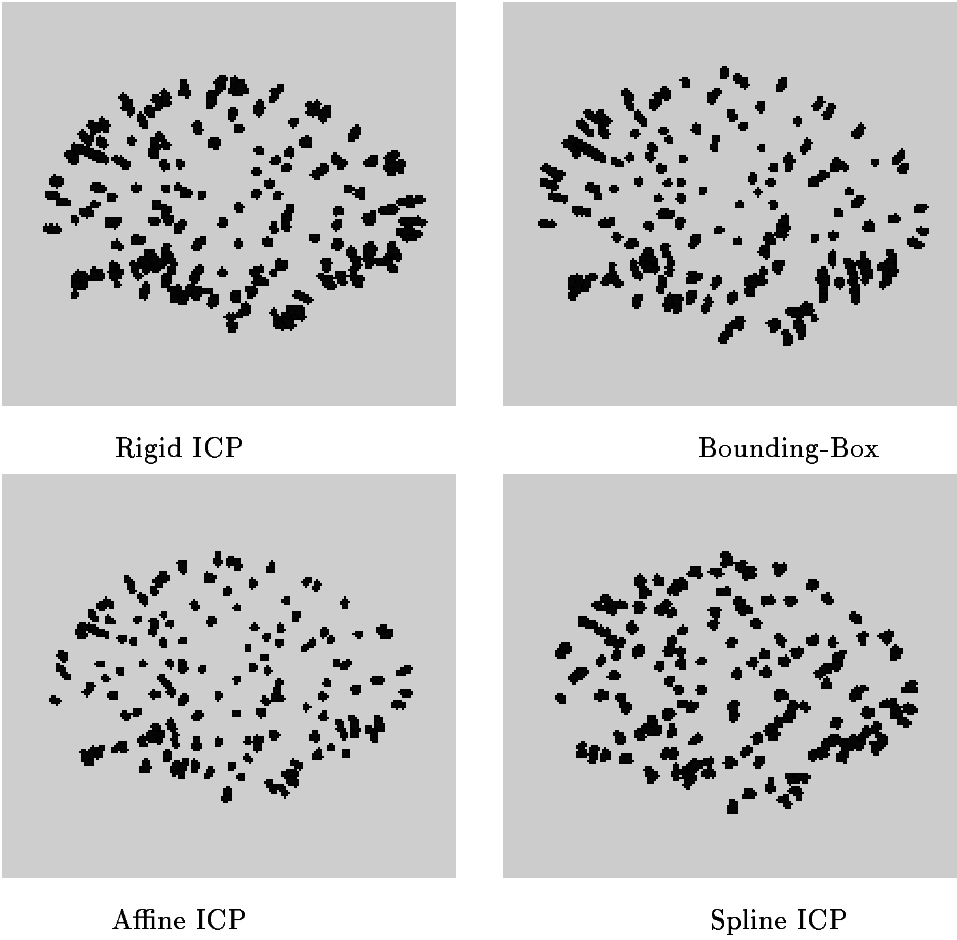


Figure 4.15: Correspondent Point Clustering The clustering of correspondents shows increasing centralisation as one goes from Rigid ICP to Affine ICP. This shows how the particular choice of registration enables one to remove redundancy from the PDM — the more centralised each "correspondent cloud" is, the less variability will be encapsulated by the model.

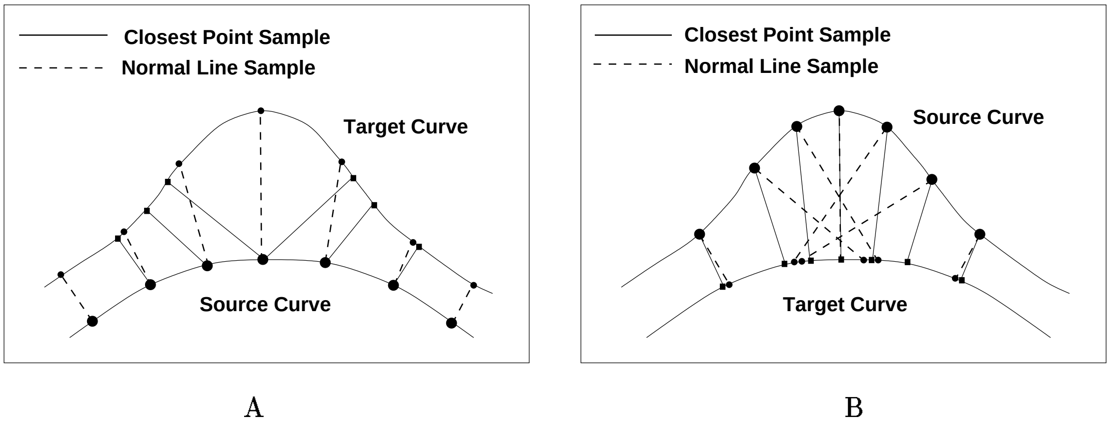


Figure 4.16: Correspondent Selection A) If the first contour "grows" towards the second in an accretive manner, then normal sampling is appropriate. B) However, if the curves "shrink" into one another, then closest point sampling yields a more intuitive correspondent set.

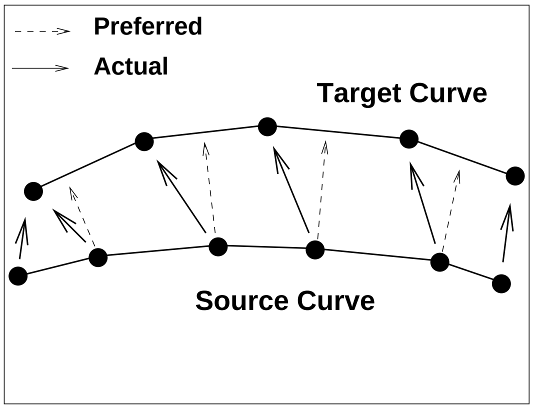


Figure 4.17: Spline Deformation The ICP spline deformation attempts to draw closest-point pairs towards one another. Unless the two point sets are very dense, the local deformation will not generally produce the kind of normal flow we would prefer.

2% for the closest point selection scheme) and large parts of the two surfaces are in alignment. This would suggest that either sampling scheme can be utilised with confidence.

In an attempt to provide even greater confidence, we applied an additional spline deformation, as explained in Section 4.5.2. By utilising this technique to warp the two globally registered surfaces onto one another, it was hoped that the sampling scheme would become superfluous and the identification of correspondents trivial. There are, unfortunately, certain factors which reduce the desirability of this approach. Foremost amongst these is the manner in which the spline warp accomplishes the registration. Since the deformation is based on the alignment of associated point pairs, the density of the mesh surfaces will affect the computed warp. The spline will attempt to move each point towards the closest point on the second surface. If this surface is sampled (comparatively) coarsely, the point will tend to move in a counter-intuitive manner — Figure 4.17 — unless a high degree of regularisation is in operation. One could sample the target surface very densely, but there is a heavy computational penalty associated with this. The cheapest "solution" is to increase the regularisation, but this leads to less satisfactory registration, since the spline becomes less flexible.

Although Figure 4.13 would suggest that there is little qualitative difference between the Spline and Affine cases, the less satisfactory clustering behaviour, coupled with ambiguous quantitative results, led us to reject the Spline approach in favour of the (Closest Point) Affine ICP scheme.

A related approach [15] to establishing surface mesh correspondences has recently been brought to our attention. This method uses the ICP algorithm to establish initial point correspondences after decimation of the surface pair. It is assumed that if the two sets of point correspondences have the same mesh connectivity, then the correspondences are correct. There is no guarantee that the ICP algorithm will yield "correct" initial correspondences, particularly when the matches must be drawn from decimated vertex sets. The sparse set of correspondents is interpolated to yield a dense correspondent set. Since no further processing occurs to validate the initial point selection, this method may yield an unsatisfactory result when complex geometry is involved.

4.8 Conclusion

Discussions with neuro-anatomists suggest that — at least across each sex — the structure of the brain is qualitatively similar. Gross differences may be removed by a process of normalisation, whereby the brain surfaces are approximately registered using a particular family of global transformations. Once such a registration has been accomplished, we are able to utilise an *ad hoc* scheme to construct a set of corresponding points on each surface, without the need for an explicit surface parameterisation or the specification of landmarks. These correspondences enable us to construct a Point Distribution Model which will be used to constrain our subsequent segmentation.

Since we are unable to segment deep gyral structure — our data is sparse — the surfaces used in the training process are taken to represent the *arachnoid* — a membrane which adheres to the brain surface, but does not penetrate sulci to any great extent. As a consequence, the mesh surfaces used in training are comparatively smooth and the correspondent extraction process quite robust. From both a qualitative and quantitative perspective the use of a global Affine transformation, computed using the ICP procedure, produced the most satisfactory selection of correspondents.

Chapter 5

Boundary Detection

The reliable detection of points on the brain surface is of fundamental importance to the mesh segmentation scheme. The accuracy and robustness of the boundary detection scheme will affect the utility and accuracy of the segmentation. Unfortunately, the identification of these points is not a trivial task. The boundary we seek is associated with complex anatomy, and may be badly degraded by the sparsity of the data. To overcome these difficulties, prior information is mobilised at two levels. Firstly, knowledge of the imaging protocol (scan weighting and other acquisition parameters) is exploited and, secondly, anatomical constraints are imposed. This chapter concerns the development of a boundary detection scheme which exploits the information at our disposal.

Section 5.1 describes the structure of the brain surface, and explains why the task of boundary detection is not trivial. In Section 5.2, we outline our basic methodology and propose a number of possible boundary modelling schemes. These schemes are evaluated in Sections 5.3,5.4 and 5.5. The issue of cortical folding is addressed in Section 5.6. Boundary detection results are presented in Section 5.7, and the chapter concluded in Section 5.9.

5.1 The Brain Surface Boundary

An idealised cross-section of the brain surface neighbourhood is shown in Figure 5.1. There is a multitude of membranes, layers and interstices ("sinuses") — see, for example, [66]. Under most MR protocols, the skin[1] and skull are easily distinguishable, as is the *venous sinus*, an intersticial layer filled with blood.

The first structure one encounters when moving inwards from the skull is the *dura mater*. This is composed of an outer layer — which is attached to the inside of the skull — and an inner layer which surrounds the brain and penetrates the inter-hemispheric fissure. The

[1]By which we mean skin, muscle and fat!

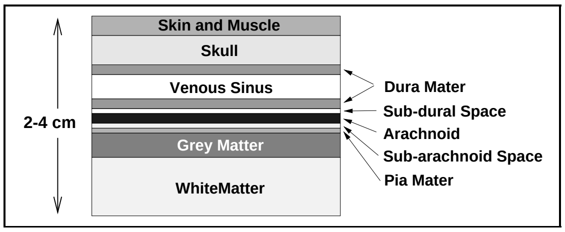


Figure 5.1: Cerebral Boundary Structure The brain is separated from the skull by a series of membranes and fluid filled sinuses, known collectively as the meninges. Each layer will have a slightly different intensity, and the localisation of the desired boundary transition is not easy without prior knowledge.

venous sinus is sandwiched between these two layers. The dura is followed by the *arachnoid*, a membrane which follows the brain surface closely, but does not penetrate the sulci. Finally, one encounters the *pia mater*, a thin film of tissue which lies on the cortical surface and follows every convolution intimately. In addition to these structures, there are two fluid-filled sinuses: the *sub-dural* and *sub-arachnoid* spaces. The collection of layers and sinuses (excluding the pia mater) is referred to as the *meninges*.

Each layer will have a somewhat different intrinsic MR signal, giving rise to a complex gradient signature which contains the boundary we wish to identify. In general, the detection of the boundary from such a signal is difficult without *a priori* information. Matters are complicated by the averaging involved in the acquisition process: many of the meningial layers are obscured or blend into one another; CSF will usually be visible, as will the skull and cortical matter. The extent to which any specific membrane is visible is heavily dependent on the imaging protocol.

A further complication arises from the complex geometry of the brain surface — Figure 5.2. Because of cortical folding, the local image intensity neighbourhood may vary quite dramatically from point to point on the cerebral surface.

These problems suggest that some sort of boundary modelling process is required to achieve a suitable level of robustness in our surface detection. The next section examines several possible modelling strategies. Consideration of the impact of PVE on this task is deferred until Chapter 6.

5.2 The Boundary Model Framework

The boundary model must enable us to distinguish the local boundary of interest from other, spurious, structures. Before we can define the model, we require a clear definition of the surface we wish to extract. Because of the geometric complexity of the cortex, the ability to detect sulcal structure within sparse data is of questionable use and desirability. The generally

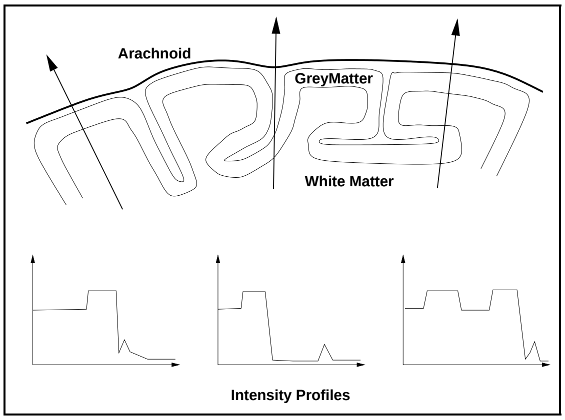


Figure 5.2: Cortical Folding The cortex has a very convoluted structure which ensures that the large-scale boundary structure varies dramatically across the cerebral surface. The figure shows 3 different grey level intensity profiles sampled normally to the cerebral surface. Note how the complex sulcal penetrations induce marked differences in the intensity profiles, all of which are nominally representing the same collection of surface points.

poor correlation between sulci over neighbouring sparse slices poses difficulties for the surface fitting procedure, which must interpolate the available data smoothly. For this reason, we have decided that the *arachnoid* is the most representative boundary for the description of the brain surface. Unfortunately, in many instances it is not visible, blending into surrounding meningial structures. The model should aim to extract as close an approximation to the arachnoid as possible; at the very least, it should not permit edge points to be detected within sulci. Furthermore, since the contrast in an image differs significantly from protocol to protocol, the model representation should be loosely coupled to the imaging protocol, to facilitate the construction of a generic approach.

5.2.1 Model Input Data

At the core of our proposed modelling process is a 1D vector of intensity values, extracted by sampling the volumetric image along a ray emanating from the mesh surface. Each vector specifies a 1D signal which may be processed by standard methods to detect boundary/edge transitions. The choice of a 1D intensity neighbourhood, rather than a 2D or 3D neighbourhood, was motivated primarily by computational economy. Such a strategy was also used for our preliminary work with snakes — Chapter 3.

In the sparse framework, these intensity samples are constrained to lie within the individual image slices which comprise the 3D volumetric image. Within the slice, the intensity sampling errors are significantly smaller when compared with those which arise when full 3D line sampling is performed. By only using data within each image slice, we provide more accurate input information for our models.

During the boundary detection phase, intensity vectors for selected points on the surface mesh are extracted. The model is applied to each vector, yielding a set of 3D boundary points which are used to update the model — see Chapter 7.

5.2.2 The Models

We have implemented three kinds of models:

M1. Edge Heuristics The simplest approach — a set of heuristics for processing gradient and grey scale information is provided which attempts to locate the desired gradient from those present. Only relative gradient strengths and grey-level comparisons are used.

M2. PCA Models During the mesh template training phase, 1D grey level information from each corresponding point is used to build an average surface-normal intensity profile with associated modes of variation. This is a variant of the 2D scheme implemented by Cootes et al [23]. In this scenario, grey level information is used directly, which means that intensity normalisation must be applied to the images. Furthermore, separate models are required for each standard protocol (T1, T2 and PD).

M3. Specified Models Using knowledge of anatomy, we may construct a model (or a set of them) which encapsulate the variability we know to be present. The model is represented as a piecewise constant curve, in which the individual segments have variable length. Because the characteristic grey level signature of a given boundary type varies with protocol, a separate set of models may be required for each.

The requirement that sulci be ignored during boundary detection will not be directly incorporated into the boundary model. Instead, a scheme will be derived whereby boundary points which deviate dramatically from points known to lie on the "arachnoid" (or our approximation to this surface) will be rejected. This issue will be fully addressed in Section 5.6.

5.3 Edge Heuristics

This scheme is ideally suited to Zone-based boundary detection: the heuristics required are manually specified. If one were required to produce a different set of heuristics for each node, the amount of work required would be excessive.

The model is built as follows:

1. A number of representative 1D grey-level profiles are subjected to a 1D Canny edge detector (of a specified scale)

2. The resulting set of gradient maxima locations and magnitudes, $\{x_i, G(x_i)\}$, provides a disjoint decomposition of each intensity profile; the intensity values are averaged over each pair (x_i, x_{i+1}) to provide an average intensity for each edge separated interval, $\{\overline{g}_i\}$

3. The set $\{x_i, G(x_i), \overline{g}_i\}$ is then examined for obvious algebraic relationships. For example, if we had a step edge going from bright to dark, we could say that, given a gradient (edge point) candidate, if the pixels to the left were (on average) brighter than those to the right, that this point matched our model. If we have a series of alternating gradients, then we may mobilise the gradient information to filter out spurious edge point candidates.

We only constructed two models: a model for the cortex-CSF boundary and one for the mid-saggital line. The MSP detector was initially taken to be a valley detector, since this is an intuitive representation for this feature. Unfortunately, in many instances this line is NOT visible (recall Figure 3.5 from Chapter 3). Furthermore, once the brain hemispheres fuse, the line will vanish.

The cortex-CSF model represents a bright region followed by a dark region, followed by a bright region, corresponding to cortical tissue, CSF and skull on a T1-weighted scan. The 3 largest gradients are selected (output from a 1D Canny edge detector), and if the edge contrast is correct, and the average grey level characteristics are consistent, a given edge point is flagged as the boundary point. To compensate for the possible exclusion of the skull from the intensity sample, a rule is coded which will enable a step edge (representing the cortex-CSF boundary) to be detected. The gradient must be of the appropriate polarity, and should be the largest present. Some examples are shown in Figure 5.3.

There are numerous problems with this simple approach. In particular, the scale at which edges should be detected is not at all clear. As the imaging plane intersects the tissue boundary at an increasingly acute angle, cosine foreshortening comes into play, and the edges are stretched or compressed. A scale appropriate to the "equatorial" region is likely to be highly inappropriate for the "polar" region of the cerebral hemispheres. Furthermore, localised intensity inhomogeneities can cause pathological gradient maxima which confound any set of simple heuristics. There is also a definite dependence on the imaging protocol, since the heuristics require knowledge of edge contrast and average grey level characteristics.

Finally, the lack of a suitable means to detect points on the mid-saggital lines is undesirable: the brain often possesses a notable curvature of this supposedly planar structure. To accurately represent the data, we need to detect at least some of these points.

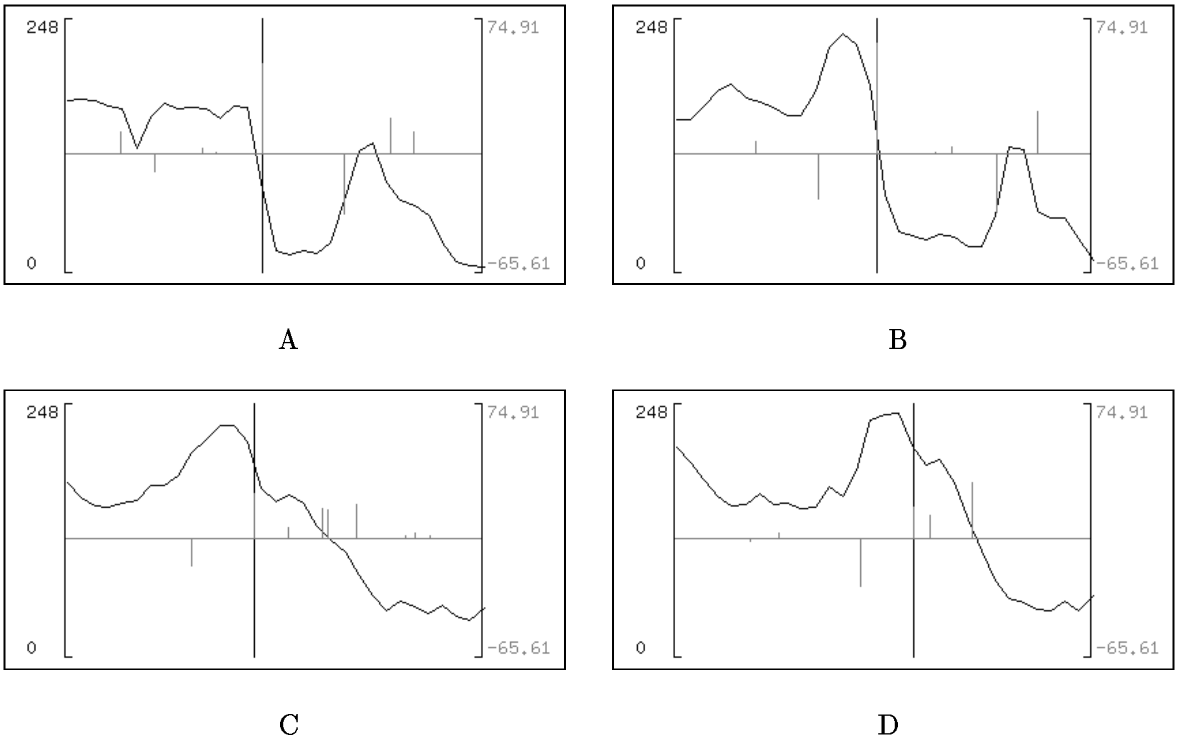


Figure 5.3: Edge Heuristics A, B) The profiles shown correspond to the cortex-CSF-skull boundary region. The vertical line indicates the edge point chosen out of the available gradient maxima (light grey vertical lines). At this location, the average grey value to the left is larger than the average grey value to the right, and the gradient has the correct sign when compared to its two largest neighbours (together these three gradient maxima have the largest absolute magnitudes). This simple rule works well in regions where the boundary is such that the three largest gradient maxima are well separated from the others, and if the edges are fairly smooth. C, D) When the gradient maxima are not well differentiated, a less satisfactory choice may result. The gradient maxima generated by the 1D Canny edge detector are quite sensitive to local edge perturbations.

5.4 Node Modelling

Since the heuristic scheme is plagued by a number of problems, a more generic scheme was sought. The method employed by Cootes [23] in 2D was extended to 3D and a grey-level model was constructed for the most common protocol (T1) in our data set. The grey-level samples used to construct each node-based model are sampled along a ray normal to the mesh surface and emanating from the node under consideration. By extracting profiles for each corresponding node over our segmented training set, we construct an input sample sequence which can be processed directly to yield a PCA. The process by which corresponding nodes are defined was discussed in Chapter 4.

The model consists of a mean grey level intensity profile — representing the average intensity along a ray normal to the mesh surface at that point — and associated modes of variation. This information is used to seek matches in the underlying volumetric image. A matching function, M_T, is defined and the vector of intensities under consideration is compared to the learned intensity model. A number of different intensity sample vectors are selected and tested: the one which provides the most satisfactory match indicates where the model node should be relocated. A *Mahalanobis distance* was used as the matching function, with a requirement that the matched profile lie within 3 standard deviations (3 s.d.) of the mean (by comparing the basis coefficients to the eigenvalues). The Mahalanobis metric reduces to an expression in the modal coefficients and eigenvalues of the PCA system — see Appendix D. The "3 s.d." criteria serves as a mechanism to reject obvious outliers (profiles which do not correspond to those observed in the training sequence).

The matching process is shown in Figure 5.4. The intensity sample is treated as a signal, and a fixed length window is translated along this signal. At every translational step, the 3 s.d. criterion is evaluated, and M_T is computed if this test succeeds. The windowed signal which minimises M_T is considered to be the best match. Because the PCA is constructed by sampling symmetrically about the surface, the centre of the matched window indicates the location of the desired boundary.

When we are using sparse data, only in-plane grey level information is available. In this case, we must devise a means of transforming the pixel values to enable the learned model to be appropriately employed. The mechanism at work in this case is cosine foreshortening (which has been discussed previously).

If we approximate the surface locally as a plane, we may compute an angle of foreshortening and transform the in-plane intensity data into the node-normal frame — See Appendix E. However, the points from which we sample are no longer mesh vertices, but the intersection points of the mesh edges with the image sampling plane. For any such point, we may establish

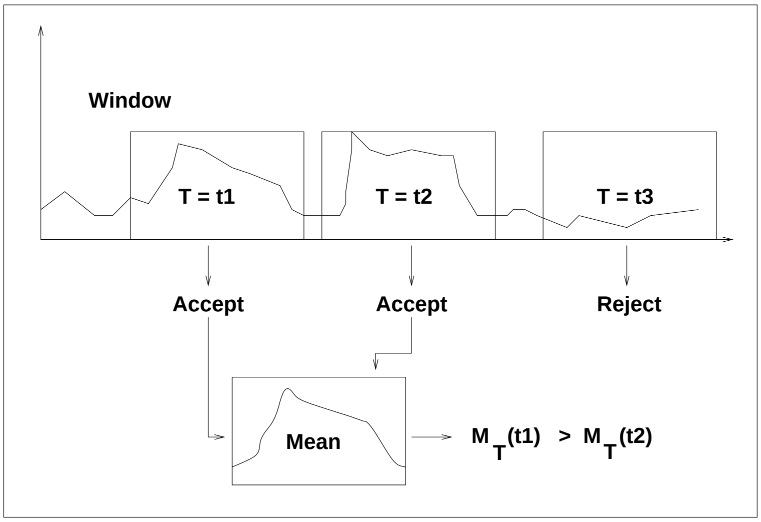


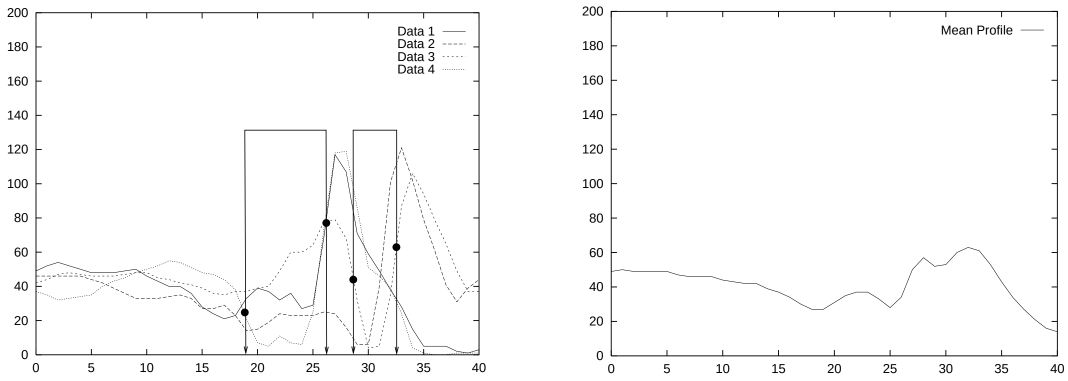
Figure 5.4: Matching Under the Mahalanobis Criterion The intensity sample may be viewed as a signal. We define a window, with a sample size equivalent to that used to define the PCA vectors, and evaluate $M_T(.)$ at various translational offsets along the signal. If the windowed signal is further than 3 s.d. from the PCA mean, it is rejected immediately. Otherwise, the Mahalanobis matching function is evaluated and the windowed signal which minimises this distance is chosen (located at $T = t_1$ in this example). The centre of this matched window indicates the location of the target boundary.

the closest mesh node, and hence the model which is most appropriate for matching local intensity data. By applying the algorithm stated above to the transformed data, we may find the best in-plane translation for each boundary point. The set of these planar translations may be utilised in the model update process discussed in Chapter 7.

The node modelling process did not yield very satisfactory models. As Figure 5.5 shows, the averaging of profiles from a number of training profiles leads to unrepresentative grey-level models. This is particularly evident in Figure 5.5B) which shows the mean grey-level model generated from profiles extracted near the top of the brain. Since PVE is maximal in this region, and the model is to be mobilised to clean up the data, this is a particularly unsatisfactory finding. Examination of this figure shows several problems with this automated process:

- intensity curves may be translated w.r.t. one another,
- some curves appear to represent very different structures.

As we stated in Section 5.1, even if precise correspondences are available, the complexity of the gyral structure near the brain surface can give rise to very different intensity profiles. It is clear from these graphs, that the correspondences are not precise, since some of the curves seem to be approximately related by a translation. Unfortunately, one cannot simply "register" these curves, since there will be non-linear warping in addition to this linear transformation (tissue widths can vary significantly across patients).



A B

Figure 5.5: Profile PCA A) Because of local differences in brain geometry, the intensity profiles extracted at points considered to be correspondents can vary considerably. The boundaries delimiting the cerebral and inner skull surfaces have been bracketed for two of the profiles. The approximate location of the edges are indicated by solid circles. Observe that the curves cannot be properly registered using simple translation. B) The mean profile is only vaguely suggestive of the underlying anatomy and is significantly reduced in intensity.

As a consequence, the grey level PCA can admit solutions which bear little relation to the underlying anatomy — Figure 5.6. It seems that the only suitable means by which we may impose a sensible grey level model is to mobilise our knowledge of boundary anatomy to directly specify the models of interest.

5.5 Specifying a model

Before describing the process by which these models may be defined and applied, we will summarise a number of important properties concerning the boundary structure we are interested in.

1. The idealised intensity profile sampled along a normal ray emanating from the cortical surface yields a piecewise constant curve (in the absence of noise, imaging artifacts and PVE).

2. The widths of the individual piecewise constant segments (which represent various tissues/fluids) will, in general, exhibit very significant inter and intra subject variability. They are dynamic entities.

3. In general, the relative intensity relationships between tissue types will be preserved for a given protocol. For example, in a T1 weighted scan, grey matter will always be brighter than CSF.

4. The ordering of tissues is generally preserved. Provided the intensity profile is sampled consistently (for example, along the normal ray pointing away from the brain surface), one will always find that cortex precedes CSF, which precedes skull etc. Note that the

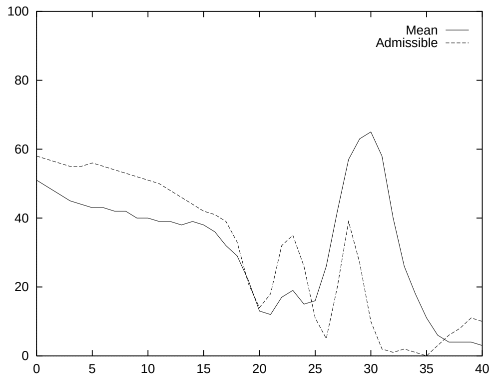


Figure 5.6: Poor Boundary Model Significant differences in brain surface geometry across patients may yield an unrepresentative PCA. In this case, the mean of the profile training set — which is unrelated to the data shown in the previous figure — is quite different from the shown profile instance. One would expect that this particular profile would fall beyond the bounds of the profile PCA space. However, under the 3 s.d. criterion, the example profile is "admissible".

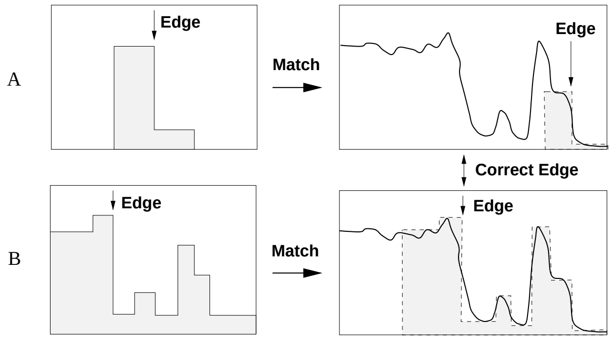


Figure 5.7: Matching with a Dynamic Boundary Model If a simple step edge is taken as the boundary "model", then any edge with the appropriate contrast may provide a suitable match. The introduction of a dynamic boundary model, which may be matched non-rigidly, allows for a more robust boundary detection procedure. The piecewise constant model provides an idealised description of the 1D boundary neighbourhood, and is understood to be flexible, in the sense that each piecewise constant segment may be stretched/compressed or adjusted vertically to enable a better fit, provided that the model constraints are not violated.

grey and white matter are folded in complex ways, so that one cannot make the same observation about their relative ordering in a 1D sample.

These observations led to the construction of a piecewise constant boundary model, in which each constant segment reflects a non-zero width of tissue (or fluid). The function value associated with each segment is chosen to represent the intensity characteristics of that tissue under the given protocol. These values are not taken to be absolute. Rather, they are intended to suggest the relative intensities of each tissue type — the model will be scaled appropriately for each new subject. Similarly, the relative lengths of the various segments are assumed to reflect the typical distribution found within the given boundary type. It should be recalled that these segments are dynamic lengths, and that the model should be viewed as a dynamic entity which is suggestive of the boundary to be modelled. Such a model is shown in Figure 5.7, which also shows how the richness of the model description serves to disambiguate potential mismatches.

5.5.1 Model Construction and Representation

The models are built using the following procedure:

1. The number of distinct boundary models is first established. Knowledge of anatomy can be used (in the case of medical data) or a number of boundary intensity profiles can be examined from each image. Intensity profiles with the the same arrangement of tissue bands, but having different tissue widths, are described by a single model. The number of tissue bands can be estimated visually (the intensities should be near uniform).

2. For each boundary model, the intensity relationships between each piecewise segment are noted. If one tissue is always brighter than another, the relative heights of the two segments will reflect this relationship. Under the currents scheme if two segments have multiple intensity relationships, a new model will be created for each permutation.

3. The intensity relationships and segment widths are directly reflected in the shape of the piecewise constant curve representing the boundary model. The model does not encode absolute width or intensity, however. During boundary detection, the model will be vertically scaled, whilst the segment width are considered to be mutable.

Each model is stored as a sequence of numbers, representing the intensity values of the piecewise constant segments. The sample length of each piecewise segment (which naturally consists of identical numbers) is proportional to the width of the corresponding tissue band. During matching, these segments are allowed to expand or contract, so the number of samples may be deduced by looking at a representative intensity profile. The edge of interest is unambiguously specified by storing the index of the appropiate edge transition (for example, the transition between white and grey matter segments). Constraints on the extent to which the models may deform are introduced in Section 5.5.6.

5.5.2 Matching the Model to Data

Given a candidate intensity profile, the model must be deployed to detect the boundary of interest. The model is stored as a discrete, piecewise constant profile, with a label which identifies the transition which corresponds to the boundary edge of interest. If the model represented a boundary with fixed widths, then a simple correlation or least squares error criterion could be used to fit the model to the curve (after appropriate scaling). The location

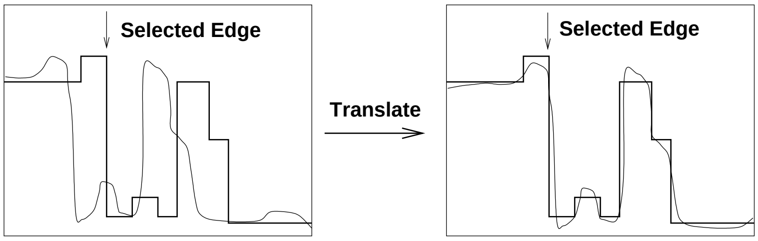


Figure 5.8: Simple Translation In the simplest case, the boundary of interest would have a static profile, and a simple matching scheme based on model translation would suffice. Since the edge of interest within the boundary model is known, the location of the appropriate edge transition within the matched boundary model can be used to infer the location of the edge within the signal.

of the tagged edge transition would then identify the best approximation to the edge point within the 1D intensity array — Figure 5.8.

Of course, this situation is unlikely to occur in practice, as we have already observed. To provide the level of flexibility we require, a local fitting scheme is required — one which will maintain the required segment ordering and intensity relationships. This is not an easy undertaking: there is no obvious minimisation procedure which will yield the best piecewise constant fit to an arbitrary curve. It is tempting to generalise the case of simple translation into a full affine transformation, W,

$$W(M(x)) = M(\alpha x + T) \tag{5.1}$$

where T is the translation of the model, M, and α is a scaling constant. This would permit a uniform compression or dilation of the model, in addition to a translation. While this would provide an additional degree of freedom, the transformation W imposes an unnatural constraint on the flexibility of the tissue segments.

This objection suggests that a non-linear matching algorithm is required. Conceptually, the non-linearity we seek should permit complete freedom for model deformation, subject to our constraints, namely:

1. The piecewise constant form of the model must be preserved.
2. The tissue/segment ordering must be preserved.
3. The intensity inter-relationships between segments must be preserved.

Our matching strategy is built around a coarse-to-fine approach. The model is first decomposed into a sequence of local "model elements", representing "prominent" structures within the piecewise constant profile. These are fixed groups of adjacent segments which will be

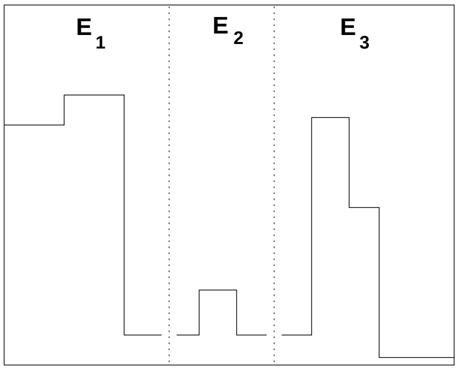


Figure 5.9: Model Elements The boundary model may be decomposed into a series of disjoint model elements. These model elements provide the basis for coarse scale matching. The dotted lines indicate how this model will be decomposed into three model elements, E_1, E_2 and E_3. Each of these elements is rigid, but the distance between them may be stretched or compressed.

used in the initial stage of the matching process. This decomposition provides a means of making the dynamic lengths separating these structures explicit, Figure 5.9.

Intuitively, one would expect that an initial non-rigid matching based on these individually rigid model elements could be used to facilitate a localised refinement process. Indeed, this is the approach we have adopted, as illustrated in Figure 5.10. The mechanism by which suitable model elements may be generated is now discussed.

5.5.3 Model Decomposition

We desire a decomposition which breaks the model along the most prominent "valleys", since these are likely to delimit model regions which can be treated as integral units for coarse scale matching. We decided to use a "watershed" algorithm. In this interpretation, the model is viewed as a landscape which is completely flooded. As the water is drained from the landscape, the fluid level gradually drops, exposing progressively more of the landscape. These land-masses get progressively larger as the water level recedes, finally merging into one when all the water has drained away — Figure 5.11A). At any point during the drainage process, one can define a "watershed" to be the highest land-point separating two water-filled valleys. We, however, are not interested in these watersheds, but in the centres of these flooded valleys — these points provide possible boundaries for model decomposition.

The criterion we use for selecting the water level is simply to find the lowest level at which the land-mass count is maximal. For the kinds of models we are interested in, this criterion seems to provide good results — Figure 5.11B). Observe how the chosen water level splits the model into structures which appear to have intuitive meaning (skull, cortex, meningial tissue/CSF). Sub-structures within the individual "masses" are ignored, as is proper: the model elements are intended for coarse matching only. Although the criterion should properly contain some

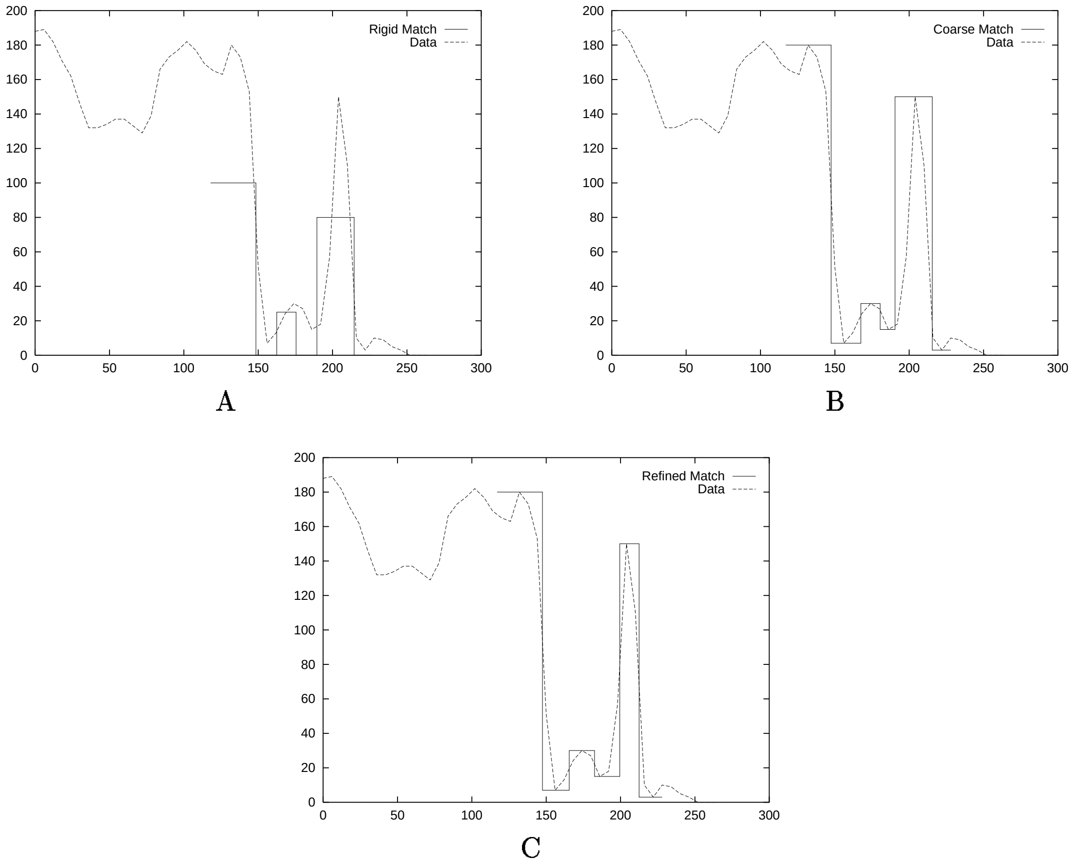


Figure 5.10: Coarse to Fine Matching A) If matching is restricted to a rigid translation, then a poor match will be obtained in many cases. B) By utilising model elements, a better fit is obtained for a wider class of boundaries. C) If the model elements are then refined locally, the final solution should provide a good estimate of the underlying data.

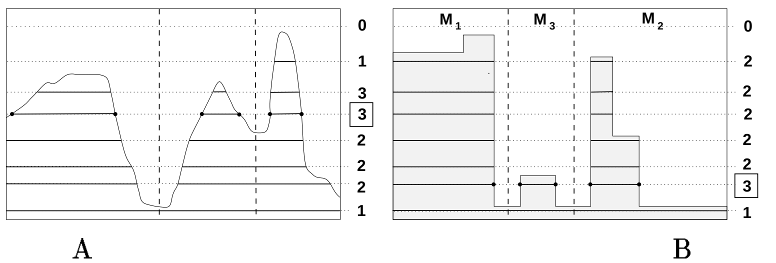


Figure 5.11: Model Decomposition A) We assume that our 1D function represents some flooded, mountainous terrain. As the water level is lowered, progressively more and more terrain is revealed, until it all merges into one mass (the ground). The horizontal dotted lines indicate the water level, while the numbers to the right show the number of "land-masses" exposed at that level. Because we wish to split our model on significant "valleys", we decrease the water level until the point at which no additional land-masses are exposed: we assume that those which have been exposed at this point are the largest. The level at which this occurs is is indicated by the boxed number. The model is split along the centre of each valley separating the land-masses. B) An example of this algorithm applied to one our our models. The labels M_1, M_2 and M_3 indicate the order in which the model elements will be matched.

notion of scale, to avoid the detection of small valleys, it is well suited to the simple piecewise constant profiles we wish to decompose.

5.5.4 Coarse Matching

The coarse scale matching process is itself based on a coarse-to-fine strategy: the *largest* model elements are matched first. We have adopted a very simple metric to quantify model element size: the area bounded by the piecewise constant curve within each element, see Figure 5.11B). Furthermore, the matching process must maintain the spatial ordering of the model elements, since anything else is anatomically incorrect. The matching energy is taken to be the sum of squared differences. Other difference measures, such a cross-correlation, could also have been employed. Under this difference measure, the fitting process for each model element reduces to finding the 1D translation which yields the smallest residual. As stated above, the elements are matched from largest to smallest and the spatial ordering imposed by the original model is enforced.

The algorithm is as follows:

Algorithm 5.1 (Coarse Matching)

1. Smooth the input data signal, $S = \{s_i\}_{i=1,\dots,N}$, with a unit variance Gaussian, to yield S_G.
2. Scale the model data, M, so that the model maximum coincides with the that of S_G.
3. Decompose the model using the watershed algorithm, to yield ordered model elements, $\{E_1, \dots, E_P\}$.
4. For each E_i, compute the model size, and thus define the order in which the E_i will be matched.
5. For each E_i, in size order:
   - 5.1 If any models have been matched to the left, set the leftmost translation limit to the right edge of the relevant matched model, L, otherwise set $L = 1$.
   - 5.2 If any models have been matched to the right, set the rightmost translation limit to the right edge of the relevant matched model, R, otherwise set $R = N$.
   - 5.3 If the new model element is too small to fit within $[L, R]$, terminate matching.
   - 5.4 For each translational position in $[L, R]$
     - 5.4.1 Evaluate the sum of squared differences residual.
   - 5.5 Select the translation position corresponding to the minimum of the residual set.

Statement 5.3 in the algorithm raises a very important issue: the role of the initial model scale. While the model is flexible, the size of the model elements imposes a lower limit on scale of the features which may be matched. Coarse level matching only permits the space between model elements to expand or contract; the model elements themselves are rigid. Consequently, if two matched structures are very close to one another, then small scale detail may be missed between them. This problem may be overcome by defining a model in which the affected structure is appropriately scaled down. The whole issue of model size is intimately related to the issue of discretisation, and will be discussed fully in Section 5.5.8. Some examples of this kind of matching are shown in Figure 5.12.

Observe that although the overall match is improved, when compared to simple translation, the edges do not generally fit the data very well.

5.5.5 Local Matching

If the coarse matching has succeeded, then both the edge estimates and the piecewise constant fit need to be locally refined. The edge refinement process is accomplished by locally translating the relevant step edge and noting the variation in the (sum of squared differences) residual. We shall assume that an "edge point" is located at the centre of a non-ideal edge. This assumption is reasonable except in the presence of large PVE — in this case, as we shall show, the edge point may be perturbed from this central location. It should be noted that, under this definition, the edge location might not correspond to a point of inflexion on the local curve.

We justify this assumption on the following grounds:

Simplicity If one only has a discrete 1D signal, and a rough idea of the height of the diffuse edge, then the edge point may be estimated by translating a step edge of the appropriate height and contrast along the signal and noting where the residual between the signal and the step achieves its minimum.

Foreshortening Effects In the absence of noise or imaging artifacts, a piecewise constant boundary profile will retain its ideal edges, except when the underlying tissue is sampled obliquely relative to the surface normal. This results in (to first order) cosine foreshortening, which linearly scales the distances between the edges. If one assumes that the edges are slightly imperfect, and could be approximated by a very steep ramp, then it would be reasonable to take the edge point as being in the centre of this ramp. The ramp would also be subject to linear separation, but the edge point would retain its central position. These foreshortening effects are clearly visible in much of the intensity profiles we have to process. A discussion of foreshortening may be found in Appendix E.

With these comments in mind, we shall define the optimal edge location as one at which the sum of squared differences between a local step edge (of size $2R$ samples) and the underlying signal achieves its minimum. The size of R will be addressed in Section 5.5.8. When

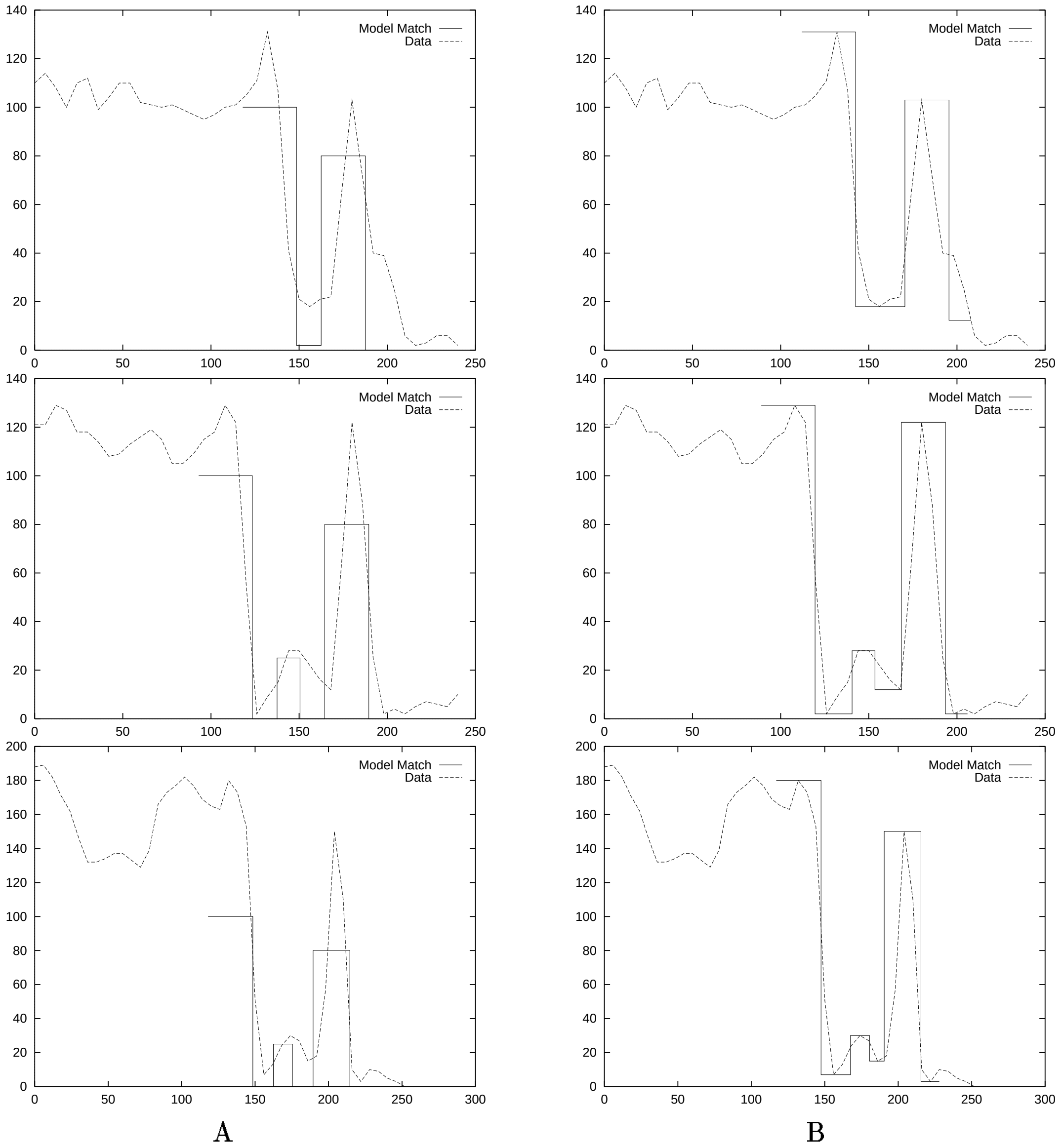


Figure 5.12: Coarse Matching A) When the model is scaled vertically and matched under translation, the fit is often poor. Note that model has not been shown as scaled in these diagrams. B) If we use the coarse matching scheme based on model elements, the fit is generally much better. The model envelope has been estimated after element matching, but even if this were not so, the edge estimates are clearly improved over the the rigid case.

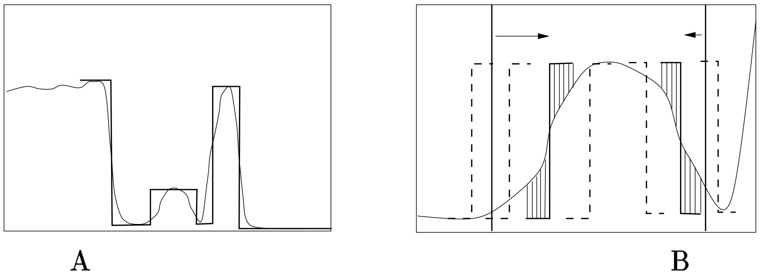


Figure 5.13: Local Matching A) The initial coarse level match is shown. Note that while the overall shape is correct, the edges are not very well placed. B) The two thick vertical lines represent the initial edge approximation. The estimates are refined by shifting the appropriate step edge left and right about each of these initial position, and evaluating the difference measure. The arrows indicate how the edges have shifted to minimise the sum of squared differences between the local step model and data. The vertical lines show the absolute differences between the local step and the data at the optimum location.

evaluating various translational possibilities, the step edge is not permitted to bump into its neighbours. Such a localised edge search is shown is Figure 5.13. Observe that the minimum is attained when the local step edge is at the centre of the blurred ramp. Real data will usually have edges which are distorted into some non-linear ramp. But to a first order, this approximation is acceptable. We shall address non-linearities (arising from PVE) in Chapter 6.

In order for the local edge adjustments to provide accurate localisation, the local step edge should be roughly the same height as the edge we wish to fit. The initial coarse matching step does not guarantee that this will be so, since the model height is fixed (after initial uniform scaling). The first step, then, is to refit the initial model match to obtain a more accurate signal representation. The fitting process requires that we compute a new intensity for each piecewise constant segment, whilst ensuring that the model intensity constraints are respected.

We chose to estimate each segment height so that the model provides a bounding envelope for the intensity data, as shown in Figure 5.14B). The alternative would have involved some sort of averaging of pixel intensities of each segment. However, the manner in which we have defined our edge location means that a simple intensity average over a segment will often over or underestimate the data intensity — Figure 5.14A). One can used a "trimmed" average, in which some (peripheral) values are ignored, but choice of a suitable trimming criterion is itself problematic. The use of an intensity envelope is consistent from the point of view of a "first order" model fit, by which we mean one which ignores PVE and other small perturbations. In the ideal case, the only mechanism at work would be cosine foreshortening, which would preserve the intensity of each model segment, providing precisely such an envelope. Blurring will be dominated by PVE, which we shall model separately (Chapter 6).

Unfortunately, when we modify the segment intensities, the best fitting local edge is likely

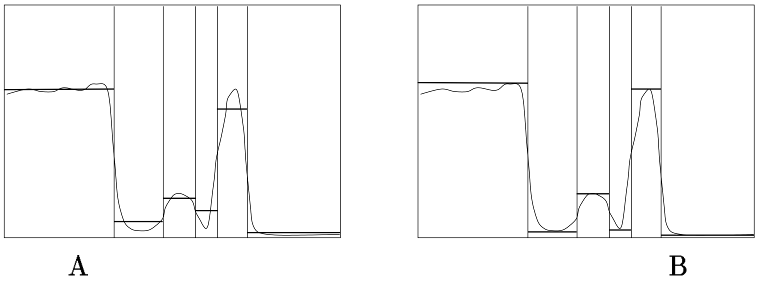


Figure 5.14: **Vertical Fitting** A) When the model has adjusted itself horizontally, the heights of each segment must be updated to better represent the underlying data. If new heights are calculated by averaging the data samples covered by each segment, then the adjusted segment heights may often provide a poor data fit. B) If the maximal/minimal data values within each segment are taken, then a much more satisfying fit is obtained. The decision on whether to use the maximum or minimum data values over a given segment depends on whether the segment is at the bottom of a valley, or at the top of a ridge (established by examining the relative heights of its neighbours). The use of a data envelope of this kind, rather than plain averaging, is also justified on the basis of the various degrading processes which affect the data.

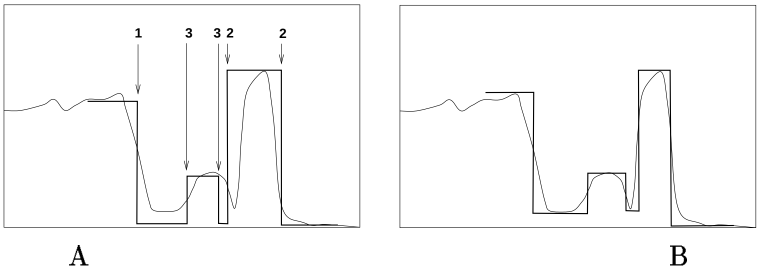


Figure 5.15: Order of Local Matching A) The initial, unrefined coarse match is shown. The numbered arrows indicated the order in which local matching will proceed — where two numbers are the same, they may be matched in arbitrary order. If edges are refined in an arbitrary manner, the refinement maybe trapped by a local minimum. In this case, by matching the edges labelled "2", we allow the rightmost "3" edge to move to the optimal location. If "3" had been matched first, it would have been forced to remain stationary, and an additional iteration would have been required to adjust "3". It is worth emphasizing that such problems may often be corrected by iterating the local refinement algorithm, but this is not always the true. B) The final solution after the the local iterative scheme has converged.

to change too. This suggests an iterative local refinement scheme. At a given iteration, we recalculate the segment heights and then for each edge, apply our local refinement procedure. The edges are matched in an order which is determined by the size of the model element to which they are attached. By processing the edges associated with large scale structures first, we provide additional scope for smaller scale edges to adjust their positions.

This is illustrated in Figure 5.15, where the coarse matching has resulted in a very loose fitting model estimate. If the small scale step edges are matched first, they will bump into the larger, unrefined step edges. This observation is really just a continuation of our coarse-to-fine strategy. Edges within each model element are matched in arbitrary order (although for simplicity we match them from left to right).

5.5.6 Model Constraints

An important issue is the manner in which model constraints are imposed. Because the model is flexible, the only horizontal constraint we enforce is that segment lengths be non-zero. The situation for the relative intensities of the segments is, however, somewhat different. If no constraints are imposed, the model will be able to match almost any signal. We insist that the relative intensities between segments be maintained. There are two further possibilities here:

Loose Constraints Given a set of contiguous piecewise constant model segments, S_i, each with associated intensity, I_i, we may impose constraints of the kind: $I_i > I_j$, for the required permutations of segment indices.

Tight Constraints The alternative is to insist on very specific constraints of the form: $\alpha_{i,j} < I_i - I_j < \beta_{i,j}$, where $\alpha_{i,j}$ and $\beta_{i,j}$ reflect knowledge of expected variation in the heights of the intensity pair under consideration.

There is a strong argument in favour of using tight constraints: they allow us to exercise finer control over the model fit. However, the generation of such constraints requires significant interactive effort, and the chosen constraints may prove too "tight" when applied to noisy data (there may be significant fluctuations in tissue contrast across a single MR image, for example). Loose constraints, on the other hand, are implicitly defined by the model and, if there are a sufficient number of such constraints, the chances of a false match should be accordingly reduced. Both these constraint types are relative, that is, no absolute intensity constraints are enforced. Absolute constraints are of the form

$$\delta_{0,i} < I_i < \delta_{1,i}, \quad i = 1, \ldots, N, \tag{5.2}$$

and provide a means of controlling the extent to which the model may be shifted vertically. Although this seemed of little relevance initially, later investigation revealed that such vertical shifts could result in spurious matches, even with fairly specific relative constraints. If one assumes that the initial model intensities provide a template for the intensity relationships we wish to enforce, then a simple scheme may be used to generate absolute constraints from a user-supplied threshold, ϵ, $\epsilon \in [0, 1]$:

1. Compute the initial vertical scaling, ρ, between the model and the signal.
2. Compute the max of all the scaled model segment intensities:

$$I_{\max} = \max_i \{\rho I_i\}. \tag{5.3}$$

3. $\delta_{0;1,i} = \rho I_i \pm \epsilon I_{\max}$, where I_i is the original model segment intensity for segment i.

Given a particular choice of constraint, how should we impose it? The rigorous approach requires the solution of a fairly straightforward non-linear optimisation. In this scenario, we may define our goal function as the quadratic functional, $E(\{I_i\}_{i=1,\ldots,N})$,

$$E(\{I_i\}_{i=1,\ldots,N}) = \sum_{i=1}^{N}(I_i - I_{i,c})^2, \tag{5.4}$$

where the set of constants $\{I_{i,c}\}_{i=1,\ldots,N}$, represents the estimated segment intensities obtained from (unconstrained) envelope fitting discussed above. The constraints may be recast in the form,

$$\alpha_{i,j}I_i - \beta_{i,j}I_j - \gamma_{i,j} > 0, \quad (i,j) \in \pi, \tag{5.5}$$

where π represents a suitable subset of index permutations.

The number of (relative) constraints is easily computed: each segment shares a single valid constraint with each of the other segments. However, because of the symmetry involved, many of these constraints are redundant, and the true number is $O(N^2)$. In fact, for N contiguous segments, there are precisely $\frac{N(N-1)}{2}$ unique segment constraints (or twice this number if tight constraints are used). The constrained minimisation may be solved using a simple barrier method [57], which reduces to inverting a matrix of size $N \times N$. This typically requires $O(N^3)$ operations.

A less rigorous — and computationally cheaper — approach, is to simply reject infeasible solutions. All one need do is find the best unconstrained match, and then test each constraint in turn. The obvious objection to such a method concerns the question of local minima. One might suspect that finding a constrained solution will provide a best compromise solution, but a solution nonetheless, whereas the rejection method may not produce a result at all. However, the only difference in the two solutions will occur when the constraints are such that the unconstrained fit is not feasible (that is, where the envelope is correct, but the overall shape is wrong). In this case, the model is clearly inappropriate and rejection should therefore be satisfactory. Accepting a constrained solution in this case would result in a piecewise constant curve that, while having the correct shape, did not fit the data (that is, would not generate a "data envelope"). Since we shall be using a database of models for matching purposes, with the express aim of rejecting model mismatches, we have adopted the simple constraint rejection scheme.

5.5.7 Matching for Multiple Boundary Classes

In general there will be more than one *class* of boundary profiles, and within each class the boundary model may need to be refined to provide a better representation of the underlying

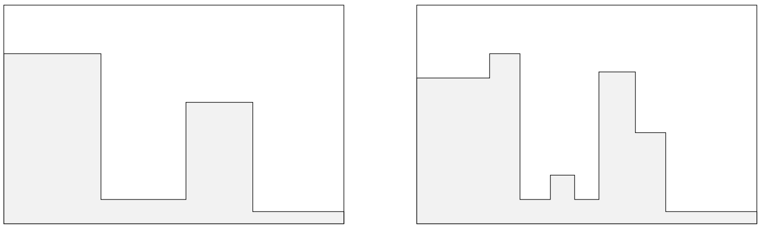


Figure 5.16: Model Refinement A) The simplest model for the cortex-CSF-skull boundary is shown. Because of its simplicity, this model may produce spurious matches. B) One may refine the model by adding tissue segments to represent, for example, the differentiation of grey and white matter. Such a refined model is less likely to produce a false match.

data. For example, in the "brain-CSF-skull" boundary class, the simplest model consists of the model depicted in Figure 5.16A).

This is clearly a very coarse representation of the underlying anatomy. One can add segments to represent the differentiation of white and grey matter and provide for meningial matter within the cortex/brain interval. An additional external layer can be added to the skull segment, to allow for a skin/muscle layer — Figure 5.16B). Using anatomical knowledge, one may build a *model database*, containing each boundary class and its various representatives. Matching is accomplished by fitting each model in turn, and choosing the one which minimises the difference criterion referred to above.

At this point, the issue of efficiency intrudes: if the model database is large, the time required to test each model might be considered excessive. The models within a given class may be matched hierarchically: coarse models would be examined first, and if matched successfully, refined versions could be tested. This would allow one to select a model class quickly, and test only the models within that class. Unfortunately, there may be problems with this approach, depending on the manner in which one defines the classes. For example, if the simple model of Figure 5.16A) is used as the lowest resolution primitive for the cortex-CSF-skull class, the information present may be insufficient to prevent a mismatch: if we use a model which is too coarse, the difference measure may be trapped in a local minimum and yield a spurious fit. On the other hand, if a fine-scale model is required for matching, you may have to test the entire database. Fortunately, because of the intrinsic flexibility possessed by the models, we can make do with a comparatively small number (currently 12 for the T2 protocol), so all may be tested to find the best match.

5.5.8 Discretisation Issues

A practical issue which needs to addressed is the role of discretisation in both the model definition and the matching process. For ease of definition, the piecewise constant model is defined as a vector of intensity values, with constant runs of a given value defining a segment.

The model is matched to another discrete entity: the 1D array of extracted pixel intensities. If the data is sampled at a spatial frequency which is very different from that represented by the discrete model, the resulting model elements may be too large or small to represent the equivalent features in the signal.

The sample size, $2R$, of the local symmetric step edge will determine the accuracy with which edges may be localised (under our edge definition). In the extreme case in which $R = 1$, the step will be represented by only two samples: one having the value of the bottom of the step, and one having the value at the top. Attempting to evaluate the optimum local fit using such a step will almost certainly lead to disaster, since it has no averaging properties and is highly susceptible to noise. On the other hand, if R is too large, the local step might not be able to move at all, since it is forbidden from over-running neighbouring edges. Alternatively, the structure may be smaller than R, in which case we may fail to detect it.

We have a simple way of coping with these issues: we upsample both the signal and model. Although this does not eliminate the problems, it reduces their severity and enables us to utilise other heuristics to choose a reasonable value for R. The rationale behind this is as follows:

- Discretisation problems are primarily caused by undersampling of the continuous signal and/or model.
- By linearly interpolating the input by a factor of L (that is inserting L regularly sampled points on the line joining each pair of adjacent data points) we may increase the sampling rate as much as we wish, whilst preserving the original signal structure.
- By scaling the model in a similar fashion, but ensuring that we do not smear the edges, we may preserve the original spatial/intensity relationship between the signal and the model.
- If we fit our upsampled data with the upsampled model, the local refinement scale, R, can be chosen to provide good matching/averaging characteristics, without having to worry too much about edge mobility etc.

The choice of R was determined experimentally. The model and data were upsampled at a specified rate L, and the smallest reasonable value of R for a given L was recorded. In general, the choice of L and R will depend on the sample length of the signal and model. However, a reasonable heuristic seems to be $L \geq 2R$, since this will ensure that the step size is roughly the same as the interval between two samples in the original signal. For the model and signal lengths we used, $L = 5$ and $R = 3$ were found to be adequate.

5.6 Dealing with Cortical Folding

In order to make the boundary model as generic as possible, no specific mechanism was included to reject boundary points detected within cortical folds.

We have implemented a rejection scheme based on the comparison of neighbouring edge candidate points. A number of *validation profiles* are extracted within the neighbourhood of our putative edge point, and edge points located within each. By examining the distance between each edge candidate and the surface, a decision can be made as to whether the point is valid or spurious — Figure 5.17.

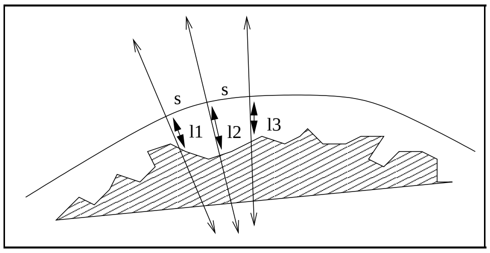


Figure 5.17: Validation Profiles In the figure, l2 is the putative distance to the edge, whilst l1 and l3 are the distances found by the validation profiles. A weighted sum of the difference between these distances is used to construct a confidence measure for the test profile edge candidate.

The validation profiles are located at uniformly spaced arc-length intervals from the origin of the test profile, and are oriented along the boundary normal at that point. The arc-length interval is chosen to reflect the locality of the validation search — see the discussion below. A distance interval, $\tilde{D}$, based on weighted distances, is then computed and compared with the specified distance threshold, D. If $\tilde{D} \subset [-D, D]$, the candidate edge is accepted. The use of an interval, rather than a single number, is intended to account for natural deviation within the set of distances. A single number (the mean of the set of distances, say) might fall within the permissible range, while the variance associated with the distances may fall outside of the interval. In this case, expanding the distance measure to reflect this will ensure that a match is less likely, since the estimated interval will almost certainly fail the containment test.

The weighted distances are intended to reduce the effect of validation profiles which are distant from the test profile. The weights are taken to be $\frac{1}{s_i}$, where s_i is the "normalised" arc-length from the origin of the test profile. In this context the normalisation means that $s_1 = 1$ i.e. the first sample is always at unit distance; subsequent samples are expressed in terms of this unit interval. This ensure that we do not incur a distance penalty when $N = 1$: the first sample is assumed to be sufficiently close to the test profile to eliminate the need for such a weighting. For this work we always take $N = 1$ and set the arc-length step to be 1mm.

The algorithm is as follows:

Algorithm 5.2 (Boundary Point Validation)

1. Extract a test profile at the point under consideration.
2. Extract N validation profiles on either side of the test profile.
3. Compute the best edge point estimate for each profile.
4. Compute the signed distance, Δ_i from the origin of each profile to the corresponding edge location. The sign indicates which side of the boundary curve the edge is located on (the sign of the test edge is always taken to be positive).
5. Construct the distance interval: $\tilde{D} = [\overline{D}_w - \kappa\sqrt{\sigma^2}, \overline{D}_w + \kappa\sqrt{\sigma^2}]$, where κ is a constant which we take as 1, $\overline{D}_w$ is the weighted average distance, and σ^2 is the variance of the weighted distances. The weighted distances are defined as: $\delta_i = \frac{1}{s_i}||\Delta_0 - \Delta_i||$.
   The mean, $\overline{D}_w$, and variance, σ^2, are computed as per normal. Δ_0 is the (positive) distance from origin of test profile to the prospective edge point.
6. If the interval $\tilde{D} \subset [-D, D]$, then the edge point is accepted.

The algorithm is also applicable to 3D edge searches. In this case the validation profiles would be centred on a set of surface points, arranged about the origin of the test profile (a symmetric distribution, for example) and the associated profiles would be directed normal to surface at the corresponding point.

The rejection mechanism will also filter out obvious mismatches resulting from the boundary detection stage. Such mismatches may occur if the profile contains structures not accounted for in the model database.

5.7 Boundary Detection Results

The boundary detection results were obtained by running the segmentation scheme through the initialisation phase, and noting the locations of detected boundary points. The full segmentation scheme is introduced in Chapter 7. It should be noted that the surface which results from initialisation will not provide optimal conditions for edge point detection, since it may be quite distant from the boundary of interest over certain regions.

5.7.1 Model Setup and Parameters

We built a model database consisting of 12 piecewise constant models. This number could have been reduced if a more efficient model representation had been devised. The models used are shown in Appendix F. They were chosen to reflect the boundary profiles within the brain MR images we wished to segment. The two coarsest models were included to address the issue of partial boundary occlusion, discussed in Section 5.8.

The size of the profile sampling window was static, and set at 45mm, since this enabled us to cover the boundary in most cases when the mesh surface was sufficiently close. If the window is significantly larger, then we may detect the other side of the brain, which will also provide a valid boundary response. The development of strategies to cope with this problem is considered to be a subject for future work.

The rejection mechanism was enabled, with $N = 1$ (one validation search on either side of the putative edge), $\kappa = 1$ and $D = 2.5$mm. These values were chosen because they gave reasonable results over the data sets we examined.

5.7.2 Results

The results presented in this section illustrate the range of boundary variation which the models encompass. After the conclusion of the initial template alignment phase, the boundary points along the mesh slice intersection curves were recorded. In Figures 5.18 and 5.19, a number of slices from two quite different brains are shown. Note that the extracted boundary points are essentially correct, although only a few points are identified in the top-most slice, and the mid-saggital line is missing. A subset of the edge models which are matched at various boundary points are shown in Figure 5.20. Note the range of variation which a given model can represent. Although many boundaries present complex profiles, the model is usually able to make a sensible match.

5.8 Discussion

For the majority of slices, the boundary models provide good edge estimates. However, in the uppermost slices, where foreshortening and PVE are maximal, the extracted boundary points are less numerous. Examination of some of the profiles — Figure 5.21 — shows that these curves are punctuated by large, sharp peaks, corresponding to CSF which has penetrated the sulci. In addition, there are sizeable dips in signal intensity which correspond to sulci which are filled with other matter. To find the boundary in this clutter is very difficult. Only a very coarse match can usually be obtained, and in many cases this match will be rejected because the validation profiles find very different edge candidates.

The inclusion of the two coarse models referred to above allows us to detect boundaries even when they are only partially covered by the sampling window. There are fairly strict constraints on the permissible variation of these coarse models. This is necessary to prevent them matching other, undesirable structures.

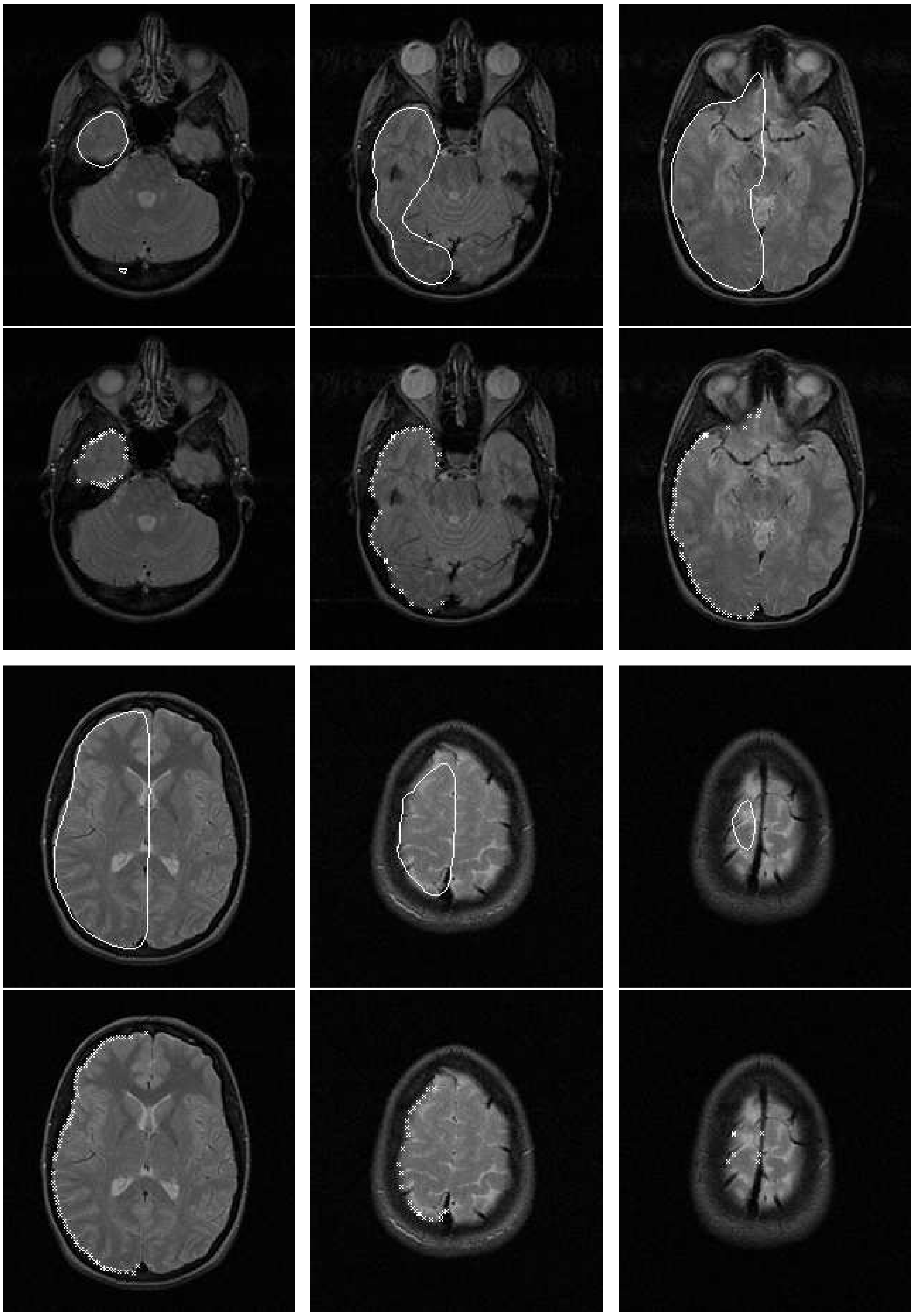


Figure 5.18: Edge Detection - Example One The mesh template was iteratively aligned with the detected boundary points, under the affine ICP scheme. The intersection curve is shown on each slice. These curves are used to define the origin and orientation of the 1D window used for the local boundary search. The detected boundary points from the last iteration are shown as x's. Observe that, although the matches are generally correct, the detection of points is dependent on the orientation and location of the boundary curve relative to the underlying surface. Under the full segmentation scheme, local mesh deformation is permitted, and the boundary curve is able to better fit the underlying surface. This will ensure that the number of detected boundary points increases as the segmentation progresses.

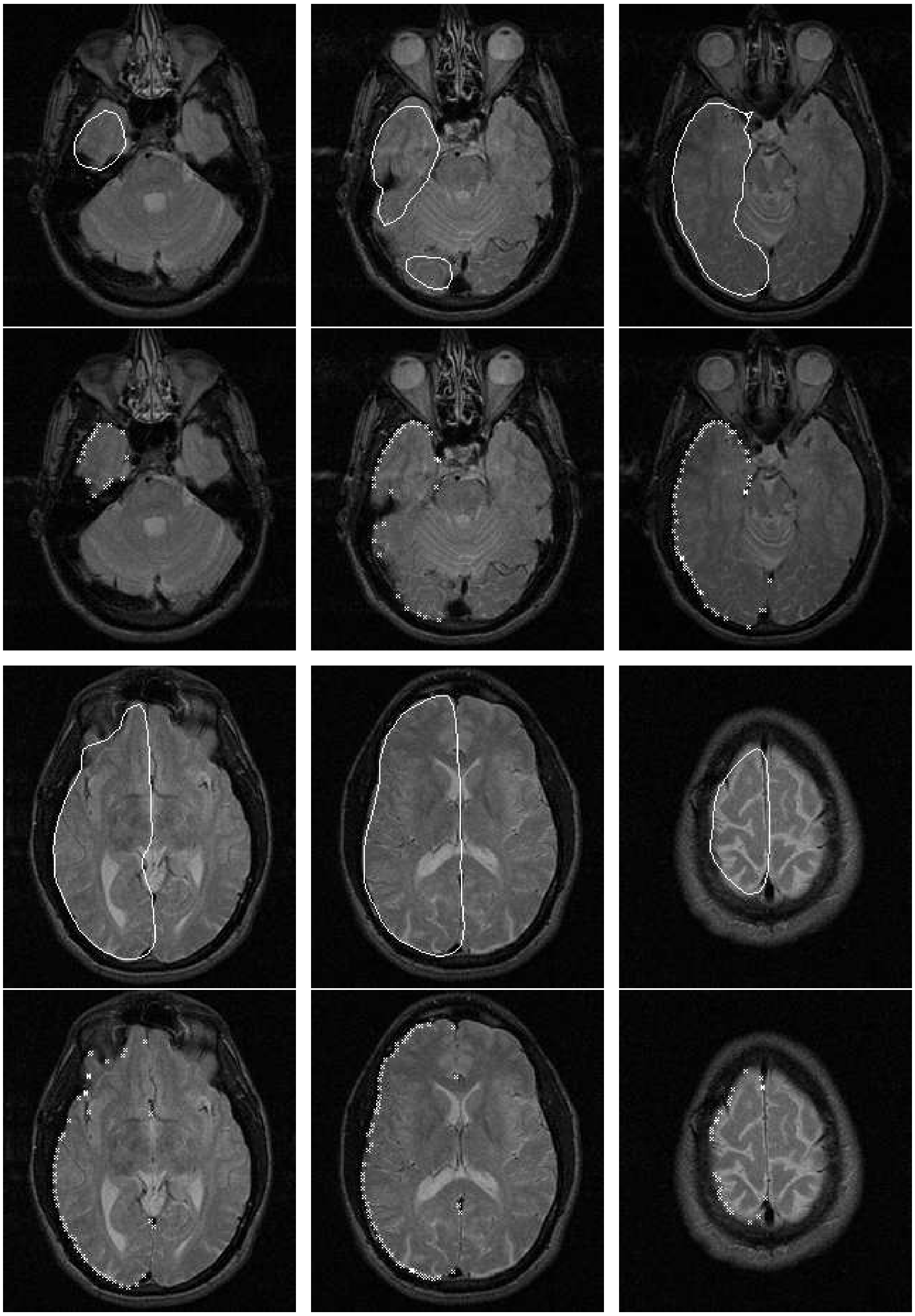


Figure 5.19: Edge Detection - Example Two The location of boundary points is dependent on the relative alignment of the mesh-slice intersection curve and the physiological boundary. In several instances, it is clear that these are not very similar, and no matches are returned, since the rejection mechanism is active. Note that when obvious valleys are present along the mid-saggital line, they may be detected.

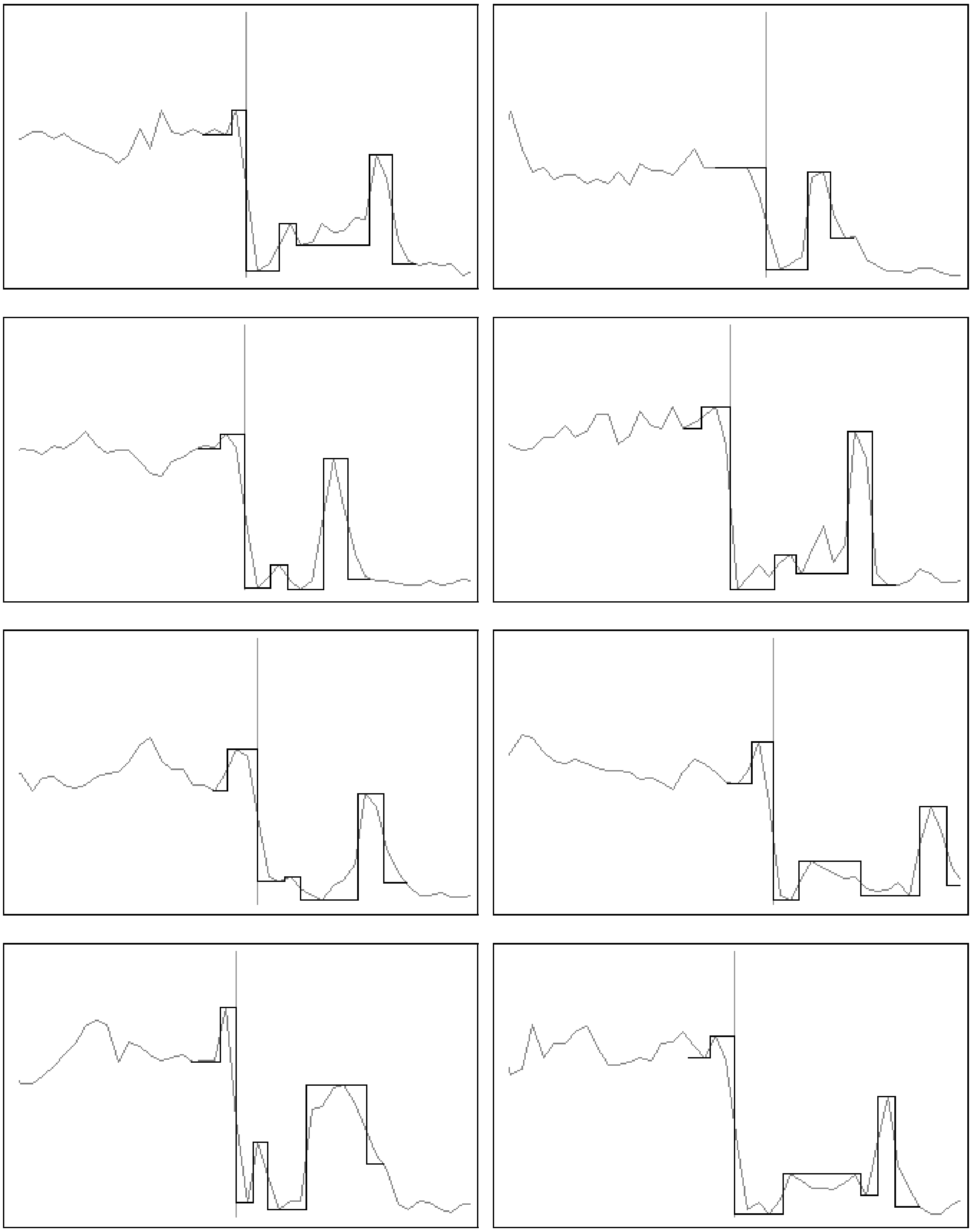


Figure 5.20: Examples of Matched Models A number of matched models are shown for the brains in Figures 5.18 and 5.19. The vertical line indicates the approximated edge location, whilst the piecewise constant curve shows the model fit. Although the models do not always produce an optimal match, they usually find a good compromise solution. The model matching is particularly good in the equatorial region of the brain, since boundary profile usually contains all the features required for accurate identification.

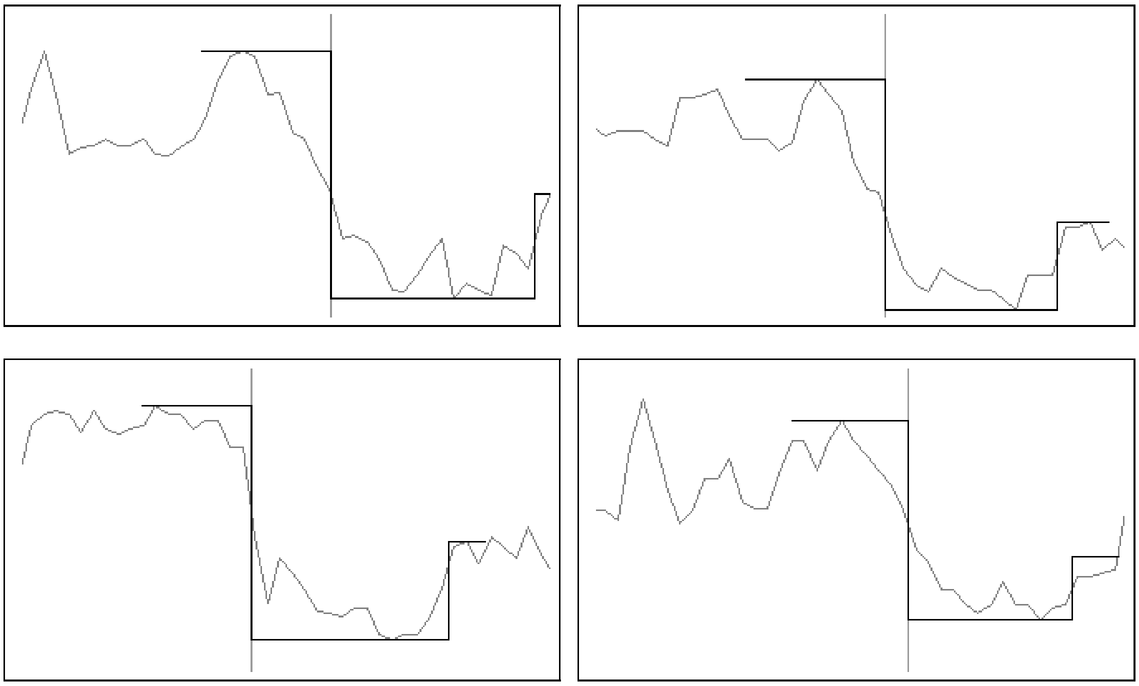


Figure 5.21: The Need for Coarse Models The intensity profiles extracted form the topmost image slices are filled with sharp peaks and dips, and warped by large foreshortening effects. Boundary location in this case is very difficult. The inclusion of very coarse models allows us to extract a reasonable approximation to the boundary, although in certain cases the edge may be shifted from the ideal location (estimated visually).

The rejection mechanism is something of a two-edged sword. While it permits us to detect boundary points with a fair degree of confidence, it also eliminates a number of correct matches, merely because the local boundary curve orientation differs too much from that of the physiological surface. However, it seems that a conservative selection of point candidates is preferable to the inclusion of spurious matches. Although the surface will be forced to update with a smaller set of points, as the surface gravitates towards the correct boundary additional correct matches should become visible and the fit can be refined. On the other hand, if we selected a significant number of false boundary points, the model update would be skewed and may yield a poor segmentation.

It is important to emphasize that the edge points shown in Figures 5.18 and 5.19 were obtained by searching along rays emanating from the indicated boundary curves. These curves represent the position of the 3D mesh model after initial alignment, which is based on the affine ICP scheme and uses extracted boundary points at each step. The initialisation procedure will be detailed in Chapter 7. Even though the initial proportion of visible boundary points was small, the model was pulled into a position where many more became visible. One would expect the proportion of matched points to improve when local mesh deformation is introduced during the main segmentation stage.

5.9 Conclusion

In order to reliably identify points on the surface of the brain, we have introduced a database of flexible boundary models and an algorithm to match these models to 1D intensity data. Modelling schemes based on heuristics and learning were considered and rejected, since they could not provide the level of robustness we required. Rather than algorithmically attempting to learn boundary structure from a set of training examples, the models are based directly on our knowledge of expected boundary anatomy. Furthermore, the matching process used to identify the required boundary points, provides us with the necessary infrastructure to predict partial volume intensities — as we shall explain in Chapter 6.

Chapter 6

Modelling the Partial Volume Effect

The voxel size and anisotropy within sparse MR data combine to produce significant partial volumes (Chapter 2). Our objective is to accurately extract boundaries in the presence of large PVE.

If we knew the volume of each tissue within a voxel, we could compute the expected MR signal. Unfortunately, such information is not directly accessible: all we have to work with are the recorded MR intensities. It is, however, possible to use this data to approximate the local tissue geometry, as we shall show. This tissue model may then be used to predict local voxel intensities, which may, in turn, be used to improve our boundary estimate.

The chapter starts by presenting a brief summary of related work. The MR imaging model we shall use for our prediction scheme is then introduced, Section 6.2. The prediction framework is presented in Section 6.3, and the manner in which the predicted intensities are employed is discussed in Section 6.4. Experimental results are presented in Section 6.5, and discussed in Section 6.6. The chapter is concluded in Section 6.7.

6.1 Related Work

Most of the work in the area of PVE analysis concerns the development of more reliable voxel-scale segmentation schemes. Many of the proposed solutions involve some sort of probabilistic modelling [18, 68] with the aim of voxel classification [60]. The data is usually high-resolution, rather than sparse and the MRI may be multi-modal [18]. In a somewhat different approach, Röll et al [55] computes a partial volume correction based on an hypothesised model of PVE variation. This is used to estimate the volume of objects with partially filled boundary voxels. An interesting approach adopted by Thacker et al. [65] uses a number of specially chosen imaging sequences to provide a set of equations which may be solved for each voxel to yield estimates of tissue volumes. There is however, a limit to the number of tissues which may be contained within any voxel, and the patient would have to be scanned several times, introducing the possibility of image registration errors. This method is not applicable for multi-modal scans which do not satisfy their very stringent protocol constraints, nor can it be retroactively applied to the kind of data we wish to analyse.

6.2 The MR Imaging model

We will assume the following idealised model for the signal from a voxel, which follows directly from the physics of MR imaging [35, 54]:

Linearity The observed voxel intensity, I_v, is a linearly weighted sum of the N_T pure tissue intensities, $\{I_i\}_{i=1,\dots,N_T}$, arising from the N_T tissue types present within the voxel:

$$I_v = \sum_{i=1}^{N_T} \alpha_i I_i \tag{6.1}$$

Tissue Weights The tissue weights, α_i, are given by

$$\alpha_i = \frac{V_i}{V}, \quad \sum_{i=1}^{N_T} \alpha_i = 1, \tag{6.2}$$

where V is the voxel volume and V_i is the volume of the tissue i present in the voxel. In other words, the voxel intensity contribution of a given tissue class is proportional to the amount of that tissue within the voxel. This model is an idealisation, albeit a common one [65, 60], since the RF pulse used to select the image slice will always introduce intensity information from neighbouring slices (see Chapter 2). This has the effect of blurring the voxel intensities somewhat; however, the model provides a satisfactory approximation for our purposes.

In general, it is not possible to recover the volumes of tissue within a voxel, and one is forced to estimate the α_i directly. In certain cases, however, anatomical knowledge may be mobilised to simplify the problem.

6.3 The Prediction Scheme

The preliminary matching stage, discussed in Section 5.5, provides a local piecewise constant boundary description at each point on the mesh-slice intersection curve. This description is really an in-plane, foreshortened approximation of the intensity profile sampled along the surface normal at that point. Given a particular location on the boundary, and an estimate of the surface normal at that point, we may use the piecewise constant model to estimate the tissue arrangement within each voxel, and hence the volume contribution of each tissue type, α_i, to the associated voxel. The estimated tissue class intensities, I_i, which are already implicit in the model and do not need to be computed, may then be combined with the tissue weights under Equation 6.1 to yield the expected voxel intensity.

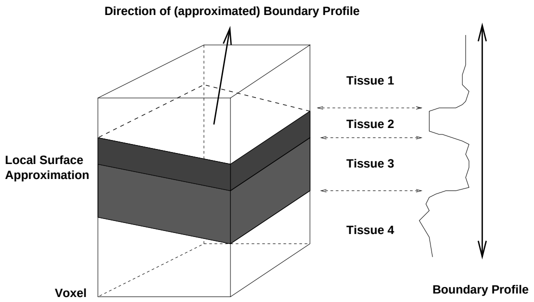


Figure 6.1: Voxel Partial Volume For each voxel, an approximation to the local boundary profile is generated which provides an estimate of the tissue organisation in the voxel. We can compute the volumes of each of these (approximately homogeneous) tissue regions and use these to predict the voxel value.

The use of a 1D piecewise constant model to approximate the local tissue distribution is justified if we make two assumptions:

1. the surface boundary is locally planar, and
2. each tissue/fluid segment is considered to have a constant length over this local planar region.

Such a planar laminar tissue arrangement is shown in Figure 6.1. Any point on the mesh surface has an associated 3D laminar element of this type, which approximates the tissue geometry in the voxel neighbourhood. Thus, if we know which laminar element affects a given voxel, we will be able to intersect the associated sequence of planes with the voxel and hence compute the nominal tissue decomposition for that voxel. Once such a polyhedral decomposition is available, we simply compute the tissue volumes and for each class and apply Equation 6.1.

The volume of each polyhedron may be precisely computed (from Gauss' Theorem) by subdividing each face into triangles and evaluating

$$V = \|\frac{1}{6}\sum_{i=1}^{N_t}(v_{i,1} \times v_{i,2}) \cdot v_{i,3}\|, \tag{6.3}$$

where N_t is the number of triangles and $\{v_{i,k}\}$ is the kth vertex of the ith triangle. Note that the order of evaluation must be consistent for each triangle (which is viewed as an oriented surface element).

For each voxel, the boundary model selected to generate the laminar element is the one associated with the closest point, P, on the surface, since this point is likely to have the most

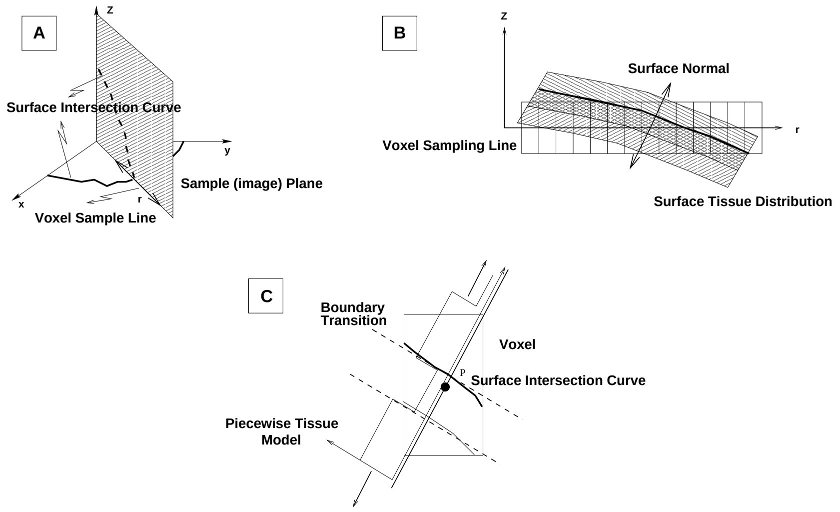


Figure 6.2: Prediction of Local Intensities A) The mesh is intersected by the imaging plane to yield a contour within the plane. For any voxel near this contour, we may predict the intensity using the local piecewise constant boundary model. B) A 1D array of voxels is extracted along a ray normal to the boundary. This set of voxels will intersect the boundary tissue distribution, which we shall assume is locally laminar. C) The 1D local piecewise constant boundary model, representing the coarse initial fit, is then expanded to a 3D "laminar element" which decomposes the voxel into a series of disjoint convex polyhedra. The volumes of these polyhedra are used to predict the voxel intensity. This 2D cross-section shows how the laminar element is shifted along the surface normal so that it is properly positioned for the decomposition calculation.

anatomically relevant model. The surface normal at P provides the orientation of the model i.e. the axis along which the transition planes are oriented. Having oriented the model in space, all that remains to be done is to "attach" it to the surface at P. Since each plane corresponds to a particular tissue boundary, and we know which boundary the surface is meant to represent, we slide the model along the normal until the correct plane coincides with P. This is illustrated in Figure 6.2C), which shows a 2D cross-section of the voxel and the decomposition resulting from the model alignment.

For a given voxel, the intensity prediction algorithm is as follows:

Algorithm 6.1 (Voxel Prediction)

1. Obtain an estimate of local oriented boundary model.
2. Treat each tissue transition in this profile as the origin of a plane which is oriented in the direction of the model axis — Figure 6.1 .
3. Compute the intersections of each transition plane with the voxel and define the bounding polyhedron for each tissue class.
4. Evaluate the volume, V_k, of each tissue polyhedron and compute α_k as $\frac{V_k}{V}$.
5. Predicted intensity is $I = \sum_{k=1}^{N_T} \alpha_k I_k$.

In itself the predicted intensity is of little use to us — we have already discussed the inappropriateness of voxel classification for our data. The next section addresses the manner in which we utilise this prediction.

6.4 The Boundary Detection Strategy

Because our data is sparse, we cannot extract reliable intensity data between slices. For this reason, the prediction scheme proceeds by examining voxels in each slice only. Furthermore, for reasons of efficiency, we do not predict the intensity of every voxel within the mesh neighbourhood. Instead, we examine 1D lines of voxels within each image slice and attempt to predict these. The setup is illustrated in Figure 6.2.

We intersect each image slice with the current mesh position to produce a set of planar boundary curves, C_i. For a given point, P_c, on such a curve, we may extract a line of voxels by sampling along the ray normal to the curve. The set of voxels is sampled so as to have N samples on either side of the intersection point, giving $2N+1$ samples in total. We wish to move the surface — represented by the intersection point P_c — along the line so that it gets closer to the object boundary. To achieve this we compute a set of $2M+1, M<N$, voxel intensities along the sample line, by applying our prediction procedure to M voxels on either side of the intersection point. Each predicted voxel has a coverage, γ, associated with it. This represents the proportion of the voxel which was covered by the laminar model element (they have bounded extent).

The predicted intensity model is translated along the search line in one pixel increments[1] and a *matching function* evaluated at each point. In our case this function is

$$M_T = \sum_{i=1}^{2M+1} \gamma_i ||I_{T+i} - P_i||^2 = \sum_{i=1}^{2M+1} \gamma_i {l_{T,i}}^2, \tag{6.4}$$

where $\{P_i\}$ are the predicted intensities and T represents some offset (in pixel units) along the search line. For a given translation T and sample, i, $l_{T,i}$ is the absolute difference between the predicted and observed intensities. Our experiments showed that partially covered voxels tended to undermine the matching process, so we discard them, that is, γ is taken to be zero if the coverage is incomplete. This only affects peripheral voxels, and is a marginal affect.

The best translational match between our predicted intensity profile and the underlying observed data is obtained when M_T is at a minimum. However, this match may still be spurious. This might happen if we are too far away to sense the boundary — since M_T will achieve a minimum somewhere along our pixel line, we will be left with an incorrect match. We have included a simple rejection mechanism to try and exclude such mismatches. Using

[1] One can use sub-pixel increments and use an upsampled model, but this is not likely to give you more accuracy.

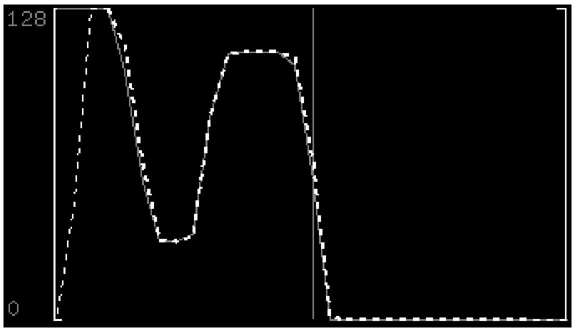


Figure 6.3: Profile Prediction and Matching This image shows how the predicted profile (white dotted line) has been matched with the real intensity data in the image (thin grey line). The vertical line indicates the position of the edge estimated through the matching process.

$\{l_{T,i}\}$, an absolute "measure of goodness" is computed for each tentative match. The measure lies between 0 and 1 and is computed as

$$p = 1 - \frac{\sqrt{\sigma_d^2}}{\mathrm{maxD}}, \tag{6.5}$$

where the variance σ_d and the maximum absolute difference, maxD, are computed over $\{l_{T,i}\}$. Since there is usually a large discrepancy between a boundaries and spurious profiles, a suitable threshold may be established by empirical means.

The end result of this process is a displacement vector in the plane, for P_c, indicating where the mesh intersection point (which corresponds to the boundary of interest) wishes to move.

Figure 6.3 shows the predicted intensity profile for a point on a mesh object (sphere) and indicates the best match, under M_T, with the real in-plane intensity data. The vertical line indicates the associated edge estimate. The match is in a plane raised from the equator by about $\frac{2}{3}$ of the sphere's radius.

6.5 Experiments on Synthetic Data

These experiments are intended to evaluate the robustness of the prediction scheme:

1. when we have perfect data,
2. when there is image noise,
3. when there is noise in the boundary model approximation.

To allow us access to ground truth, we have utilised analytically defined test objects.

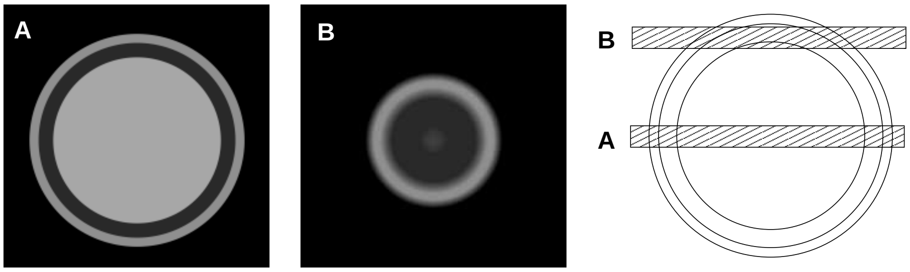


Figure 6.4: Synthetic Volumetric Data A sphere composed of several tissue classes, represented by a unique intensity. A) shows a slice near the equator while B) shows a section from the poles. PVE was simulated by recursive voxel sub-division.

6.5.1 Test Data

The test objects consisted of a variety of spheres with shells of different "tissue" near the surface. These tissue types are represented by different absolute intensities. Partial volume effects are simulated by recursive sub-division of voxels using the sphere formulae to determine intensity values at sub-voxel scales. The voxel dimensions were $0.9375 \times 0.9375 \times 5\text{mm}^3$ with a 2mm gap, since this corresponds to the sampling protocol used to collect the data used in this thesis. Typical cross-sectional images are shown in Figure 6.4.

Our spherical model has a range of tissue shells varying in thickness from 2–6mm, since this corresponds to the level of detail within the boundary of interest. For the synthetic data presented here, the boundary models are static and known *a priori* so no additional processing is required.

6.5.2 Quantifying Edge Estimation Errors

The surface will intersect the predicted intensity profile at the centre of the computed voxel span. As the predicted profile slides along the line of intensities, one may think of it as pulling the "embedded" surface along with it. We can compute the location of the edge (given by the final position of the centre of profile model) in 3D. Likewise, for the series of analytic shapes we have investigated, we can compute the true edge position along the line of extracted pixels.

Thus, we have a means of computing the discrepancy between ground truth and our estimate of edge location. The statistics given here are based on slice-wise averaging of absolute distances. For each slice, we can compute the absolute difference between the true and approximated edges (in mm) and then generate the mean of standard deviation of this set of measurements. We can also compute the angle the sampling plane makes with the surface normal on each slice (this is constant for the shapes we have investigated; in general, this need not be so). We use the angle to index the statistics. Negative angles index measures from

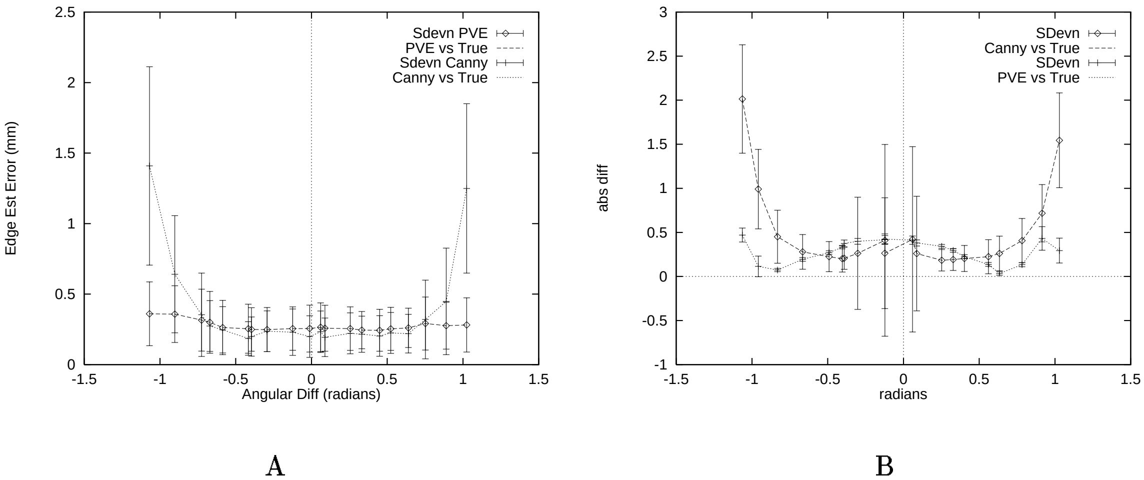


Figure 6.5: Partial Volumes and Edge Location The graph shows the accuracy with which the two schemes (Canny and PVE) isolate the object boundaries. The data represents the mean and standard deviation of the boundary estimation error on each slice. The slices are indexed by the angle between the horizontal and a ray penetrating the sphere boundary on the relevant slice. A) The tissue bands have a separation of 4–6mm, which is typical in many cases. Observe how the Canny estimates get progressively worse as one nears the poles. The standard deviation of the both methods B) The tissue bands have a separation of 2–3 mm, which is very narrow. In this case the results are a little less decisive.

below the “equator”. We compute edge estimates using 1) PVE prediction and 2) Canny (with contrast sensitive edge selection), to illustrate the effects of partial volumes on edge estimation. The Canny parameters were selected for optimal edge detection at the equator — we do not assume any further knowledge of boundary scale etc. This is reasonable, since in general we do not know how edge/boundary scale will vary over an object.

Test I: Partial Volumes and Edge Location As one moves towards the poles the effects of PVE become marked and edge detectors based on gradient maxima will no longer provide a good estimate of edge location. This is illustrated in Figure 6.5.

Observe that partial volumes ensure that Canny provides consistently poor results near the poles. When the tissue widths are 3mm or greater, the PVE model provides excellent results (for the given sphere size). Canny wins out near the equator, since our model is based on a planar approximation, whereas the sphere surface is curved. In this region, partial volume effects are relatively small and Canny detects the edge to high precision.

Test II: Performance in Noisy Images To examine the effectiveness of the matching strategy, we added Gaussian noise to the sphere images. This noise was zero mean and unit variance, scaled by some factor we specified. With the addition of significant noise, Canny performs extremely poorly. One can increase the smoothing factor, but this impacts on edge localisation. Furthermore, there is no obvious way to select the scale at which edges should

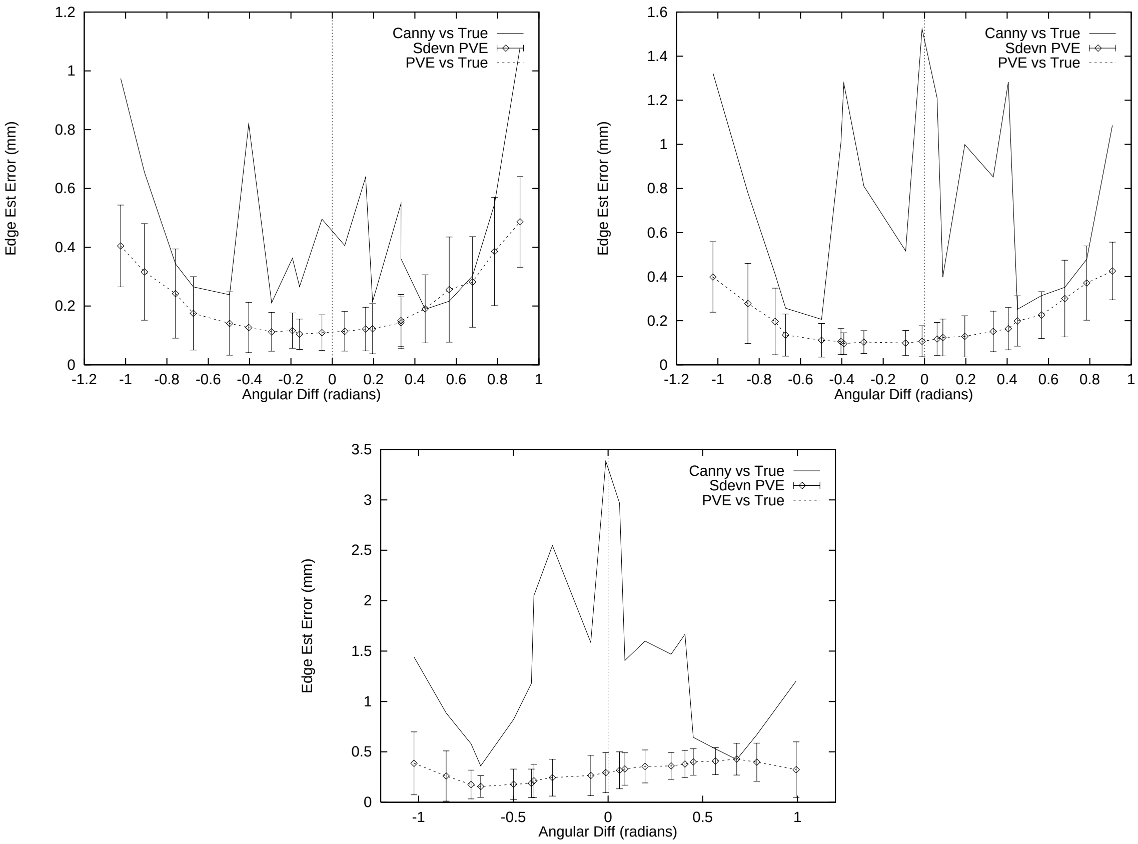


Figure 6.6: Matching in the Presence of Noise These graphs show the boundary estimation error when zero-mean Gaussian noise $\sigma = 3, 6, 10$ ($\sigma = 6$ is about 2% of intensity range) is added to the image intensities. Canny performs poorly, whilst the prediction scheme remains largely unaffected.

be detected for arbitrary data. We chose parameters which worked well near the equator and kept these values constant throughout. (only unit variance smoothing was utilised). Figure 6.6 shows results for 3 levels of noise.

In all cases, the simple matching provides good estimates for the edge location, while Canny gets progressively worse — Figure 6.6. The PVE matching involves the use of a correlation-type operation (over a 1D region) it is inherently more robust than the maximum signed contrast used in our Canny implementation. Since the noise essentially cancels, the PVE results certainly make sense.

Test III: Performance with Noisy Models In general, our boundary model will not correspond completely with the tissue layout over the neighbourhood of interest. To evaluate the effect of this discrepancy on our modelling process, we have perturbed the locations of tissue transitions in the model by adding scaled uniform random noise. The noise is distributed over the interval $[0, s]$, where s is the scale in mm. Transition crossings are not permitted i.e. the tissue ordering must remain fixed. If the random shift would violate

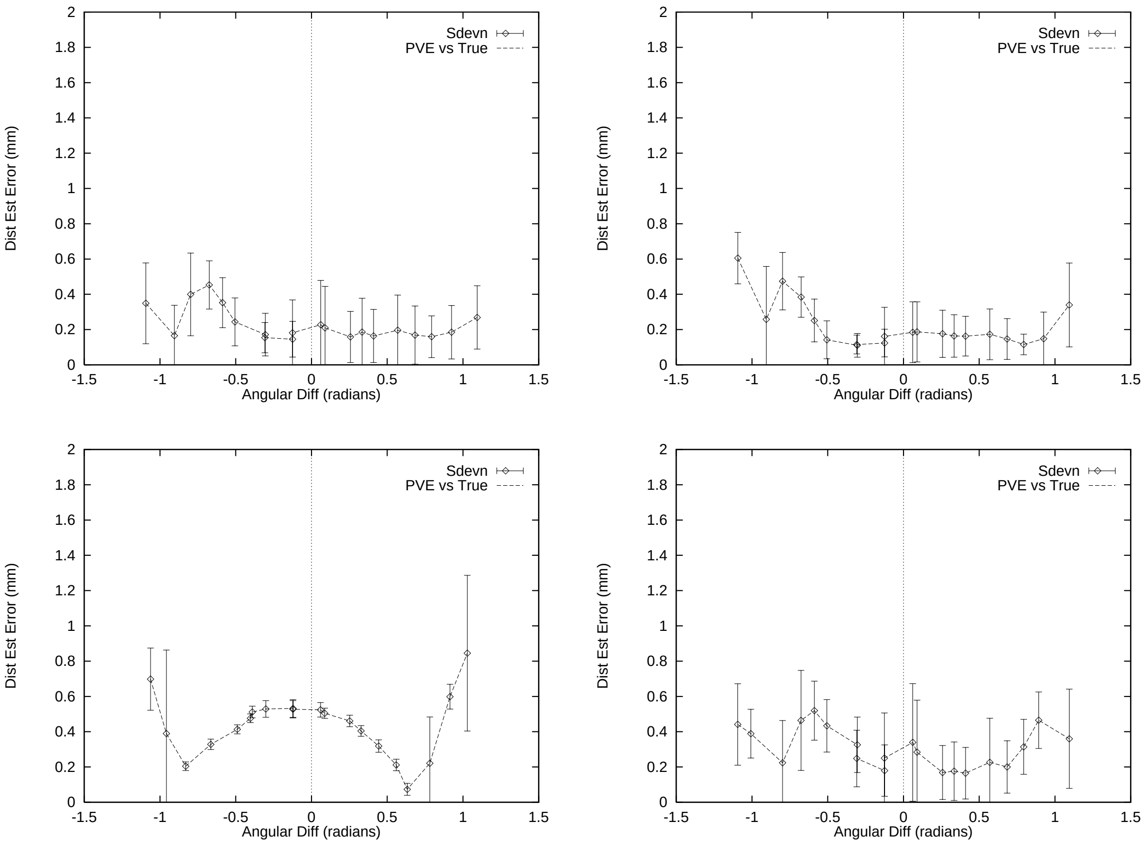


Figure 6.7: Estimates with Model Noise (2-3mm sphere) Two examples are shown which illustrate the effect on boundary localisation of perturbing the model transitions. The data is for the case of partial volume boundary matching, and the noise is uniformly distributed over the interval $[0, 1]$mm. Even for the case of the 2—3mm sphere, the average localisation error remains low.

this constraint, we take the average of the two transition locations to either side as its new location. Figure 6.7 shows some results for the very narrow tissue sphere (2–3mm) for the case of 1mm scaling.

Figure 6.8 shows the results for the case of the 4–6mm sphere. For this object the prediction is quite robust, even with the 2mm noise.

However, the 2-3mm tissue boundary may provide a considerably less robust estimate when 1mm noise is applied to the model. Again, this makes sense, since the smallest tissue band is only 2mm wide!

IV: Noisy (non-ideal) Profile Estimates with noisy data Finally, we looked at noisy boundary models in noisy data. The results shown in Figure 6.9 show what happens for the 4-6mm models in this case. The results are still good in this case.

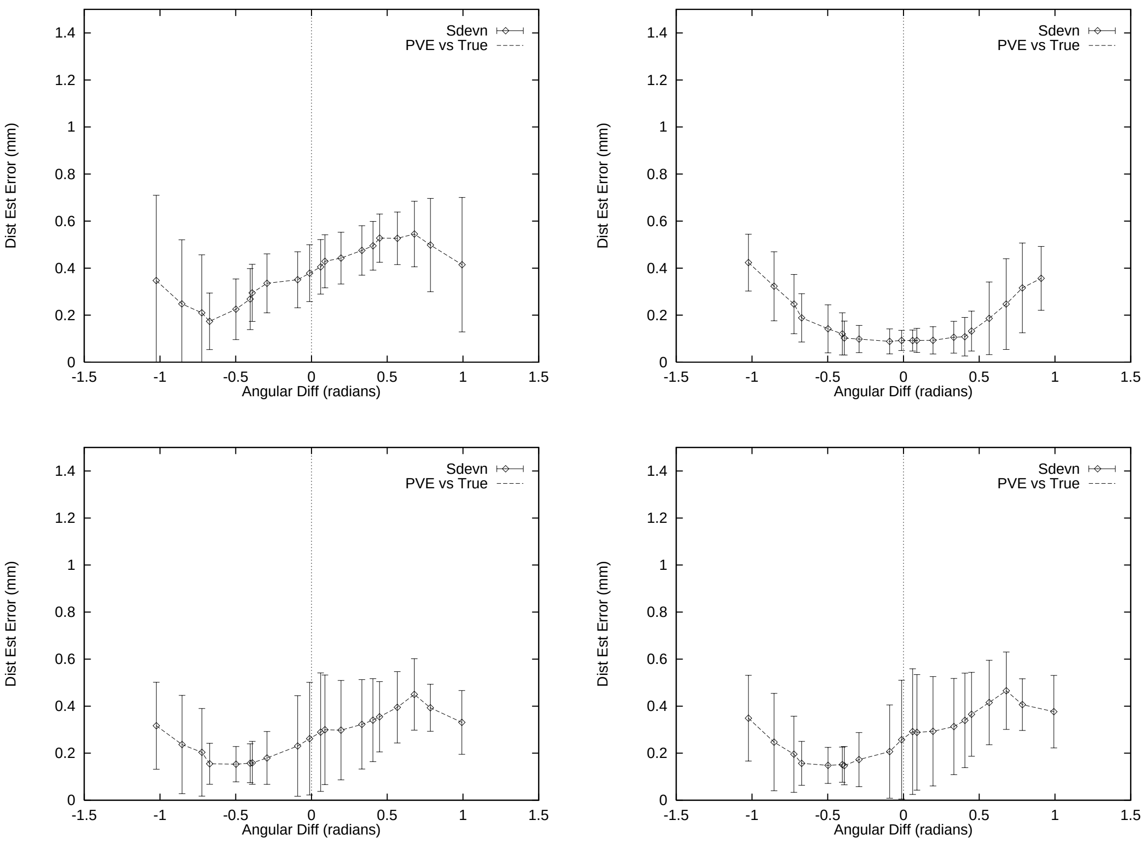


Figure 6.8: Estimates with Model Noise (4–6mm sphere) Boundary localisation examples in which the location of model transitions were perturbed by uniform random noise in the interval $[0, 2]$mm. The average boundary estimation error is less than 1mm in all cases.

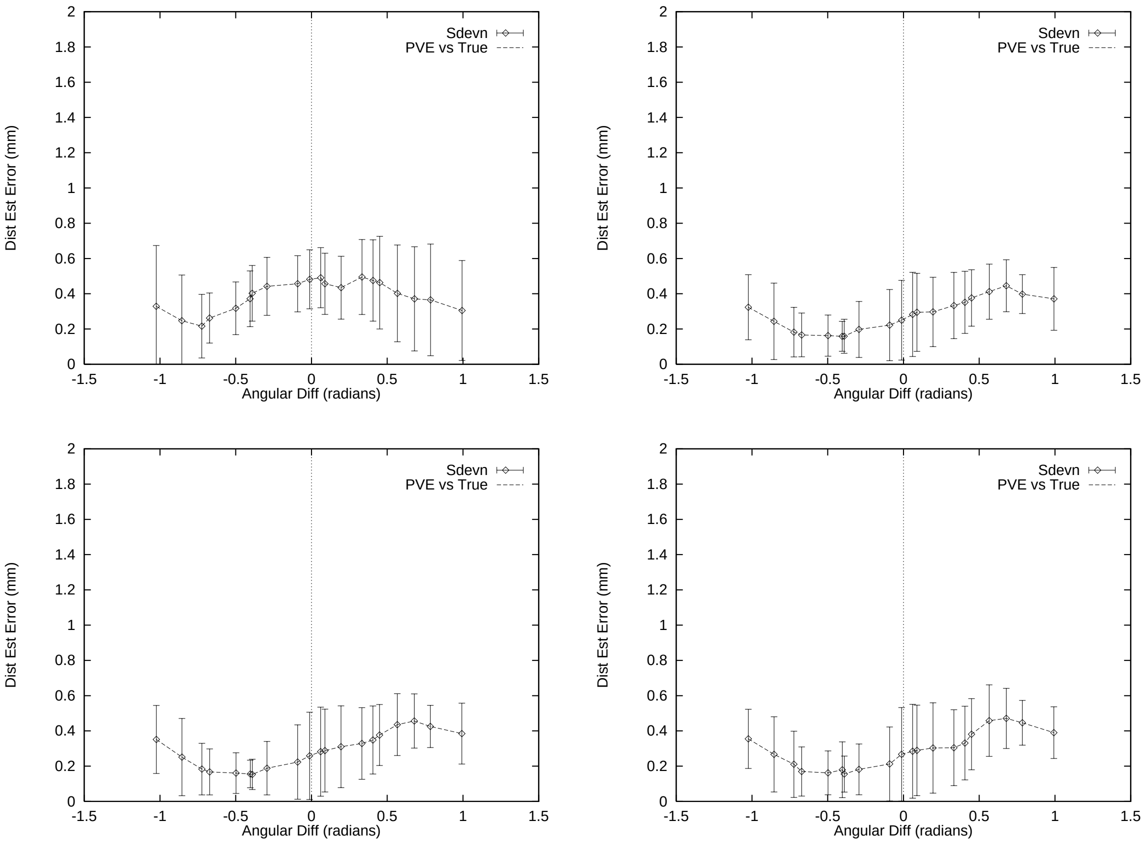


Figure 6.9: Image and Model Noise (4-6mm sphere) In these examples, the model noise is scaled by 2mm and the image noise is N(0,1) scaled by 6mm.

V: Using a Fixed Boundary Model to predict real MR Data To obtain some insight into the robustness of prediction process for more complex data sets, we applied a simple boundary model to a sparse MRI image. Figure 6.10 shows matched profiles from this data set. Whilst there is a clear discrepancy between the real intensity data and the predicted profile, the results are encouraging, since they show that even with such an extremely crude approximation we can extract correct boundary matches.

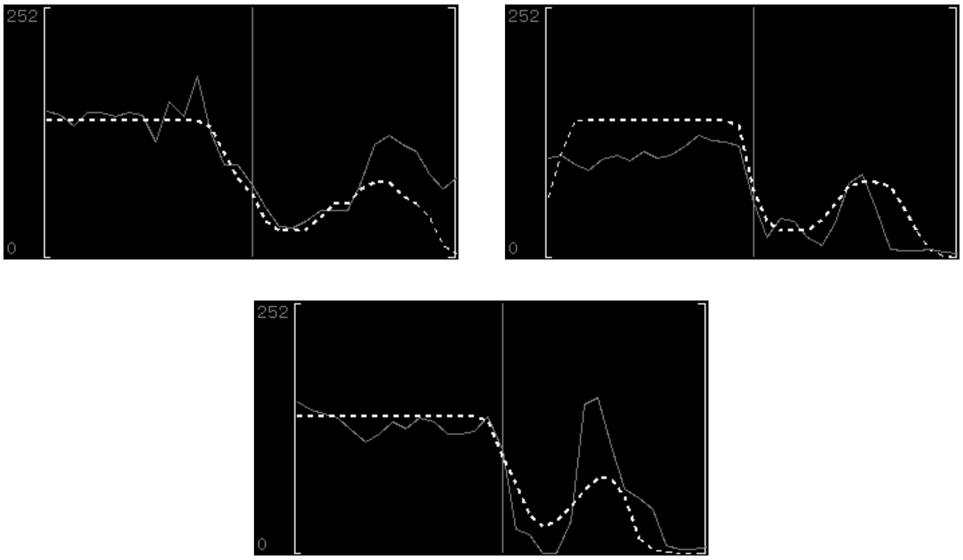


Figure 6.10: Prediction in Real Data Using a Single Boundary Model A single, simple boundary model was used to predict these profiles, which were subsequently matched under M_T. The 1st and 2nd profiles were extracted from the top of the brain, whilst the last was extracted from the bottom. The vertical line indicates the position of the boundary edge. The grey curve represents the real 1D intensity data, while the dashed line shows the predicted intensity curve.

The results presented here show two important things. Firstly, that foreshortening effects are correctly handled — as indicated by the sequence in Figure 6.10. As we move further up the brain, the predicted intensity profile opens up, just as the the observed intensity profile does. As far as boundary detection is concerned, this equates to a change in boundary scale, and a static edge detector would almost certainly experience localisation problems. However, the boundary localisation under the PV prediction scheme is unaffected. Secondly, even with predicted intensity curves which do not precisely match the observed data, we can still obtain good boundary localisation — as long as the gross profile features are present. This is a direct consequence of the matching scheme.

If the mesh surface is not lying on the appropriate boundary the predicted curves will appear warped — stretched or compressed — relative to the observed intensity samples — Figure 6.10. However, provided there is sufficient gross similarity between the predicted and observed data, there will be a gradual improvement in the curve correlation as the surface converges towards the goal boundary.

6.5.3 Prediction Using Matched Models

If the boundary matching scheme outlined in Section 5.5 has been applied, we may utilise these models to define the local laminar element at that point. The in-plane model can be subjected to a cosine foreshortening which will map it onto the direction of the surface normal, and used to directly predict the intensities of the voxels in the immediate vicinity of the boundary point.

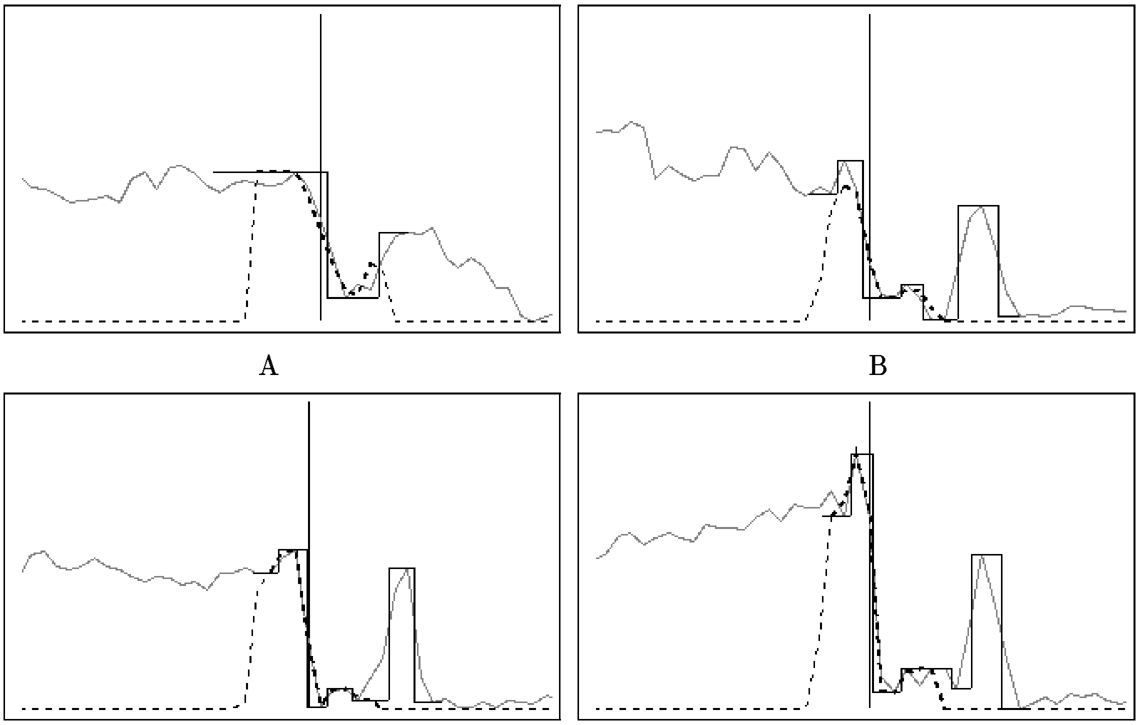
This kind of matching is similar to that proposed above, but uses only the boundary model located at the intersection point, rather than those further away. This initial scaled model estimate may be refined in an attempt to obtain a better match in the vicinity of the edge estimate.

The refinement proceeds by brute force: the normal orientation and individual piecewise constant lengths are varied differentially, and the resulting predicted intensity curves compared with the observed signal in the boundary point neighbourhood. By varying the normal orientation, we can account for mis-alignment of the local surface patch, which would otherwise create spurious degradation in the predicted intensity. Varying the segment widths allows also one to compensate for errors in the location of the closest surface point. This process is very expensive, requiring an exponential number of partial voxel calculations (which are individually complex and expensive). More specifically, if there are N_T segments, which means that $N_T - 1$ transitions are to be varied, and each may be shifted into one of l different locations, the number of voxel evaluations will be $N_\theta l^{N_T - 1}$. The value N_θ represents the number of different normal orientations which are tested. This number rapidly becomes large, and given that a set of voxels at each boundary point is required for matching purposes, the time required to traverse the search space may be excessive[2]. In general, the number of planes intersecting a voxel will be less than or equal to three[3] — if this aspect could be exploited, computation time could be accordingly reduced.

To test the usefulness of this scheme, the initial matched models were used to predict a local intensity span, 11 voxels in length. In this instance, only angular refinement was permitted: there was no differential modification of segment widths. Nonetheless, the predicted curves over much of the model were quite good — Figure 6.11. The exception was found in the polar regions of the brain, where the surface was not close to the boundary of interest, and the boundaries have complex signatures. Figure 6.12A,D). These results show that the scheme is workable, provided improved estimates can be obtained for models at the top of the brain.

[2]Hours, perhaps, for a complex set of boundary models, a large number of voxels, and a reasonably fine parameter increment.

[3]There are seldom more than 4 tissues in a voxel, or so empirical evidence would suggest.



A B C D

Figure 6.11: Prediction Using A Good Model A-D) When the matched model is fairly accurate, a good prediction is obtained. The matched model is shown, as well as the predicted profile (the dashed curve). The vertical line indicates the edge estimate resulting from matching the predicted profile under M_T. Note that the dimmer part of the predicted curve has a coverage of less than unity, and is excluded under the matching criterion. The length of the predicted voxel span was 11 samples.

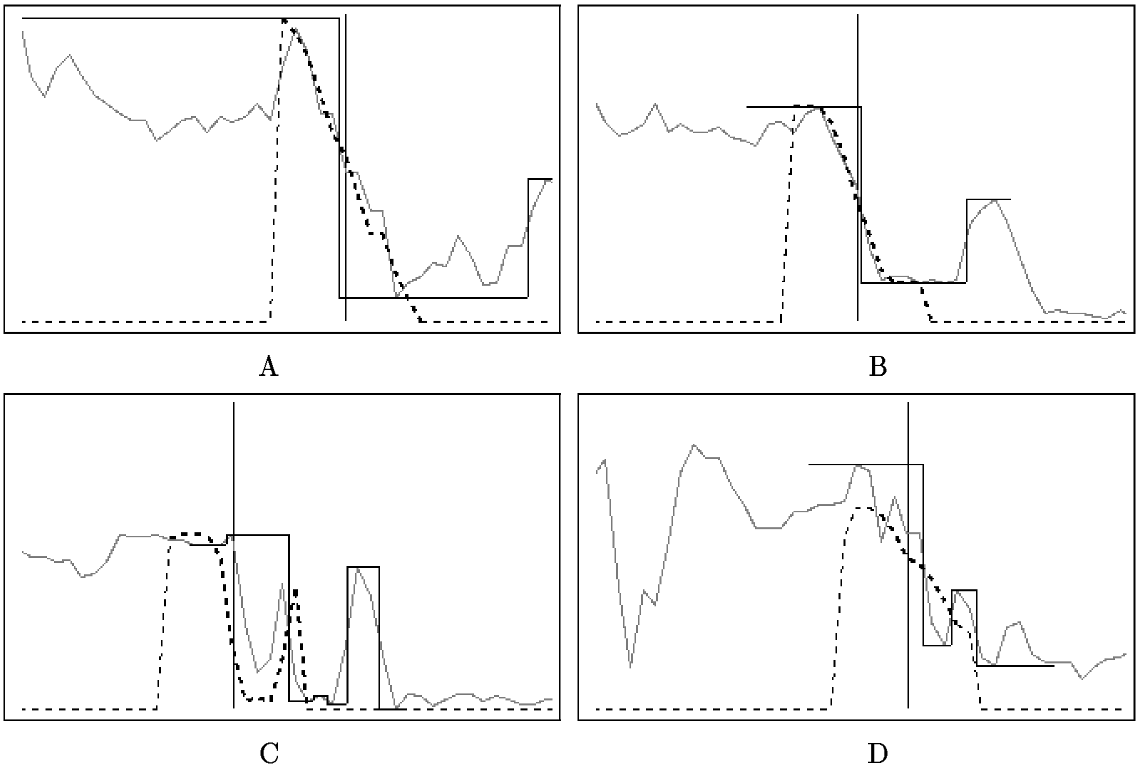


Figure 6.12: Prediction Using Coarse or Incomplete Models A,B) If the model is coarse, but the predicted span is sufficiently local, matching can still yield a reasonable edge estimate. C) In this case, the model has actually failed to match the curve properly — the PVE prediction, although based on a erroneous match, has actually improved the edge estimate! In general, however, such poor model matches will generate a poorer PVE edge estimate. D) If the model is coarse, but the underlying data is locally complex, only a very inaccurate profile will be predicted. Although the matching criterion is robust, and may find a good compromise, there is no guarantee that matches based on such coarse models will improve the edge estimate.

6.6 Discussion

The sparse segmentation algorithm requires the calculation of the closest point on the surface mesh to a given neighbourhood point. Clearly the sampling density of this surface will impact on the accuracy with which we can approximate this point. We have experimented with spline interpolants and the results seem to indicate that the planar approximation used is sufficient for our purposes. Further indication of the robustness of the scheme may be deduced from the experiments in which the boundary model transitions were perturbed. These experiments showed that the prediction remains good even when the model is displaced from its correct location.

The simple matching function we used proved adequate. Because it is a correlation-type operator, it is also robust in the presence of noise and global luminance shifts. Furthermore, the predicted in-plane intensities do not have to be highly accurate for a good match. As the predicted span becomes larger, there is a chance that the closest point will no longer be located on the correct part of the surface. However, we have stressed that this segmentation algorithm is only appropriate for *local* model updates.

The principal problem with this approach is the construction of a representative piecewise constant model in the voxel neighbourhood. While the models generated by the initial boundary detection process provide just such a representation, this is often too coarse in the areas in which PVE is most acute. However, if the initial model fit is assumed to provide a good match, then it is still possible to use the model to predict a small voxel neighbourhood about the edge. The most serious objection to this scheme arises from the poor model fits obtained at the top of the brain. These need to be improved if the PV prediction is to be useful in this region.

6.7 Conclusion

In this chapter, we have introduced a framework which enables us to predict partial volumes in the vicinity of the surface model. These predicted values can be used to refine the local boundary estimate. The prediction scheme is based on a local geometric construction, which is generated directly from the matched boundary models resulting from the initial boundary detection phase.

Chapter 7

Surface-based Segmentation

The preceding chapters have established the infrastructure we require for our segmentation scheme. For any given model position, the techniques introduced in Chapter 5 enable us to generate a new set of 3D data points, representing an improved estimate of the surface we wish to extract. This chapter describes the process by which the current model position will be updated. We desire an update strategy which will ensure that the resulting surface smoothly interpolates the given data, whilst respecting our shape constraints (Chapter 4). Once again the restrictions imposed by sparse acquisition must be considered. In particular, significant truncation and (acquisition direction) sub-sampling artifacts must be accounted for in the update scheme. Finally, the robustness of the entire scheme must be validated against a set of examples for which we have some notion of ground truth.

The chapter begins with a brief overview of the active shape model (ASM), the basic mechanism we shall use to enforce our shape constraints. Section 7.2 introduces our extensions, and explains why they are necessary. The issue of template initialisation is addressed in Section 7.3. In Section 7.4 we give a detailed description of the segmentation strategy. Segmentation results are presented in Section 7.5 and analysed in Section 7.6. Section 7.7 concludes this final chapter.

7.1 Utilising the Shape Constraint

The point distribution model, introduced in Chapter 4, describes the permissible shape variation a class of objects may exhibit, but does not include an algorithm to exploit this information during the segmentation process. The active shape model (ASM) proposed by Cootes and Taylor [24], may be used for this purpose.

Under this scheme, a preferred displacement is computed for each mesh vertex, based on the the available boundary data. The set of these vertex displacements — which represents the model deformation — is then "projected" into the PDM subspace to arrive at the permissible

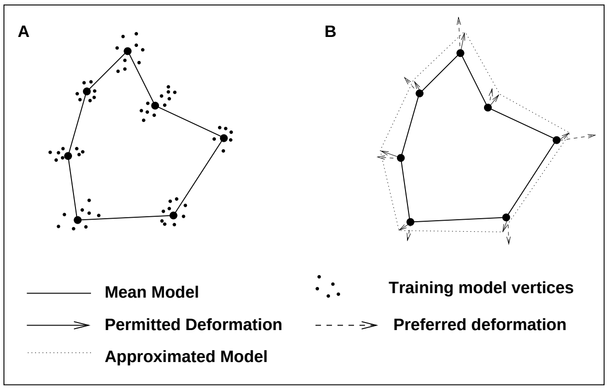


Figure 7.1: 2D Active Shape Model A) An input training sequence, represented by a set of corresponding points with the connectivity shown. The mean shape is overlayed. B) A preferred displacement is computed for each vertex. The set of these displacements is "projected" into the associated PDM subspace to yield a permitted model deformation. The distribution of the input training sequence directly determines the permissible deformation of the ASM.

deformation, Figure 7.1. The projection operation also provides a set of parameters which may be used to deduce information about the shape instance under consideration. The ASM segmentation scheme will be considered in more detail in Section 7.4.3.

The PDM/ASM framework provides a coherent approach to constrained segmentation. Unfortunately, in its basic form, this methodology provides no mechanism for the segmentation of out-of-class variation, nor does it provide a means of dealing with sparse data.

7.2 The Segmentation Framework

The basic ASM approach provides a sound methodology for constrained segmentation, but assumes that:

1. a similarity transformation is sufficient to align shape instances,
2. a displacement can always be computed for each vertex, and
3. the shape instance may be well approximated within the learned shape space.

We require an affine normalisation prior to the construction of the shape template. Consequently, the ASM segmentation procedure, which is based on the PDM representation generated during template construction, cannot properly utilise the template information if only a similarity is permitted prior to deformation projection.

Since we have adopted an in-plane boundary detection scheme, there is no direct means of coupling the detected boundary points to the mesh vertex set. One may, of course, evaluate the closest data point to a given vertex, and use this to define the required vertex deformation, as in Delingette [30]. Unfortunately, due to the sparsity of the data, the resulting deformation may produce a shape which is truncated or flattened.

The ASM is designed specifically to improve the robustness of segmentation by excluding out-of-class shape variation (where the "in-class" variability is encapsulated by the training examples). We require a means of segmenting out-of-class structure, since we wish to investigate pathology. However, we cannot simply discard the shape constraint, since this would sacrifice the robustness we have worked so hard to obtain. Furthermore, for clinical assessment, it would be useful to determine the extent to which a given shape instance varies from the "normal range". The learned shape constraints enable us to estimate the best normal approximation for the given shape instance.

In order to obtain a more descriptive segmentation, we require a scheme which will allow us to smoothly and locally move away from our initial shape estimate. This new deformation should respect the outcome of the first stage as far as possible. The segmentation approach adopted by Montagnat [52] (see Section 4.1 in Chapter 4), uses a static simplex template to perform constrained segmentation. The simplex template provides a reference shape which is used to interpolate missing data, as well as locally constraining the permissible surface deformation. Unfortunately, the use of a static shape template may result in an over-constrained segmentation, particularly for highly variable anatomy. However, since we have a patient specific simplex model — the output of the ASM segmentation step — we can apply this approach to generate a less constrained, more representative shape description.

In light of these observations, we have implemented the following segmentation strategy:

Affine ASM The basic ASM is extended to include an Affine normalisation.

Modified Closest Point For each vertex, the closest data point is used to define the required displacement. The closest point definition is modified to compensate for problems arising from data sparsity and truncation. Furthermore, the distance between the vertex and the closest data point determines the magnitude of the vertex displacement.

Two-Tier Segmentation The segmentation proceeds in two stages. In the first, the ASM is employed to achieve a constrained segmentation. This provides the normal component of the shape instance. In the second phase, the ASM constraint is dropped and the deformation is allowed to proceed until the surface update method converges. For this stage, a *simplex template* is used, which provides a means of preserving the basic shape which emerges from initial segmentation.

The basic segmentation strategy is simple:

1. Map the model into the volumetric image.

2. Locate the boundary of interest.

3. Update the model to better fit the boundary data.

4. Repeat Steps 2 and 3 until a suitable fit is obtained.

The remainder of this chapter addresses each stage in detail, starting with template initialisation.

7.3 Template Initialisation

The segmentation is initialised by placing the shape template in a suitable location within the volumetric data set. In this case, a location is deemed suitable if we are able to compute displacements for each model vertex which will draw the surface closer to the desired boundaries. Because the model update contains a global fitting stage, even partial recovery of the boundary data will allow for a refinement of the shape approximation in subsequent iterations.

We seek a transformation which will map the shape template from its coordinate frame into the data frame. The transformation should ensure that the model is suitably placed and correctly oriented. The estimation of our transformation is complicated by the following considerations:

- the MRI data may be truncated (i.e. the Field of View may exclude the extremities of the brain) and is sparse,
- the shape instance is (generally) not related to the template by a rigid transformation,
- the data will contain additional structures (such as the cerebellum) which are not easily distinguishable from cerebral matter.
- we need to ensure that the correct half of the hemisphere is selected during the mapping process.

Fortunately we have access to several kinds of prior information which can facilitate this task. In particular, we have a description of the mid-saggital plane which provides a means of orienting the model.

In a similar fashion to the approach we adopted in Section 4.5.1, we have chosen to construct the mapping based on the identification of bounding boxes. We have chosen this approach because:

- the MSP constraint can be easily utilised — we simply force one side of the bounding box to lie on the plane, and,

- the establishment of correspondences between the model and the shape instance is very difficult: if we knew the shape instance, our segmentation would be complete. In contrast, the bounding box method provides a crude but robust way of computing our desired transformation in which only a very rough idea of the shape instance location is required — this can be obtained by simple thresholding.

The approach relies on the identification of two object centred descriptions, one for the shape template and one for the data instance.

7.3.1 Computing the Template Bounding Box

The construction of a mesh object bounding box and the inclusion of the MSP constraint were discussed in Section 4.5.1. The MSP was semi-automatically extracted for each brain in the training set, using a propagating snake curve.

The template exists in a (right-handed) canonical frame in which the plane normal is oriented along the unit vector $\mathbf{i}$. A principal components analysis, based on the convex hull of the entire model point set projected onto this plane, provides two principal axes which are mapped onto the remaining coordinate axes, $\mathbf{j}$ and $\mathbf{k}$. The centroid of the resulting model bounding box (with axes along $\mathbf{i}, \mathbf{j}, \mathbf{k}$) is projected onto the plane and mapped onto the origin. Using this information together with our prior knowledge, we can identify the template type (left/right hemisphere; axial or coronal acquisition[1]) as well as its spatial extent (represented by the bounding box).

7.3.2 Computing the Shape Instance Bounding Box

The extraction of the shape instance bounding box requires the (approximate) identification of the brain in each slice of the volumetric image. Because MR data is characterised by a dark background (for all protocols), we can apply a simple thresholding technique to obtain a binary image in which background (corresponding to air and, possibly, the intracranial cavity and ventricles) is black and all else is white. We do not, however, wish our binary segmentation to contain the skull or any non-cerebral matter. By selecting an appropriate threshold and computing the largest connected component on each slice, we can essentially eliminate the pixels associated with spurious structure.

The brain component within each slice may be placed within a 2D bounding box defined by its principal components. The longest medial line of the bounding box (for axial scans) provides a crude approximation to the mid-saggital line, the accuracy of which depends on the

[1] Saggital scans are more difficult to deal with, since they have no line of symmetry. We could ignore this problem since our database did not contain any!

degree of head-tilt and the location of the slice in the brain (see below). Points on these lines from each slice may be used as input to a plane fitting procedure to estimate the MSP. Then, applying the same procedure utilised in Section 4.5.1, we may construct a plane-constrained bounding box, suitable for use in the estimation of our transformation.

The following steps are applied (for axial scans):

Thresholding The choice of the intensity threshold for each slice is problematic. If the value is too low, unwanted structure emerges. Conversely, a high threshold may eliminate the intensities we wish to detect. In general, there may be luminance changes from slice to slice which make the selection of a global threshold untenable. One would expect, however, that tissues will retain their relative intensity separation from slice to slice. We have opted for a simple solution which works well on most slices: a pre-specified cut-off of the dynamic range of intensities is chosen. More specifically, the threshold value, I_t, given by

$$I_t = I_{\min} + r(I_{\max} - I_{\min}).$$

where $r \in [0, 1]$ is fixed at 0.35 for the data we wish to segment. This value was chosen empirically, by examining the results over a number of representative (T2-weighted) data sets. The threshold will also adjust correctly under uniform intensity rescaling.

Largest Connected Component (LCC) We then compute the largest *connected component*, C_i, for each slice. This will eliminate the skull and non-brain matter if the threshold has been set appropriately. There is usually a high level of image noise in lower slices due to flow artifacts. The combination of our thresholding procedure and 4-connectivity appears to be sufficient to eliminate these points. The largest connected component will usually encompass both hemispheres and (in most cases) the cerebellum. The inclusion of the cerebellum is not desirable, since this structure is not represented within our model and may induce a large distortion of the bounding box estimate. Most studies of gross brain asymmetry exclude the cerebellum, so if the results are to be comparable, this structure must not be segmented. In order to remove obvious cerebellar components, we apply the heuristic: *area of cerebellar cross-section is less than half of the largest axial cerebral cross-section.* Of course, since the cerebellum is on the underside of the brain, we do not apply this rule to the upper half of the brain. This heuristic is rather crude, and assumes that the brain is roughly aligned with the imaging axes, and that the scan is axial. In this case, the cerebellum is isolated from the cerebrum on the bottom-most slices. Fortunately, as we shall show, the initialisation can be quite poor and still provide a satisfactory starting point for our segmentation scheme.

Compute the Convex Hull of C_i The computed LCC may contain holes and concavities which will skew our PCA. To overcome this problem, we compute the convex hull, H_i, of this point set. A uniform sampling of points along this boundary is then used as input to the PCA procedure. This has the added benefit of reducing computational time, since only points along a contour are considered rather than points within an complex region.

Estimate Mid-Saggital Line For each slice we perform a PCA on H_i to extract the principal components $\{e_1, e_2\}$ and data mean, μ — these are used to estimate the mid-saggital line on each slice as a set of points along $\mu + \alpha\, e_1$.

Compute MSP The MSP is computed by performing robust [34] orthogonal regression on the estimated point set from each slice.

Construct a Bounding Box The full 3D point set generated above is projected onto the estimated MSP and the procedure outlined previously is used to construct the object centred representation. Because we know the image acquisition direction/protocol, we can deduce which half of the bounding box corresponds to the hemisphere model.

To associate the template with the data we must select an appropriate mapping. A rigid transformation will not, in general, be sufficient. Ideally, we would like to compute an affine transformation or affine with 3 independent scalings (along each axis of the bounding box — Chapter 4). However, since our data is truncated, we must exercise some caution when selecting bounding box points for the calculation. The most reliable bounding box extents are provided by the components (approximately) oriented in the slice direction. We use the relative expansion/dilation of the bounding box in the MSP normal direction to define a global scaling for the transformation, and then compute the similarity which associates the the two coordinate frames.

7.3.3 Initialisation Examples

Figure 7.2 shows the initial placement of the template,and its position after the application of the initialisation mapping. Observe that the model is correctly oriented and that the placement and gross scaling are essentially correct.

Figure 7.3 provides a more detailed view of the mechanics behind the estimation of the initialisation mapping.

In the first example, Figure 7.3A), the MSP is very well approximated because the connected components are fairly symmetric. There is also good lateral correspondence between the model and the data. There is is clear foreshortening of the data in the posterior-anterior direction — this is due to a rotation of the model about the MSP normal, rather than being indicative of poor scaling. This rotation is unavoidable (without considerably more effort)

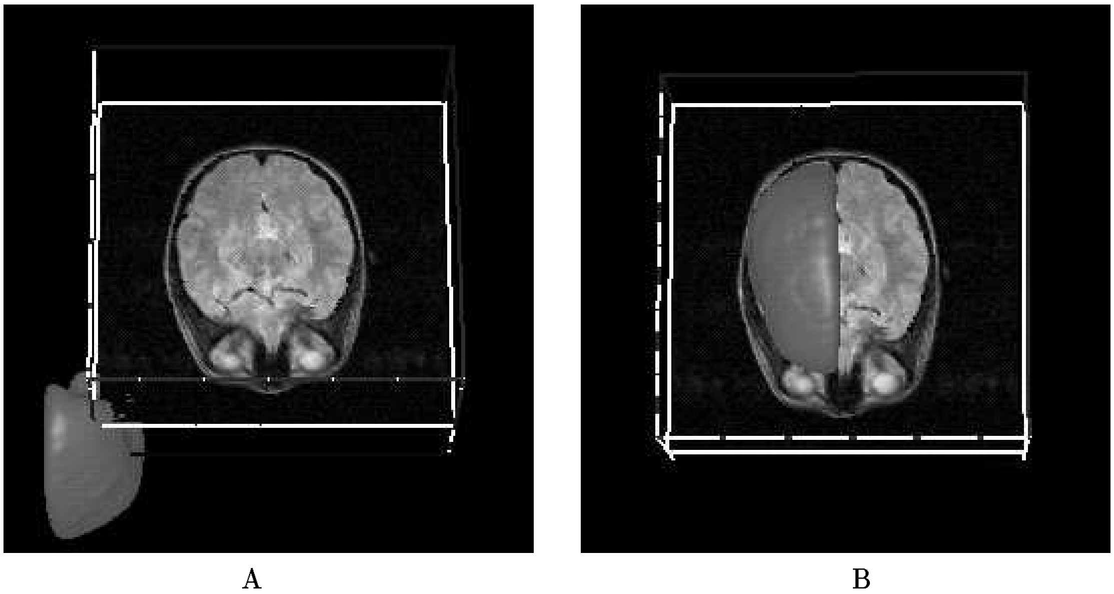


Figure 7.2: Template Initialisation A) The template is shown in its initial position. B) The mapping generated by the initialisation procedure.

and illustrates the effects of data truncation and the inclusion of unwanted cerebellar matter in the connected components.

The second example — Figure 7.3B) — shows a less satisfactory result: the plane estimate appears to be out by 10-15 degrees. Nonetheless, there is still considerable overlap between the template and the data. In part the poor estimate is induced by the lack of symmetry in the selected components near the base of the brain. In this case, the intersection of sulci with the imaging plane caused the cerebral component to "fragment", disturbing its symmetry. The rotational component observed in the previous example is also present.

There are occasional strong inhomogeneous intensity fluctuations which may result in spurious structure within the maximal component. Fortunately, the method is able to cope satisfactorily with such events: this error generally manifests as a small skew in the plane estimate, since the robust plane estimation will reject grossly perturbed mid-line estimates. Figure 7.4 illustrates this rejection mechanism. In general, the innermost slices provide the most reliable/consistent approximation to the MSP and these will be favoured by the plane fitting process.

While there are almost certainly ways to improve the accuracy, the implementation of such methods impinges on our segmentation task. By the simple expedient of thresholding and robust plane estimation, we are able to achieve satisfactory results: the initialisation will ensure that the data and model are always in reasonable proximity and that the model is correctly oriented. These are the only requirements we have for this stage of the segmentation process.

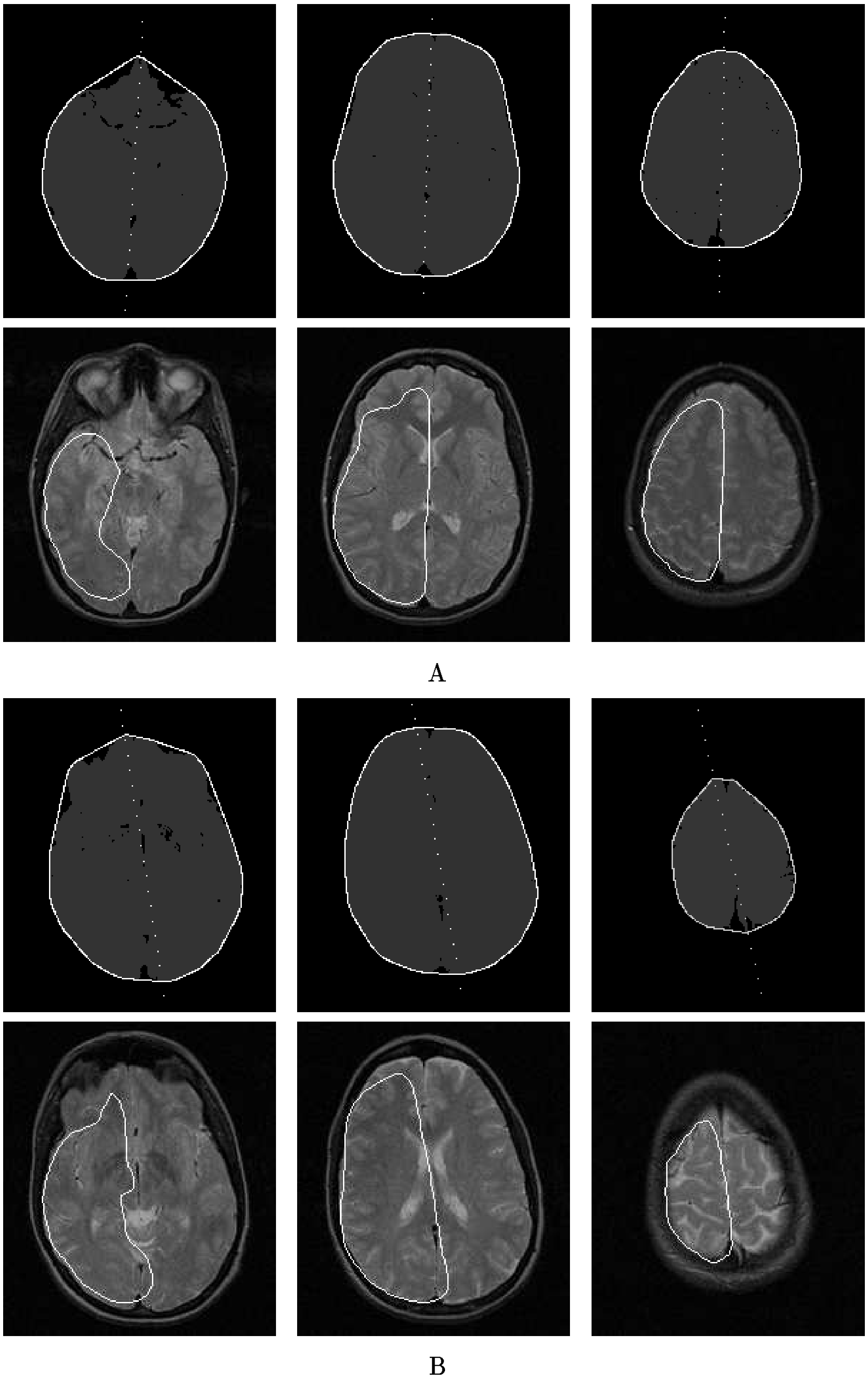


Figure 7.3: Computing the Initialisation A) The first row of images shows the convex hull for the thresholded image data on a number of slices, and the resulting symmetry axis estimate for each. Points from each symmetry axis are used to robustly estimate the MSP, which is used in conjunction with the thresholded data to compute the initialisation mapping. The second row of images shows the intersection curves of the initialised mesh with the corresponding slices. B) The MSP estimate may be quite poor in certain cases, but the general orientation and position of the initialised model will be correct.

A B

Figure 7.4: Robust Plane Estimation A) The image shows the mid-line estimate given by the PCA, as well as the intersection of the template with this slice. B) The bifurcation of the brain in the topmost slices may cause only one component to be selected, thus corrupting the mid-line estimate. By using a robust plane fitting procedure, these points will be rejected and we do not require additional heuristics to provide a satisfactory fit — the intersection curve is also shown on this slice. Observe that it lies where one would expect.

7.4 Computing the Surface Update

The surface update should provide an improved estimate of the underlying boundary. To achieve this, intuition suggests that each mesh vertex should be displaced towards the closest boundary point. In the case of sparse data, however, the closest boundary point may prove to be a completely inappropriate attractor. This section begins with an examination of the constraints imposed by our data.

7.4.1 The Implications of Truncation and Sparsity

MR image truncation poses a particularly difficult problem for surface recovery: we have no way of knowing the true geometry of the truncated surface. If we assume that the shape model provides a good description of the missing boundary, then the ASM framework should allow us to recover at least some of this information. In other words, since flat brains do not exist in practice, the (constrained) shape space should not admit such solutions. Of course, the resulting boundary geometry is still likely to vary somewhat from the underlying reality. It should be possible to minimise this discrepancy by ensuring that the segmentation is as accurate as possible for those regions in which boundary data is defined. Since the relative arrangement of hemispherical structure should be preserved, accurately isolating this component should enable us to recover a more representative surface in truncated regions.

Another issue which needs to be considered is the manner in which missing boundary data

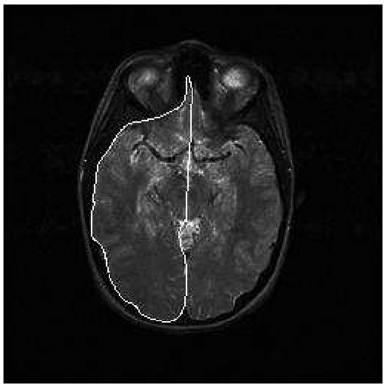


Figure 7.5: Poor Boundary Discrimination Over parts of the brain, the boundary we wish to extract may be invisible, or too diffuse. The white contour indicates the boundary representation we wish to recover. Observe that there are no strong internal features to guide the segmentation process: the model must provide a good interpolation over such regions.

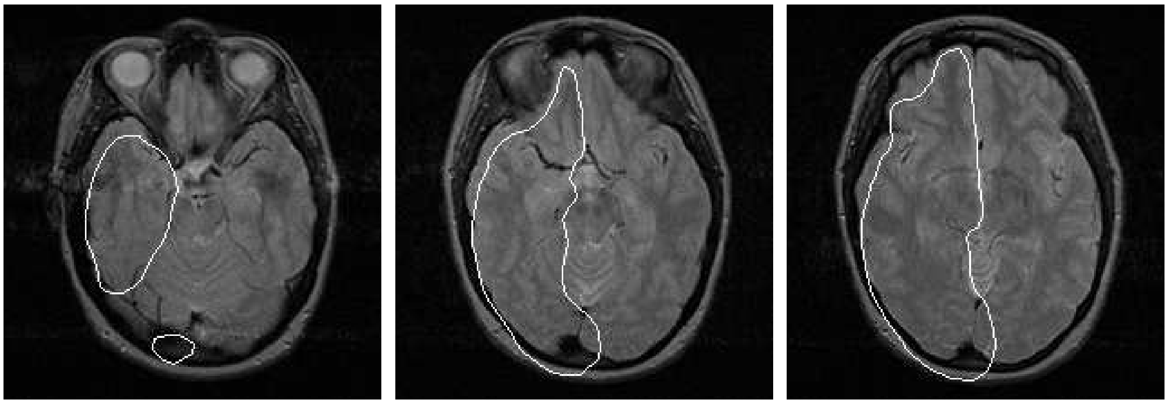
will be interpolated, particular over regions where no suitable boundary model could be constructed. The two boundaries in question are the cerebrum/cerebellum interface and divide between the hemispheres. Under the T2 protocol we investigated, neither was clearly distinguishable — Figure 7.5. In this case, it seems that the only option is to allow the model to interpolate the missing data. This is, after all, part of the function of constrained segmentation. Provided there are sufficient boundary points over the rest of the model surface, such interpolation should provide a physically meaningful solution.

To facilitate model interpolation, we apply a full affine segmentation, prior to engaging the ASM framework. This enables us to overcome poor initial placement of the surface model, and produces the optimal global (affine) alignment between the model and data[2] — Figure 7.6. The ICP algorithm is used to find the affine transformation from the boundary point set to the mesh vertex set. The inverse of this transformation then yields the map that transforms the mesh onto the data set. This procedure was adopted since there is always a reasonable closest mesh vertex for each boundary data point. The converse does not hold: the boundary data is sparse and may be entirely absent over the MSP.

7.4.2 Computing the Closest Point Deformation

The closest data point to any given mesh vertex defines an intuitive update target for that vertex. To improve the smoothness of the surface update, and to account for non-isotropic data point distribution, this displacement may be projected onto the surface normal [30]. Unfortunately, in many instances the resulting (unconstrained) mesh deformation is counter-intuitive, and does not preserve the general structure of the shape. This is particularly evident within truncated regions, where the mesh becomes flattened under the closest point

[2] Assuming that a sufficiently representative boundary point set can be extracted.



A

B

Figure 7.6: Affine (ICP) Segmentation A) The initialisation may result in a surface which is distant from parts of the brain boundary. B) Affine segmentation, improves this initial mapping significantly and brings the surface much closer to the boundary.

deformation. An example of a truncated MR image was shown in Figure 2.8.

Since sparse boundary data exists within each image plane only, the closest data point to a given vertex must be located on such a plane. As a consequence, there will be a tendency for vertices to gravitate towards the closest slice plane. If unchecked, this behaviour will result in a stepped appearance over parts of the mesh surface. These problems are illustrated in Figure 7.7.

These examples show that the closest point does not provide the most intuitive target in all cases. This was observed by Feldmar [33], who introduced an alternate measure of distance which includes surface normal and curvature information. Unfortunately, we do not have access to this information for the extracted boundary points. Rather than modifying the distance metric, we decided to adopt the following strategy:

Closest Point Interpolation The closest data point to a given vertex, is defined in terms of the closest data points on each of the adjacent image planes. As the vertex in question

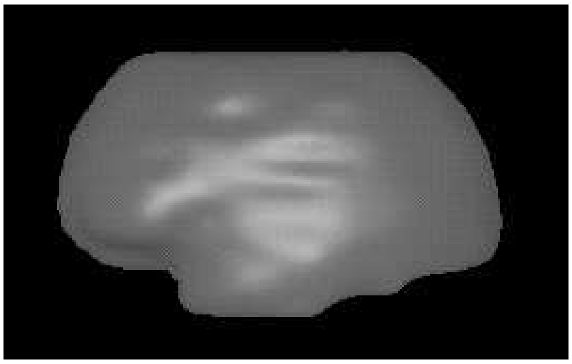


Figure 7.7: Closest Point Deformation If each vertex moves towards the closest data point, the final segmentation may be unacceptably stepped or truncated. This behaviour is a direct consequence of the planar arrangement of the boundary data, and shows that a simple closest point deformation is not appropriate in this case.

draws nearer to the relevant image plane, the corresponding closest point plays a more prominent role in the estimation of the interpolated closest point.

Distance Scaling The displacement required to move the vertex towards the estimated closest point is scaled according to the distance between the vertex and the adjacent image planes. As the distance between the vertex and the (data bearing) image planes increases, the displacement is scaled down.

The interpolation scheme ensures that there are no abrupt transitions in the closest point from one slice to another. The interpolated closest point, $\mathrm{CLPT}(\mathbf{P})$, to the vertex $\mathbf{P}$ is defined as:

$$\mathrm{CLPT}(\mathbf{P}) = \mathrm{clpt}(\mathbf{P}, \mathrm{i}) + \mathrm{s}.\left(\mathrm{clpt}(\mathbf{P}, \mathrm{i}+1) - \mathrm{clpt}(\mathbf{P}, \mathrm{i})\right), \tag{7.1}$$

where $\mathrm{clpt}(\mathrm{P}, \mathrm{i})$ is the closest point to $\mathbf{P}$ on slice i, and s is given by

$$s = (\mathbf{P} - \mathrm{clpt}(\mathbf{P}, \mathrm{i})) \cdot \frac{\mathrm{clpt}(\mathbf{P}, \mathrm{i}+1) - \mathrm{clpt}(\mathbf{P}, \mathrm{i})}{||\mathrm{clpt}(\mathbf{P}, \mathrm{i}+1) - \mathrm{clpt}(\mathbf{P}, \mathrm{i})||^2}. \tag{7.2}$$

In other words, $\mathrm{CLPT}(\mathbf{P})$ is the closest point to $\mathbf{P}$ on the line defined by the two neighbouring closest points. In this formulation, the vertex is assumed to lie between slice i and $i+1$.

The closest data point is constrained to lie within a radius of $2d$, where d is the inter-slice gap. This ensures that vertices in surface regions with no boundary data “beneath” or “above” will not erroneously select a distant point on the adjacent slice. Such a limit is also necessary to reject blatant mismatches, such as those which can occur in the absence of MSP boundary data. The use of such a distance constraint is not overly restrictive, since global deformation will have been accounted for in the initial affine registration phase.

As a consequence of the distance constraint, either one of the clpt operations may fail. If both searches succeed, then the interpolation scheme is applied. If only one search succeeds,

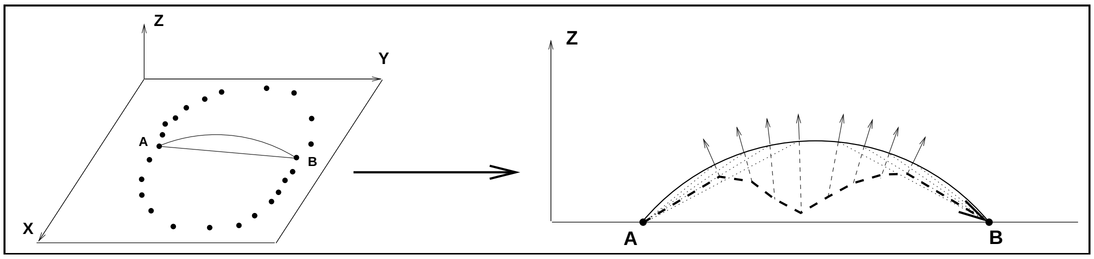


Figure 7.8: Deformation Cusp For surface regions in which only one adjacent image plane has data, the closest data point may switch suddenly from one side of the object to the other as one moves across the surface. In the figure, a cross-section of such a region is shown, along with dotted lines to indicate the effective closest data point for each selected surface point. The closest point displacement along the surface normal then results in a central dimple, since two different closest points are active.

the returned point is taken as the closest point. If both searches fail, then the vertex remains stationary.

A linear interpolant was chosen since this method requires no higher order knowledge, such as local curvature and requires only two data points. The shape constraints we impose encode local curvature, and should induce the necessary adjustments to ensure that the surface update is consistent.

In regions where only one closest point is detected, such as the top and bottom of the brain, vertices will still move towards the closest image plane. Furthermore, if the closest point deformation is projected onto the vertex normal, a central cusp may develop in such regions, Figure 7.8. This occurs because the closest point switches suddenly from one side of the closest slice to the other, as one moves across the surface. By weighting the vertex displacement function appropriately, we can ensures that a more natural deformation takes place. The function we use is

$$W(\mathbf{P}) = \begin{cases} \frac{1}{2}\left[1 + \cos 2\pi \left(\frac{\|P_z - z_c\|}{d}\right)\right], & \text{if } \|\mathrm{P_z} - \mathrm{z_c}\| < \frac{\mathrm{d}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{7.3}$$

where z_c is the z coordinate of the closest image plane. This is a restricted form of the equation used to moderate the data force attraction in Section 7.4.4.

This weighting function is only applied when we cannot compute a closest point for both neighbouring slices. It gradually reduces the effectiveness of the closest point attraction to zero at a distance of $\frac{d}{2}$ from the closest slice. Vertices located beyond this point are not attracted by the data at all, but are subject to the shape constraints. By applying these rules, we can generate a mesh deformation which is largely in keeping with our intuition.

7.4.3 Stage I: ASM Segmentation

The strategy employed by active shape model segmentation is two-fold:

1. Compute the global transformation which maximally aligns the current model estimate, $\mathbf{M}_i$, with the nominally updated model, $\mathbf{M}_i^*$.
2. Project the residual into the PDM/PCA space, to provide an updated shape template estimate, $\mathbf{x}$, which, when subjected to the global transformation, will produce a better match, $\mathbf{M}_{i+1}$.

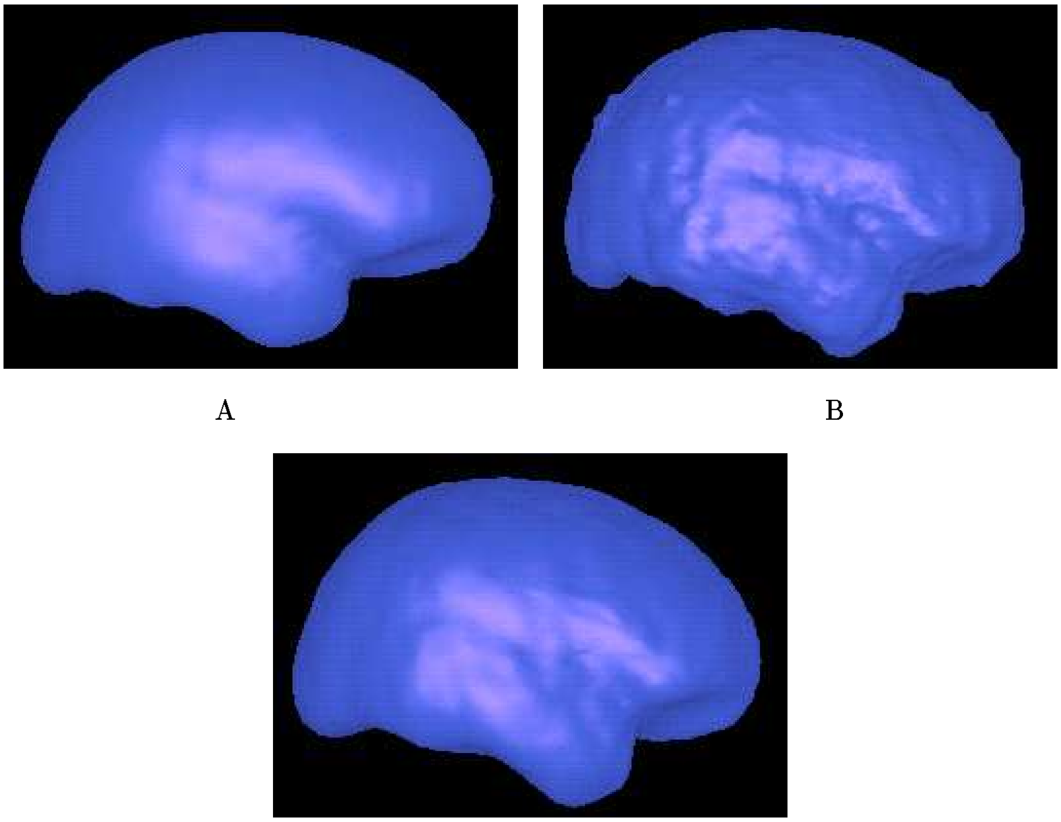
This scheme is applied iteratively until some termination criterion is satisfied. In our implementation, the shape template, $\mathbf{x}$, is positioned at the origin and oriented so that its principal axes correspond to the coordinate axes. At each iteration, $\mathbf{x}$ is modified, but its overall position and orientation will remain similar. The updated template is mapped into the data space to yield the current shape model approximation, M. The iteration is terminated when the maximum vertex displacement is less than a specified threshold, or the iteration count is exceeded.

The algorithm may be stated more formally as follows:

Algorithm 7.1 (ASM Update Scheme)

1. Use the detected boundary points to obtain an unconstrained estimate of the shape instance, $\mathbf{M}_i^*$.
2. Compute the global transformation, T_{i+1}, which best aligns the current template, $\mathbf{x}_i$, with the estimated shape: $T_{i+1} = \mathrm{argmin}_{\mathrm{T \in S_T}} ||\mathrm{T}\mathbf{x}_\mathrm{i} - \mathbf{M}_\mathrm{i}^*||_{\mathrm{E_N}}$
3. Compute the local shape template deformation, $d\mathbf{x}$, required to provide a better match under T_{i+1}:
   $d\mathbf{x}_{i+1} = T_{i+1}^{-1}\mathbf{M}_i^* - \mathbf{x}_i$
4. Project $d\mathbf{x}$ into the PDM subspace and compute the best constrained template approximation:
   4.1 $d\boldsymbol{\lambda}_{i+1} = P^T d\mathbf{x}_{i+1}$
   4.2 $\mathbf{x}_{i+1} = \mathbf{x}_i + P\ d\boldsymbol{\lambda}_{i+1}$
4. Compute the current shape model approximation, M_{i+1}:
   $\mathbf{M}_{i+1} = T_{i+1}\mathbf{x}_{i+1}$.

In this formulation, which is slightly different from one outlined in [24], we have allowed for the possibility of a more general class of transformations, S_T. We start by taking our current mesh surface, $\mathbf{M}_i$, and applying the boundary detection scheme to arrive at a provisionally improved shape estimate, $\mathbf{M}_i^*$. We then search for the global transformation, $T_{i+1} \in S_T$, which minimises the (Euclidean) distance between our nominal update and our current template. As stated in Chapter 4, we utilise an affine transformation to normalise our data sets prior to performing the PCA. This means that to properly employ the PDM description we



A B

C

Figure 7.9: ASM Constrained Segmentation A) The initial template. B) The closest point deformation, which represents the unsmoothed, unconstrained update. C) The final constrained segmentation. Note that the basic shape of the unconstrained solution is preserved.

must ensure that our new transformation comes from the same space. The ICP algorithm is used to compute the required affine map.

This global transformation will not, in general, provide the best approximation to the shape instance. The updated shape estimate, $\mathbf{M}_i^*$, is transformed into the model frame (using T_{i+1}^{-1}) and a difference vector is computed for each corresponding vertex. The set of vertex residuals is projected into the PCA space to provide the parameter update, $d\boldsymbol{\lambda}_{i+1}$, which will move the template to a more accommodating shape within the PCA subspace. The matrix P contains the eigenvectors of the PCA arranged column by column, whilst the parameter vector $\boldsymbol{\lambda}_{i+1}$ specifies the contribution of each eigenvector to our current template estimate.

An example of this scheme in action is shown in Figure 7.9. The initial template is updated according to the modified closest point deformation. This tentative update is then projected into the PDM shape space to provides the best permissible shape approximation.

The constrained segmentation produced by the ASM, can only represent shapes in (or very near to) the learned shape space. Since the range of normal structural variability within the brain is large, it is unlikely that the ASM will achieve a perfect segmentation of patients deemed to be healthy, although the results may be very good. The size of the training set will obviously be an important determining factor in the accuracy of the segmentation. These observations, coupled with our desire to represent structural abnormality, necessitate a further segmentation step.

7.4.4 Stage II: Simplex Segmentation

The refinement of the initial segmentation should enable us to fit the boundary data more closely, whilst ensuring that we do not stray too far from our preliminary result. Where there is no boundary data — between the image planes and beyond the FOV — the segmentation should, as far as possible, maintain the current shape estimate.

We considered using the spline ICP scheme (Chapter 4) to obtain a smooth surface update. Unfortunately, experiments showed that this approach produced unacceptable deformations over the MSP, where there was no data. At the very least, we require that the surface remain stationary if no local boundary data is available. This spurious deformation is a consequence of the global energy minimisation underlying the ICP computation[3]. There is also no explicit mechanism for enforcing shape constraints, beyond the inherent smoothness of the spline warp. Since any global transformation, even one with many degrees of freedom, is likely to modify the mesh when it should not, we have opted for a more local approach.

The simplex template segmentation scheme combines global and local deformation, subject to the constraints imposed by the simplex template, [52]. Since the ASM segmentation effectively eliminates the need for additional global deformation, and given the observations above, we shall use local deformation only. In our implementation, the simplex template is derived directly from the result of the constrained segmentation, providing a patient-specific model for further refinement. Local deformation is based on the closest point scheme discussed in Section 7.4.2. The simplex template uses the force model introduced in Section B.3. Each vertex, $\mathbf{P}_i$, of the simplex mesh evolves under a simple, discretised Newtonian model:

$$\mathbf{P}_i^{n+1} = \mathbf{P}_i^n + \mathbf{F}_{\text{ext}} + \mathbf{F}_{\text{int}}, \tag{7.4}$$

where the $\mathbf{F}_{\text{int}}$ and $\mathbf{F}_{\text{ext}}$ represent the internal and external forces, respectively, acting on the vertex. We use undamped dynamics, and set the vertex mass to unity.

[3] Since no energy penalty is incurred when moving the surface in regions where there is no data, there is no guarantee that the surface in such regions will remain stationary.

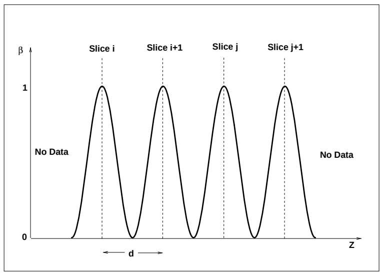


Figure 7.10: External Force Decay The strength of the external force is modified according to a periodic function (sinusoid) which achieves its maximum on the image plane and its minimum at the centre of the slice gap. The function varies with distance along the acquisition direction only. If the vertex under consideration has only one (or no) closest points, then the periodic behaviour is broken and the function does not rise above zero beyond the image plane on which the data point is located.

The external force is given by

$$\mathbf{F}_{\text{ext}} = \beta(\mathbf{P}_i)\ \left[(\mathbf{P}_i - \text{CLPT}(\mathbf{P}_{\text{i}})) \cdot \mathbf{N}_{\text{i}}\right] \mathbf{N}_i, \tag{7.5}$$

where $\mathbf{N}_i$ is the surface normal at vertex $\mathbf{P}_i$ and $\beta(\mathbf{P}_i)$ is a function which modifies the strength with which the data attracts the vertex.

The internal (regularising) force is based on the local simplex angle, ϕ_i:

$$\mathbf{F}_{\text{int}} = \frac{\alpha}{2}(\tilde{\mathbf{P}}_i - \mathbf{P}_i), \quad \alpha \in [0, 1], \tag{7.6}$$

where $\tilde{\mathbf{P}}_i$ is the position of the vertex using the updated value of ϕ_i. The various possibilities by which ϕ_i can be constrained, were discussed in Section B.3. In the case of the shape constraint, one merely insists that the simplex angle at each vertex remain constant. The appropriate simplex angle is obtained from the corresponding vertex in the simplex template. In the absence of external forces, the mesh will eventually relax back to the template configuration. One may also introduce a scale parameter, r, which defines the vertex neighbourhood, $N_r(\mathbf{P}_i)$, over which simplex angles should be averaged [30, 52]:

$$\tilde{\phi}_i = \frac{1}{|N_r(\mathbf{P}_i)|} \sum_{i \in N_r} \phi_i. \tag{7.7}$$

The parameter r specifies the topological distance between the vertex and its neighbours. Two vertices are a distance r away if the shortest path connecting them contains only r edges.

In the original framework, α and β were constant for the entire mesh (or designated zones), although the external force was modified by a fixed inverse square decay law. In order to reduce the number of parameters in the system, we have fixed α at unity, and only allowed

β to vary. To deal with the special structured nature of our boundary data, we define β as follows:

$$\beta(\mathbf{P}) = \frac{\beta_0}{2}\left[1 + \cos 2\pi\left(\frac{||P_z - z_{\mathrm{cl}}||}{d}\right)\right]^n, \quad \beta_0 \in [0,1], \quad n \in \mathbb{Z}. \tag{7.8}$$

This function is shown in Figure 7.10. The function $z_{\mathrm{cl}}(\mathbf{P})$ returns the z coordinate of the closest image plane, whilst n may be used to vary the fall-off of the inter-slice decay. For vertices which have only one (or no) closest point, the function is clamped to zero if the z coordinate is located beyond the centre of the slice gap.

This function ensures that the simplex template will dominate in regions which are distant from the boundary data, particularly at the extremities of the brain. A sinusoid was chosen since it is naturally periodic and provides the desired decay behaviour.

By applying the simplex constraint at each step, we can ensure that a reasonable local deformation is obtained. Points which are not in the vicinity of data — such as those on the MSP — will remain stationary, provided the simplex averaging neighbourhood is sufficiently local. We have used a fixed neighbourhood size, which seems to provide good results.

7.5 Results

Results are shown using 3D visualisations as well as mesh-slice intersection curves. Once the governing parameters have been set, the segmentation process is automatic. The initialisation step may, however, require user intervention. This would be necessary if the initial affine map placed the template too low down in the image volume, a possibility if the largest connected components were incorrectly selected during bounding box estimation. In this case, one need just use the mouse interface to translate the mesh into a better position.

7.5.1 The Template

A separate template was used for each cerebral hemisphere. The decision to use two models, rather than a single model, was based on the intended purpose of the surfaces: laterality and asymmetry measures. This task is facilitated if one has two surfaces in which the plane of asymmetry is made explicit (see Chapter 3).

A total of 20 (male and female) normal controls[4] were used to build the templates. This number was dictated by circumstance, since only 20 (11 male, 9 female) subjects had been manually segmented at the time this work was done. Control subjects were age, sex and handedness matched to the available patient data (as far as was possible). Control subjects

[4]Supplied by the EC Biomorph Project

have no history of mental illness and have passed a verbal test (DSM I/II/III) which establishes whether or not they exhibit any of the symptoms common to a variety mental disorders. Handedness was predominantly right (85%), and the age range was 20—60, with more than 80% of the subjects being 35 or younger.

Seventeen PDM modes were sufficient to represent 95% of the training sequence variability. Because the arachnoid surfaces are smooth, and the contribution of the high frequency modes is relatively small, one would not expect this number to grow proportionately if the training set were, say, doubled in size.

7.5.2 Test Database

The sparse MR images used in the following experiments were taken from the sparse database of Section 2.2.2.

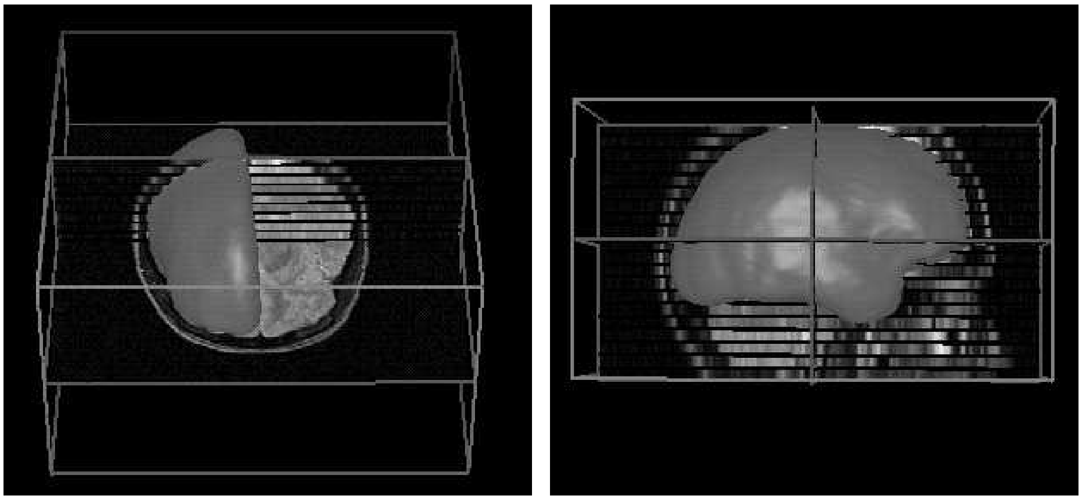
7.5.3 Parameters

There are a number of parameters which must be specified before the segmentation can proceed.

The "search scale" for boundary detection was set at 35mm. Intensity profiles extracted at this scale provided sufficient information to enable a good model match over most of the brain. If this value is too large, the intensity sample may include data from the same boundary on the other hemisphere, which can confuse the matching process. The boundary rejection threshold was set at 2mm, based on empirical evidence. Using this level of rejection ensured that about almost all spurious matches were rejected. Two validation profiles were used.

Partial volume refinement was disabled. The improvement in boundary localisation is small except in the polar region. Unfortunately, as discussed in Section 6.6, the matched boundary models from these peripheral slices often provide a poor approximation to the underlying tissue distribution, and the resulting prediction is likely to distort matters further.

The simplex segmentation step requires the specification of 2 parameters. The base strength with which the data attracts the surface, β_0, was set to 0.3: this value provides a strong level of attraction, but is not sufficient to override the shape constraints. The smoothing neighbourhood was set at $r = 4$. For each step of the simplex segmentation scheme, we applied 20 iterations of the Newtonian model update. This was necessary, since the vertex update scheme converges slowly for a given set of vertex displacements.



A B

Figure 7.11: Simplex Template Segmentation A, B) Two views of the final extracted surface embedded in the MRI volume. Despite the coarseness of the MR data, a reasonable cerebral surface is extracted.

7.5.4 Sparse Segmentation

The bulk of the scans in our database are sparse T2 axials. In certain cases, there are also high resolution T1 coronals, acquired at the same time-point, which may be used to "validate" the low resolution segmentation. The first example, Figure 7.11 shows a visualisation of the simplex segmentation for a sparse scan. The MR data has been rendered accurately, so that the inter-slice gap and voxel intensity are properly represented. Observe that the fit is quite good over most of the surface, although there are noticeable differences on the topmost image slice. The intersection of the surface mesh with each slice is shown in Figure 7.12. The mid-saggital line is well represented in this case, as is the greater part of the cerebral boundary. However, on the top-most slice, and over regions in which the cortex is very narrow, the surface does not provide a very good approximation.

Using software provided by the Biomorph project[5], it was possible to register low resolution with high resolution scans, taken at the same time-point. This provides a simple means of checking the accuracy of the low-resolution surface approximation. A visualisation of the original surface mesh and its registered counterpart is shown in Figure 7.13. In this case, both the left and right hemispheres have been segmented from the sparse scan. The mesh-slice intersection curves for the sparse scan are shown in Figure 7.14, whilst the intersection curves for the registered high resolution surface are shown in Figure 7.15. As with the previous example, the surface estimate is good in all regions, except on peripheral slices, and where the cortex is very narrow.

[5]Produced by the Laboratory for Medical Imaging Research, K.U. Leuven, in Belgium.

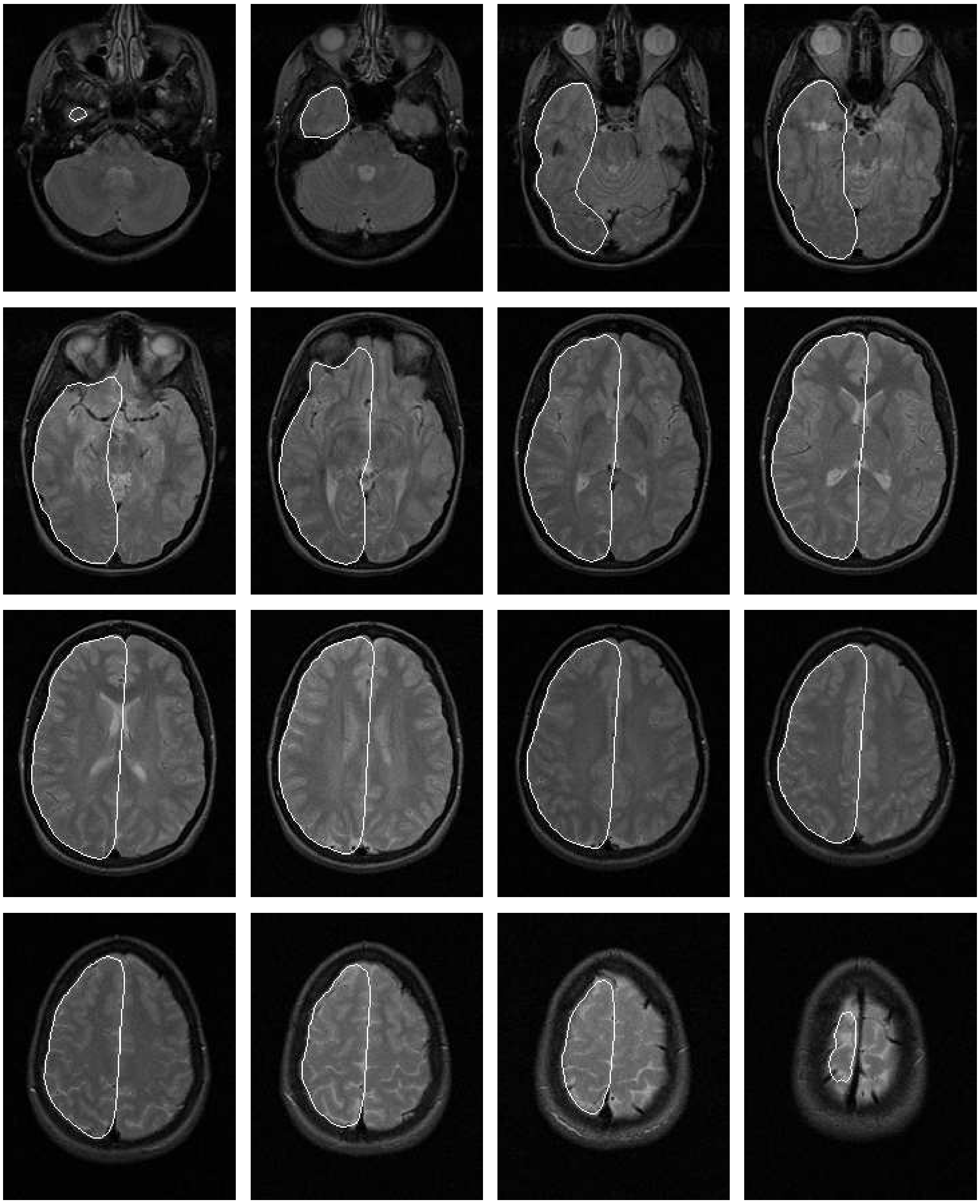


Figure 7.12: Cerebral Contours — Example 1 The sequence of contours shown represent the mesh-slice intersection curves for the surface shown in Figure 7.11. The cerebral surface is well approximated over most of the surface, although there are noticeable deviations on the polar slices and in regions where cortex is very narrow.

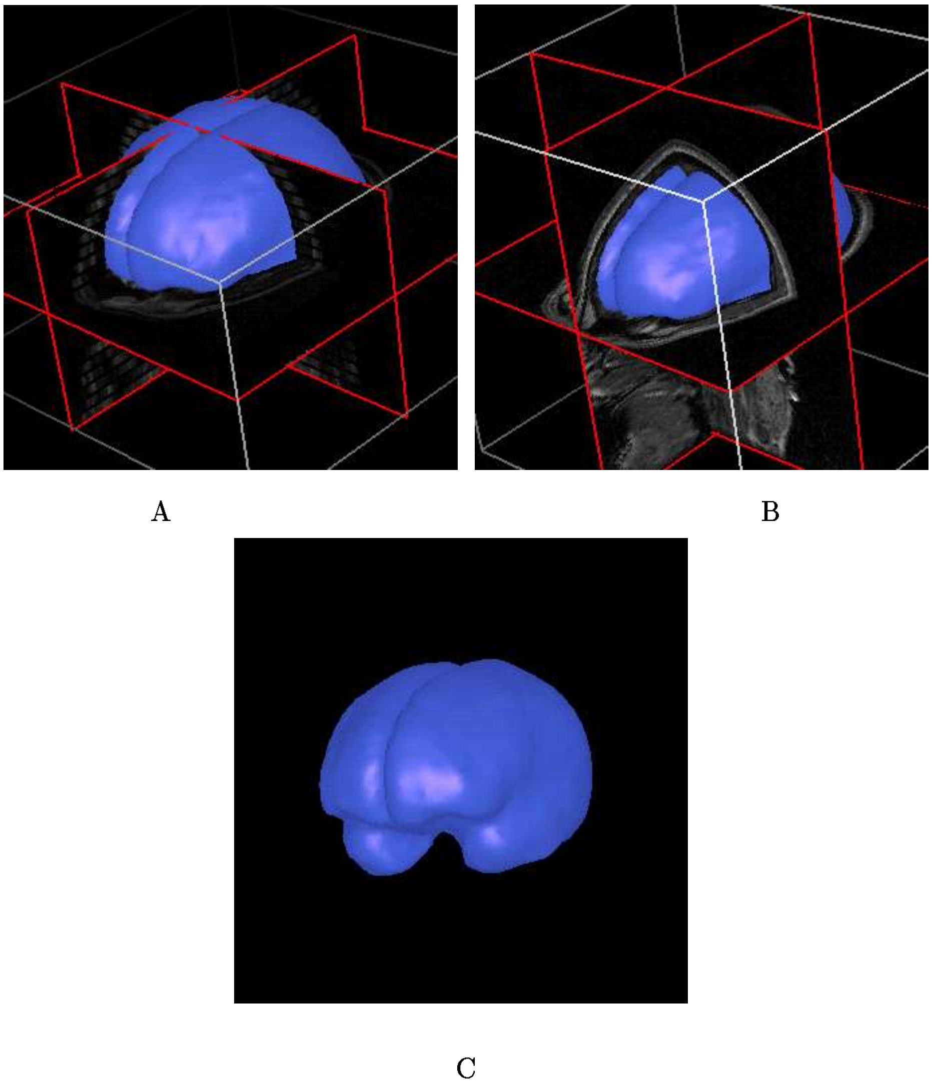


Figure 7.13: Full Cerebral Surface An example in which both the left and right hemisphere have been segmented. A) The low resolution segmentation — the data set is T2 axial. B) The surfaces estimated from the sparse T2 scan have been mapped onto a high resolution T1 coronal scan taken at the same time-point. C) An oblique view of the extracted surfaces. Cross-sections of these surfaces are shown in Figure 7.14 and Figure 7.15.

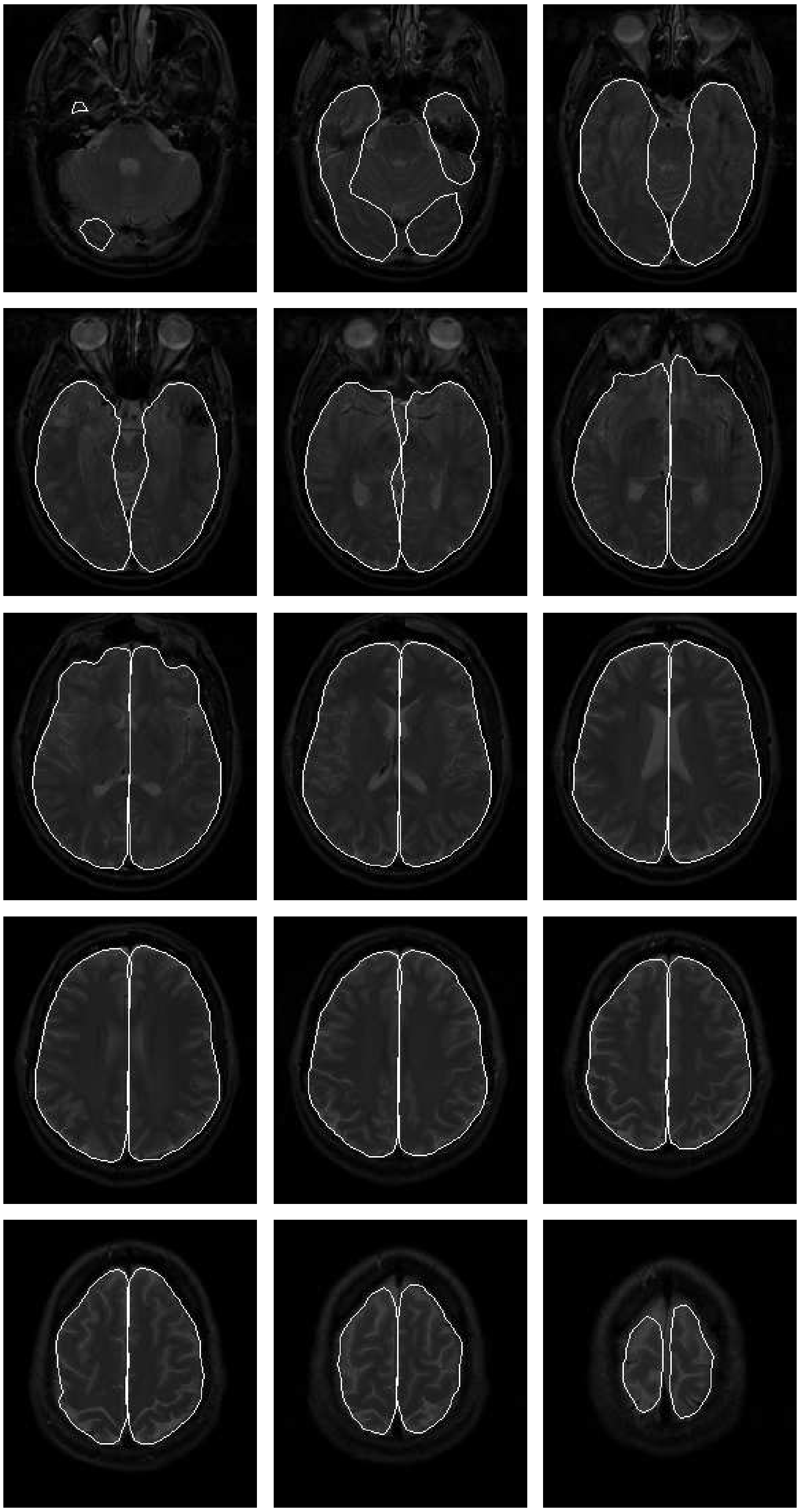


Figure 7.14: Mesh Contours in Sparse Data — Example 2 A detailed view of the sparse segmentation corresponding to Figure 7.13. The mesh-slice intersection boundary is indicated in white. Observe that the boundary is well represented over most of the surface, except for two "horns" above the eyes, and the top and bottom-most slices.

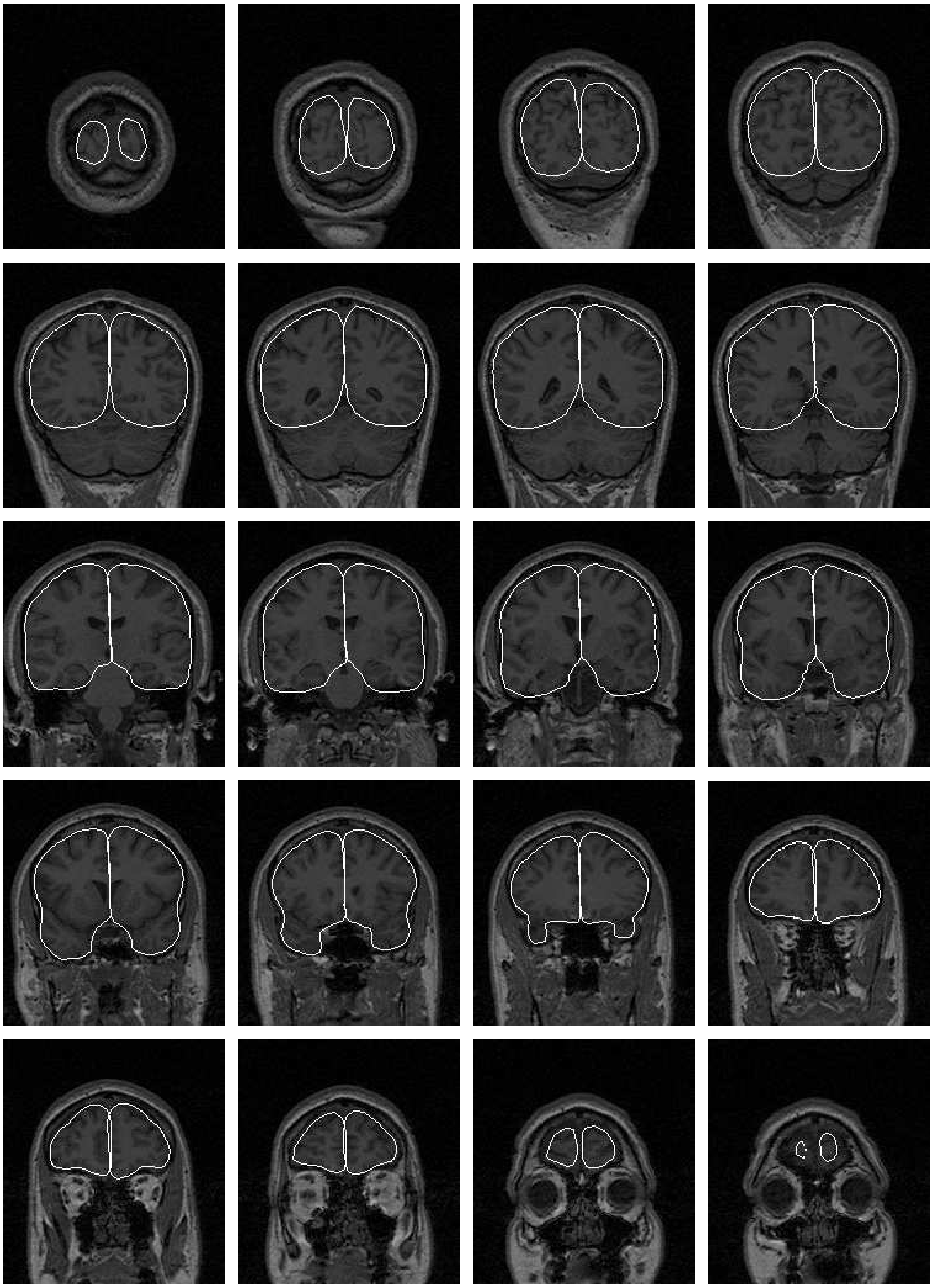


Figure 7.15: Mesh Contours in Dense Data — Example 2 This sequence of images shows the slice-mesh intersection curves for the transformed mesh in shown Figure 7.13B. Observe that the boundary provides a surface very much like the arachnoid i.e. not following deep sulci. The similarities between these boundary curves and those shown in Figure 4.3 are clear, indicating that our results are consistent.

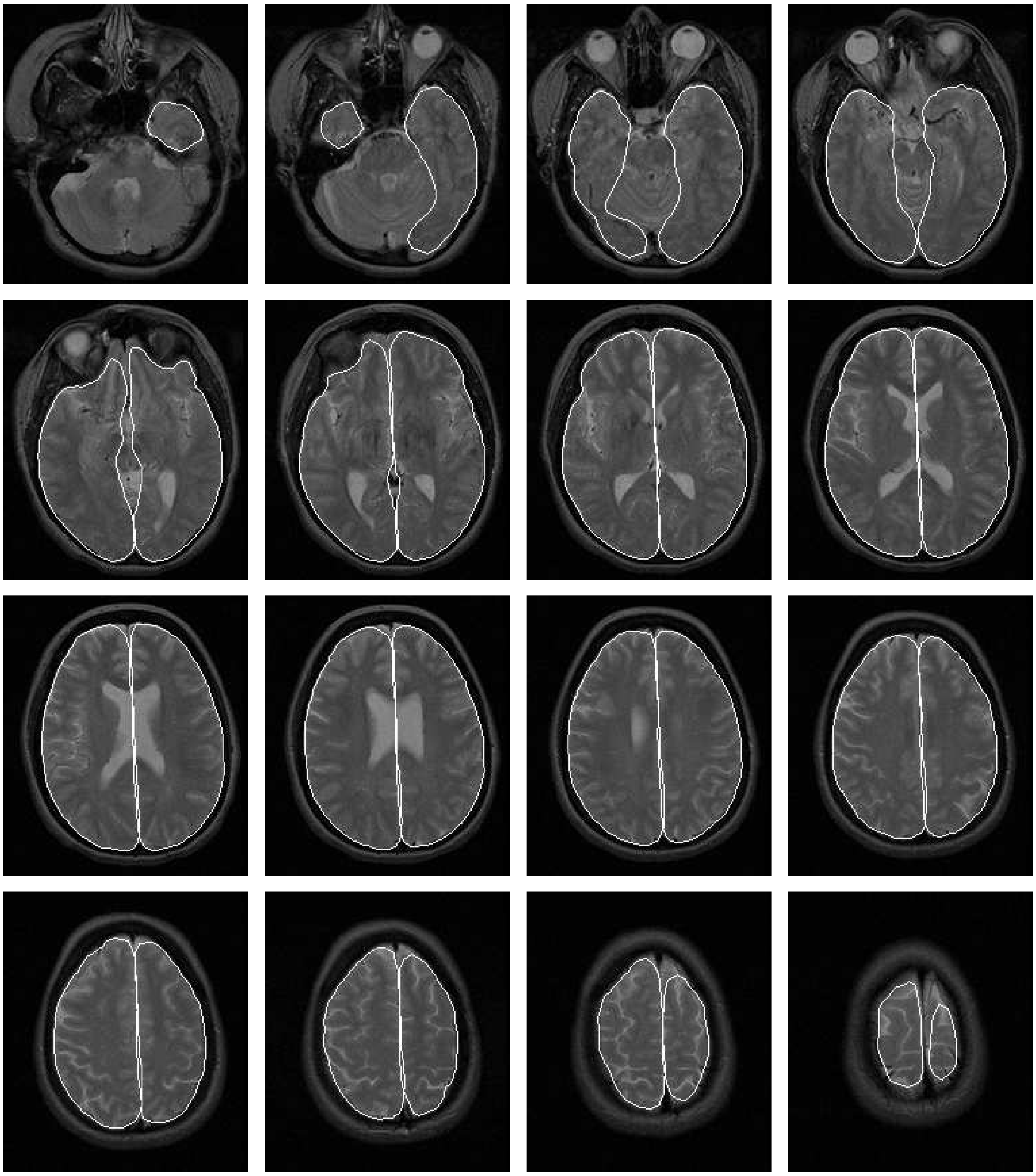


Figure 7.16: Mesh Contours in Sparse Data — Example 3 In this example, the patient's head is rotated away from the MRI coordinate frame. The resulting asymmetry of the imaged brain is evident within each slice. The segmentation scheme recovers a reasonable approximation to the cerebral surface, but has missed some peripheral detail. The registered surface is shown in Figure 7.17.

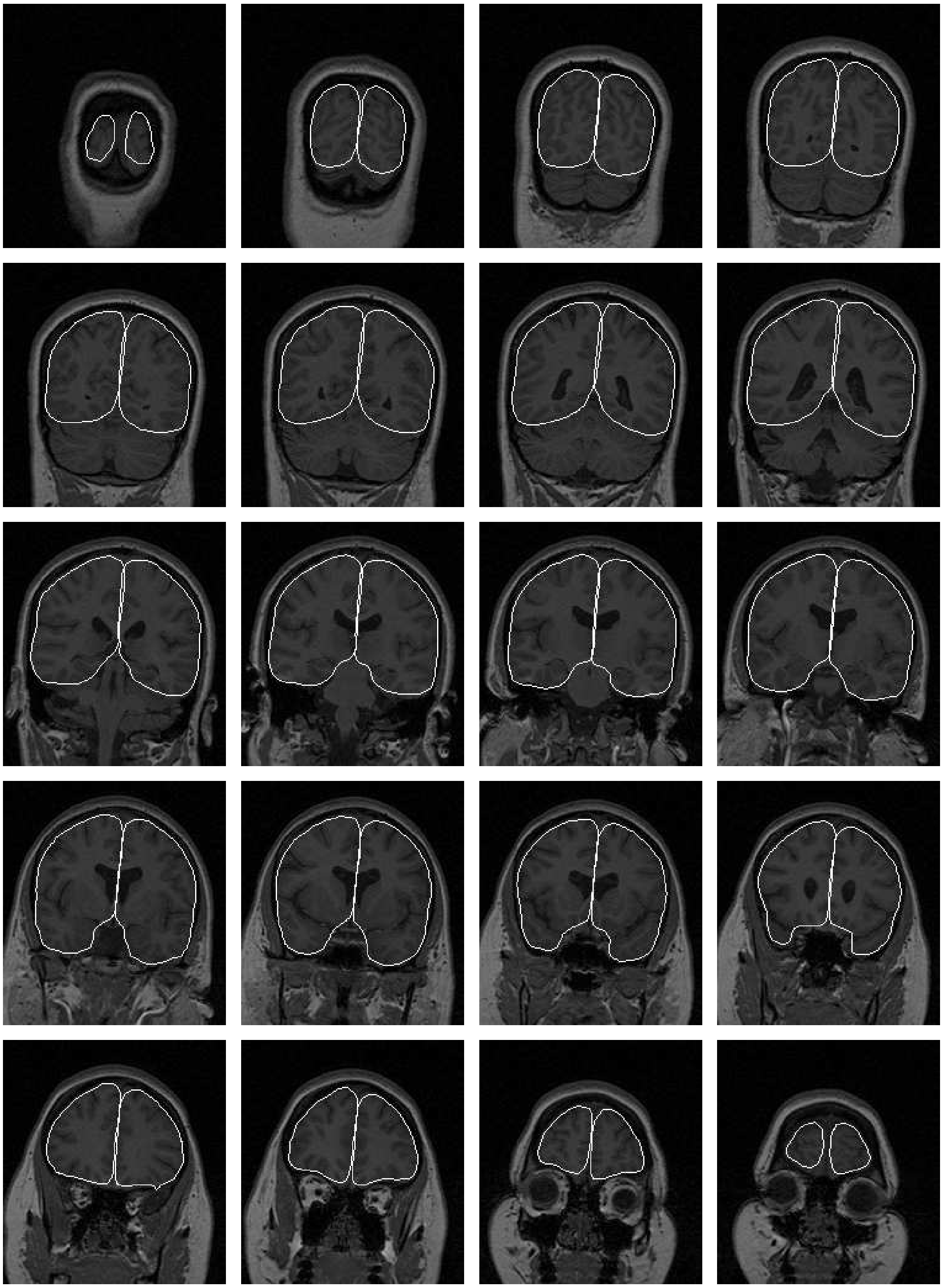


Figure 7.17: Mesh Contours in Dense Data — Example 3 A number of mesh intersection curves are shown for the surface recovered in Figure 7.16. The head tilt is clearly apparent. Observe that the MSP estimates for each hemisphere overlap somewhat. This occurs because their is no boundary data to constrain the mesh in this region, so errors arising during the initial global fitting stage are propagated.

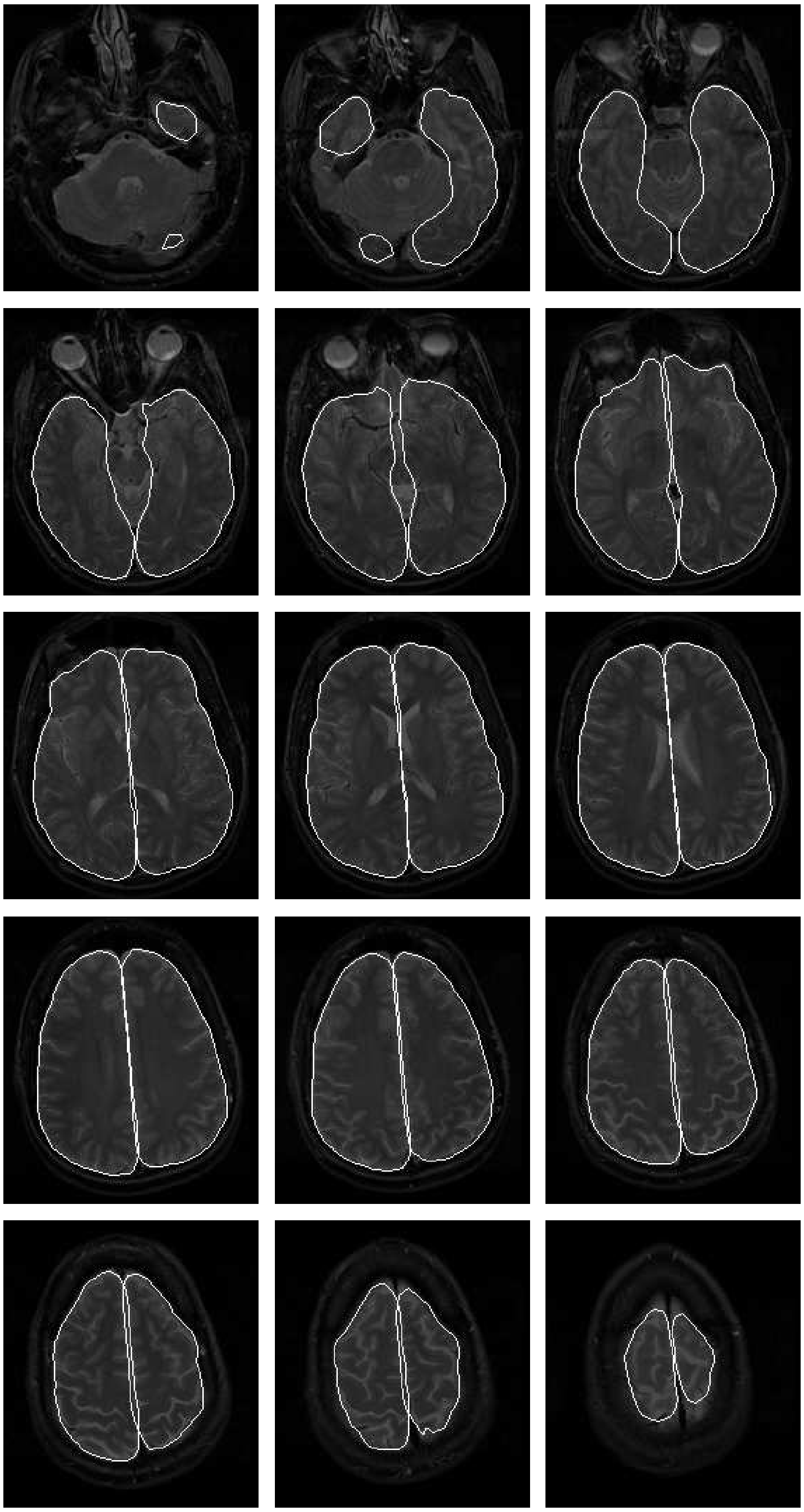


Figure 7.18: Mesh Contours in Sparse Data — Example 4 This example shows a significant amount of MSP overlap, as well as tissue underestimation on the peripheral slices. Tissue underestimation is caused by a combination of boundary mismatches and the model smoothing. The former arises mainly at the top of the brain, where CSF filled sulci complicate matching, whilst the latter occurs near the base of the brain, where the cerebral cross-sections change rapidly from slice to slice.

Another example for which registration information was available, is shown in Figure 7.16 and Figure 7.17. In this case, the patient's head was not well aligned with the MR coordinate frame, resulting in rather asymmetric cross-sections. The model provided a good approximation to the cerebrum-cerebellum interface, and also excluded the brain stem. As before, a sliver of cortical tissue between the eyes has been missed, as well as some detail on peripheral slices. The high resolution images shows that the MSP exhibits some overlap too.

The final example, Figure 7.18, exhibits many of the same features as the previous examples. As before, the interpolated boundary satisfactorily excludes the cerebellum and brain stem. In this example, however, the MSP shows considerable overlap. There is also a significant underestimation of the cortical boundary on the bottom-most slice.

7.6 Discussion

Qualitatively, the simplex segmentation produced quite good results. Of course, such assertions need to be accompanied by quantitative validation. In this case, however, validation requires that the clinician invest a significant amount of time tracing boundaries. Until this can be arranged, qualitative assessment must suffice.

In order to facilitate qualitative validation, a small set of T2 scans for which we had high resolution T1 coronal data, was identified. Using existing software based on mutual information registration, it was possible to compute the affine map that transformed the surface from the sparse to the dense image[6]. This showed that the extracted surfaces were indeed consistent, and provided a good approximation to the arachnoid. The results were unaffected by patient-machine alignment.

In general, the (nearly invisible) boundary separating the cerebrum from the cerebellum and brain stem was well represented, showing that the model interpolation was anatomically consistent in this region.

There are certain regions of cortex, however, which are not well recovered. Inspection of the intersection curves in the sparse image sequences shows a significant and consistent underestimation of the cerebral boundary at the top and bottom of the brain. There is also a underestimation of the narrow ridges of cortical tissue which lie between the eyes. In the latter case, the underestimation is caused primarily by the nature of this structure: it is almost parallel with the imaging plane, and disappears from one slice to the next. Furthermore, since the structure is narrow, the model constraints may not allow a suitable degree of surface penetration.

[6] The registration accuracy is reportedly $0.5 * V$mm, where V is 7mm in our case. But this number assumes isotropic voxel dimensions — visually the registration seems to be more accurate than the predicted 3–4mm.

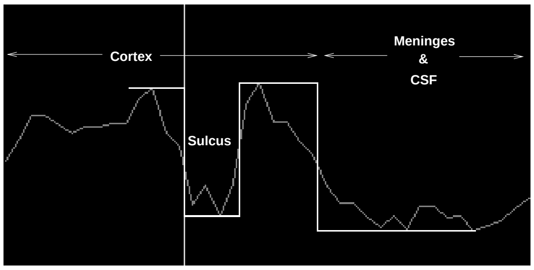


Figure 7.19: Sulcal Mismatch The intensity profile shown was extracted from the topmost slice of a sparse T2 image. In this case, a deep valley, corresponding to a sulcus filled with meningial matter, has resulted in an incorrect boundary match (the selected edge is indicated by the vertical line). The correct edge is the one separating the cortex from the "meninges". When such sulci are filled with CSF, we get a prominent ridge, which can also interfere with the boundary matching process. Such spurious matches will not be excluded by the rejection mechanism if the sulcus runs parallel to the brain boundary.

The boundary under-estimation at the top and bottom of the brain can be ascribed to two factors: poor boundary identification and poor shape model representation. As we have mentioned previously, the boundary models which work so well over the rest of the brain, do not function particularly well on the topmost slices. Consequently, the model only has a sparse set of boundary point data, some of which are incorrect because the rejection mechanism has failed. One would expect the shape model to provide some relief, since the bulk of the cerebral boundary is well recovered. Unfortunately, the shape model seemed unable to recover a better approximation in this region, regardless of the constraints imposed by the rest of the data. This is somewhat disappointing, but given the small number of training examples, not entirely unreasonable.

Although the thickness of the underestimated tissue regions is only a few millimetres, one would like to obtain as accurate a boundary representation as possible. The use of in-plane boundary detection is based on the assumption that part of the mesh surface will intersect the image plane in the vicinity of the boundary of interest. Unfortunately, in certain cases, this does not hold. In particular, for regions in which the surface changes very suddenly from slice to slice, it is likely that the mesh will be quite distant from parts of the boundary. The "horns" referred to above illustrate this point. In this case the edges will not be sensed, and it is likely that the mesh will continue to evolve away from these structures as it moves to better fit the available data. One possible solution to this problem is to adopt a two stage boundary detection process: a full 3D search of interpolated intensity data, would allow us to obtain a rough idea where the boundary was. We could then apply an in-plane search to obtain more accurate edge localisation.

The mis-identification of boundaries arose from two sources: degraded boundaries and unmodelled sulcal penetrations. Boundary degradation — Figure 7.20 — in the form of bound-

Figure 7.20: Merged Boundaries This image shows a region where the boundary of one tissue has merged with another (within the box). In such cases, the boundary model will match the wrong boundary.

ary merging was only observed in a few cases. However, this artifact[7] was sufficient to cause a mismatch, with the next available boundary being selected. The primary source of other boundary mismatches was sulcal penetrations. As Figure 7.19 shows, such a feature can cause the boundary model to incorrectly identify the boundary type. In this example, the search scale does not include the skull, so the chance of a mismatch is increased. Although one would expect that such a match would be rejected by the validation profiles, this will only occur if the sulcus does not run parallel to the brain boundary within the validation range. Although such sulci are less common, they do occur; in any event, the rejection threshold allows a certain amount of latitude, so even matches from non-parallel sulci can be selected.

An important issue which affects the final surface segmentation is the availability and correctness of MSP points. In general, there will be very few points associated with the MSP, since we have no model for this region in our database. Consequently, the surface segmentation must use the boundary point set to manoeuvre itself into position. Since the shape space only permits flat MSP estimates, one would always expect a reasonable plane approximation to emerge. However, if the initial affine segmentation stretches or shears the model so that the MSP estimate is slightly off, the lack of data in this region is likely to ensure that this error will be propagated through the entire segmentation process.

The initial constrained segmentation is less sensitive to the choice of governing parameters than the simplex segmentation, since the latter is significantly more flexible. The simplex parameters we used provided a good compromise between data interpolation and model extrapolation. Ideally, these parameters would be automatically selected to optimise surface recovery for a given data set.

A number of possible strategies to tackle these problems are outlined in the Future Work section of this thesis.

[7]It is not clear what causes this, but we shall refer to it as an imaging artifact.

7.7 Conclusion

In this chapter we have developed a scheme for constrained segmentation within sparse MR data. The segmentation algorithm is based on a two-tier approach, and uses boundary data obtained from the boundary matching scheme introduced earlier. The first stage of the algorithm applies learned shape constraints to produce the best segmentation permitted within the associated shape space. The second phase uses this result as a patient-specific simplex template, which loosely constrains the evolving surface and permits a more accurate final solution. The method is essentially automatic, requiring no additional user intervention after the initial specification of parameters. Examples where shown for the segmentation of the left and right cerebral hemispheres. These surfaces may be used in subsequent asymmetry analysis.

Chapter 8

Conclusion

This thesis investigated the segmentation of the cerebral surface from sparse MRI. Sparse images pose difficulties for many image segmentation methods, since they are poorly subsampled and heavily truncated. In order to overcome these problems, we have made extensive use of *a priori* information. This information was primarily encapsulated in the form of two models: a shape model and a model of the expected boundary structure.

The segmentation process consists of boundary detection followed by surface fitting. A database of 1D piecewise constant boundary models, corresponding to idealised boundary cross-sections, is specified in advance. For a given 1D intensity profile, each model is matched in turn, and, provided the model constraints are respected, the best match is accepted. The matching process returns the location of the associated 3D boundary point. The set of boundary points must then be interpolated to yield a surface — but in a robust and anatomically meaningful manner. This is achieved through the use of a shape template. The template is generated from a representative set of (apparently!) "normal" individuals and encodes the permissible shape variation of the sample normal population. A simplex mesh is used to represent the model surface, and at each iteration, the mesh vertices are adjusted to better fit the 3D boundary point set, subject to the learned shape constraints. Upon convergence, the mesh provides the "optimal" constrained segmentation for the boundary under consideration. To represent pathology (or deformation not permitted under the model), the shape constraint is relaxed. A local update scheme, based on the properties of the simplex mesh, is used. Under this constraint, the mesh smoothly evolves from the result of the initial surface estimate. The simplex shape constraints are weak, and seek only to preserve the existing surface geometry over regions which are far from the boundary data.

The segmentation scheme is specifically tailored to suit the requirements of the sparse framework. The boundary detection component only considers voxels within each image slice, thus eliminating the need to consider (spurious) interpolated intensity data. A method is also developed to predict the intensity of partially filled voxels, since these are shown to have a significant effect on boundary localisation. This is achieved by combining a simple model of MR image formation with the matched models resulting from the boundary detection phase. These predicted intensities are used to build a local intensity model which may be used to refine the boundary estimate. The surface update is also modified: the definition of the closest point, which is used in both stages of the segmentation, is adapted to account for the planar arrangement of the boundary data.

On the whole, the resulting surfaces were satisfactory. Currently, only qualitative "validation" has been performed, through the use of registered high resolution data sets. For those data sets which did not have a high incidence of boundary degradation, the final surfaces provided a good compromise between data interpolation and model extrapolation. When there were clear discrepancies between the surface approximation and the cerebral boundary, these can be traced to a small but recurrent set of problems which are addressed in the future work section below.

8.1 Future Work

While the results presented are encouraging, there is much that can be done to improve the accuracy of the recovered surfaces.

8.1.1 The Shape Template

The shape template was built from a training set of 20 normal (male and female) subjects. Given the variability exhibited by anatomical structures, it seems likely that a larger training set would provide a more representative constrained segmentation. Since the simplex shape constraint is built from this initial segmentation, one would also expect an improvement in the accuracy of the final extracted surface. A separate template should also be built for each sex, since there are gender-specific structural differences which it may be useful to eliminate from subsequent shape analysis.

8.1.2 The Boundary Model

The boundary detection process is sensitive to the choice of controlling parameters. In particular, the rejection threshold may be such that valid matches are ignored because the difference in the orientation of the boundary and the local surface patch is too large. In many cases, these points become feasible as the surface updates. However, in other cases, particularly when narrow structures are involved, the lack of data ensures that the surface will not provide a good representation of local structure. An adaptive rejection threshold, based on the local curvature of the template could be developed. Alternatively, a new "distance metric" could be introduced which included the grey-scale difference between neighbouring profiles as well as the separation of the detected boundary points.

The model matching process works well in most cases, but if the boundary is severely degraded or merges with a neighbouring tissue class, then an incorrect match may result. Since this degradation tends to be very localised, the constrained segmentation can cope adequately. Unfortunately, the simplex segmentation is likely to develop unwanted bumps or dimples in such regions. The boundary model constraints play an important role in determining which boundary points are matched: by insisting on tighter limits on model variability, this problem can be reduced or eliminated. In particular, absolute constraints can be introduced on the permissible length variation of each tissue interval. Such knowledge could be acquired by consultation with neuro-anatomists, for example. In this case, a spurious boundary match would result in a unnaturally dilated or contracted tissue interval which would cause the match to be rejected.

8.1.3 PVE Prediction

The PVE prediction scheme produced encouraging results, but was unable to much improve boundary localisation near the top of the brain. The primary cause for this failure is the existence of fluid-filled sulci, which produce sharp ridges or valleys in the intensity signal. These features are sufficiently wide to ensure that simple smoothing will not remove them. One possibility is to model these fluctuations as variable width tissue bands, although given their arbitrariness, this may prove difficult. Another approach would involve the preprocessing of the intensity profile, using some sort of structure preserving multi-scale scheme. At a coarse scale this would eliminate these signal fluctuations, thus enabling us to obtain an initial boundary estimate. Once the boundary locality had been established, a small window of the original signal intensities could be examined in order to refine the fit.

8.1.4 Segmenting Other Structures

The techniques developed for the cerebral surface were tested on T1 and T2 axial scans. Only the cerebral hemispheres were extracted. However, the methodology we developed is suitable for use with any structure for which we can construct a shape template and a boundary model database. One possible candidate for study is the ventricular system. In this case the topology is more complex, but the boundary model is fairly simple. The one component of the segmentation scheme which will need to be adapted is the initialisation step. However, since the initialisation may be very coarse, it should be easy enough to construct an appropriate initialisation scheme, using, for example, thresholding.

8.1.5 Asymmetry/Shape Analysis

An elementary form of shape comparison may be accomplished by computing the shape parameters for each mesh (through projection into the PDM shape space) and comparing them under some metric. For example, one may compare a set of segmented hemispheres by evaluating a simple sum of squared differences measure, with the shape vectors as inputs. However, such a global measure tells one little about subtle changes across the input set. Furthermore, there is no explicit information about lateral asymmetries.

To obtain such information, a *symmetry map* can be constructed [40]. The symmetry map represents LR brain laterality in the form of a 2D function which may be analysed using eigen-function techniques — Figure 8.1. This description requires an estimate of the MSP for the hemisphere pair. The MSP plane definition is implicit in the reference template: each cerebral template is aligned in such a way that the MSP is coincident with the coordinate plane[1], $x = 0$. If we compute the best affine ICP transformation which aligns the template with the shape instance, this transformation will map the coordinate plane to the shape instance MSP.

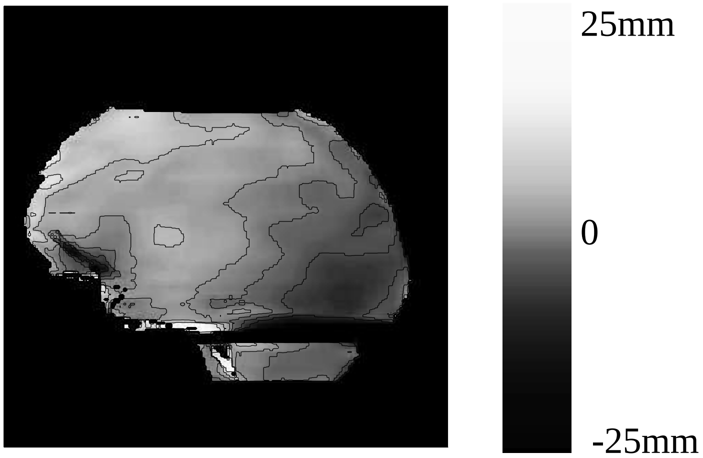


Figure 8.1: Symmetry Map Dark values correspond to negative asymmetry (the right hemisphere is smaller than the left) whilst lighter values represent positive asymmetry (the right hemisphere is larger than the left). The background has been set to match the darkest value (i.e. the highest negative asymmetry). The map shows that the left hemisphere is larger than the right just above the cerebellum; this asymmetry progressively diminishes as one moves away from this region. The values of brain asymmetry are plotted here for a range of -25mm up to 25mm to allow for border effects (where the hemispheres are laterally shifted, and so misaligned). The key gives the scale of asymmetry to grey-level.

The unreliability of the laterality measures presented by Bilder [6] (see Figure 1.4) and others, provided much of the impetus for this work. It is fitting, therefore, that we should conclude this thesis with a brief discussion of possible contributions in this area.

When discussing such measures, we suggested that the inaccuracy arose, at least in part,

[1] Or $x - b = 0$, where b is the width of the template bounding box in the x direction.

from the sparsity of the data and the use of voxel-based segmentations. Based on the results presented earlier, it seems clear that a significant improvement in accuracy could be achieved if the sparse segmentation framework was adopted. Dense laterality estimates (symmetry maps) may be obtained using the scheme suggested above. If the brains are registered to a canonical reference brain, as suggested in [40], one can generate a normalised set of laterality measures, and produce a graph similar to that shown in Figure 1.4. Because of the improved surface correspondence arising from registration and more accurate boundary localisation, one would expect a marked decrease in the width of the error bars. Such an improvement is necessary if the asymmetry hypothesis is to be investigated with any confidence.

These various tasks constitute a significant amount of additional work, but for the most part they require only small modifications to the existing infrastructure.

Appendix A

B-Spline snakes

A.1 Definition

A *B-spline* [4, 32] is a smooth curve which consists of polynomial segments, called *spans*, joined at *knot points*. Within each span the curve is a weighted sum of a set of *control points*, $\mathbf{X}_i$, where the weighting or *blending functions*, denoted $B_j(t)$, are simple polynomial functions of a curve parameter t. For example, for a cubic B-spline function (Figure A.1), there are four control points per span, and four blending functions:

$$\begin{aligned}
B_0(t) &= \frac{1}{6}t^3, \\
B_1(t) &= -\frac{1}{6}(3(t+1)^3 - 12(t+1)^2 + 12(t+1) - 4), \\
B_2(t) &= \frac{1}{6}(3(t+2)^3 - 24(t+2)^2 + 60(t+2) - 44), \\
B_3(t) &= -\frac{1}{6}((t+3)^3 - 12(t+3)^2 + 48(t+3) - 64),
\end{aligned}$$

where $t \in [0, 1]$. Within the span the curve is given by

$$\mathbf{x}(t) = \sum_{j=0}^{3} \mathbf{X}_j B_j(t). \tag{A.1}$$

In general, the B-spline does not pass through the control points, which is why it is called an approximating rather than an interpolating spline. Observe, however, that if a control point, say $\mathbf{X}_j$, is moved, say to become $\mathbf{X}_j + \mathbf{T}$, then equation A.1 shows that the spline will be pulled in the same direction $\mathbf{T}$ by an amount that is determined by the set of weights $B_j(t)$.

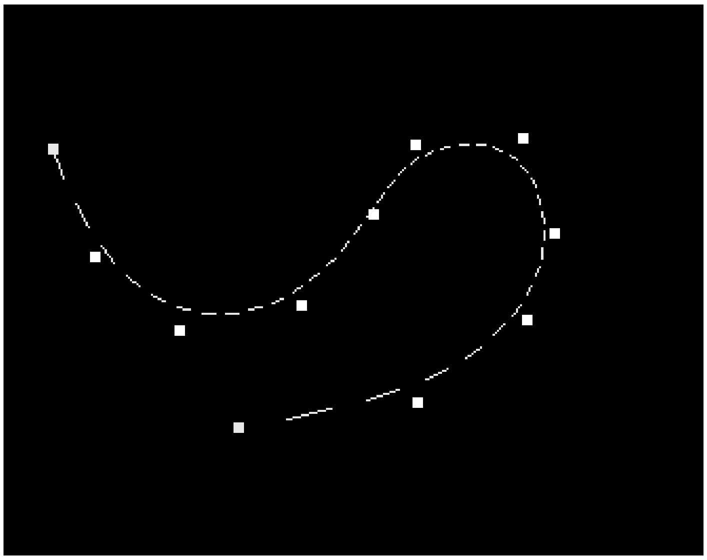


Figure A.1: Cubic B-spline The control points define the spline. By moving them, one can adjust the position of the curve.

Usually there are more than four control points, in which case the control points are ordered and the spans are determined by consecutive sets of $d+1$ control points, where d is the degree of the polynomial. For a closed curve the number of spans is equal to the number of control points.

Generally, if there is a set of spans, the curve on each span is given by

$$\mathbf{x}_i(t) = \sum_{j=0}^{3} \mathbf{X}_j B_{i,j}(t), \tag{A.2}$$

where the range of the parameter t is broken up into unit length intervals, one for each span. The index i in the expression above indicates the ith parameter span (which will have a value determined at each point by the weighted sum of the $B_{i,j}(t)$). Note that in general the blending functions may vary from span to span, in particular they tend to be different in spans near the end of the curve, in order to ensure that the end-points are interpolated correctly. This is why Equation A.2 uses the notation $B_{i,j}(t)$. In fact, this subtlety is not necessary in the work presented here: if the initial and final control points of the curve are the same, i.e. $\mathbf{X}(0) = \mathbf{X}(N)$, then it can be shown that the set of weight functions $B_{i,j}(t)$ is the same for each span.

The degree of the spline used determines the order of continuity along the curve. In the case of a cubic spline derivatives up to second order (curvature) are continuous. A quadratic spline requires only three control points per span, but is continuous only up to first derivative (tangency), and the curvature is piecewise constant along the curve.

The spline equations are generally rewritten to make them amenable to standard matrix manipulations. For general degree d, we have:

$$x_i(t) = \begin{pmatrix} 1, t, \cdots, t^d \end{pmatrix} \mathbf{B}_i \mathbf{S}_i \begin{pmatrix} X_0 \\ \vdots \\ X_l \end{pmatrix} \tag{A.3}$$

where $0 \leq t \leq 1$. The expression for $y_i(t)$ is similar, but with Y_i substituted for X_i.

The matrix $\mathbf{B}_i$ provides the B-spline coefficients for the ith span (the coefficients of the blending functions) while $\mathbf{S}_i$ selects the appropriate control points for the same span — it's just a matrix with ones for each control point to be selected and zeros elsewhere. Span j requires control points indexed by $j-d, \cdots, j-1, j$. If the spline is closed, this is simply the index modulo the number of control points.

In the following the snakes used are closed, and the matrix $\mathbf{B}_i$ is the same for all spans. For example, the cubic B-spline matrix (cf. Equation A.1) is

$$\mathbf{B} = \frac{1}{6} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.4}$$

The number of control points determines the range of the spline parameter. For M control points, the parameter s has range $[0, M]$, if the spline is closed; $[0, M-d]$, if the spline is open.

A.2 Area measurements

For a closed parameterized plane curve, Green's Theorem can be used to to compute the enclosed area as a one-dimensional contour integral. For a spline (which has, say, M control points, and consequently, M spans), the expression reduces to

$$\text{Area} = \sum_{n=0}^{M-1} \int_0^1 x_n(s) \dot{y}_n(s)\, ds \tag{A.5}$$

where $\dot{y}_n(s)$ denotes the derivative of y_n with respect to s. Given the spline definitions above, this may be rewritten as

$$\text{Area} = \sum_{n=0}^{M-1} \sum_{i=0}^{d} \sum_{j=0}^{d} X_{(n-i)\%M} Y_{(n-j)\%M} H_{ij}, \tag{A.6}$$

in which: $(n-i)\%M$ means modulo M, since the curve is closed and the $(M+1)$st point identified with the first; the matrix is given by $\mathbf{H}$

$$H_{ij} = \int_0^1 B_i(t)\dot{B}_j(t)dt. \tag{A.7}$$

where $B_l(t)$ are the B-spline blending functions we referred to earlier. For example, in the cubic case, we find

$$\mathbf{H} = \begin{pmatrix} \frac{1}{72} & \frac{1}{80} & -\frac{1}{40} & -\frac{1}{720} \\ \frac{71}{720} & \frac{5}{24} & -\frac{61}{240} & -\frac{19}{360} \\ \frac{19}{360} & \frac{61}{240} & -\frac{5}{24} & -\frac{71}{720} \\ \frac{1}{720} & \frac{1}{40} & -\frac{1}{80} & -\frac{1}{72} \end{pmatrix} \tag{A.8}$$

Because the spline is closed, the $B_l(t)$ are independent of the span, and the matrix is the same for all spans. The summation over i and j only considers the relevant control points points for each span, i.e., the explicit $\mathbf{S}_i$ matrix is not necessary.

Appendix B

The Simplex Mesh Formalism

A Simplex Mesh is a set of connected vertices which possess fixed vertex connectivity. More formally, a *k-Simplex Mesh*, $\mathcal{M}$ of $\mathbb{R}^d$, is defined by the pair $(V(\mathcal{M}), N(\mathcal{M}))$ where $V(\mathcal{M})$ is the vertex set and $N(\mathcal{M})$ provides the vertex connectivity mapping. For this work, $d = 3$ and $k = 2$ (surface in 3D). Since our work will concern 2-simplex (surface) meshes only, we shall focus on them and drop the explicit k prefix unless we wish to state a more general result.

The vertex connectivity mapping ensures that

1. each vertex is connected to $k + 1$ other vertices

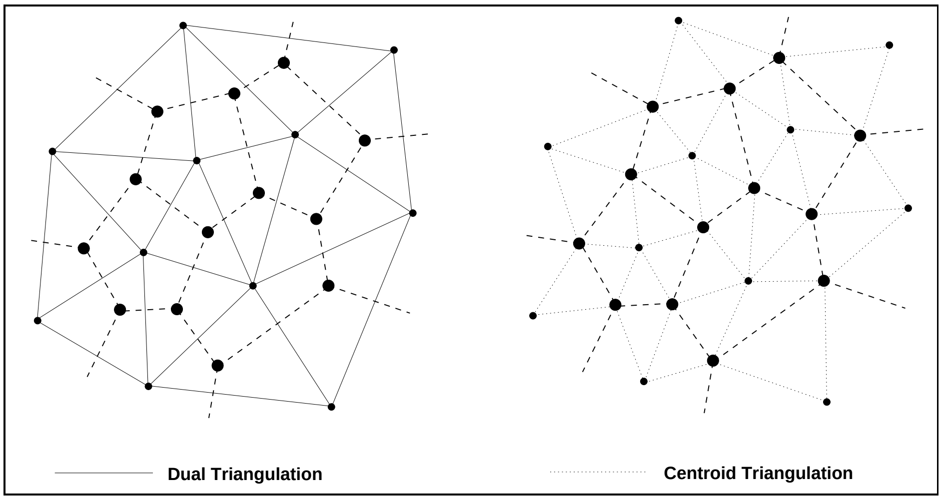
2. there are no cycles or loops in the edge representation.

A mesh with these properties is so named because the set of points directly connected to any given vertex comprises a k-simplex, according to the following definition:

Definition B.1 (k-simplex) *A k-simplex of* $\mathbb{R}^d$, $\quad k < d$, *is a set of* $k + 1$ *independent points in* $\mathbb{R}^d$.

For example, in a 2-simplex (surface) mesh, each vertex is directly linked to three neighbours, which comprise the local simplex in $\mathbb{R}^3$. Since the mesh is not degenerate, the 3 neighbouring points are independent, as required by the definition.

There is a close relationship between the simplex mesh and a triangulation of the vertex set, as suggested by the usage of the term "simplex". They are in fact *dual* representations i.e. we can derive one from the other — Figure B.1a). This duality is topological — the simplex mesh and its associated dual triangulation *need not be geometrically equivalent*, in the sense that the "dual of the dual" will not, in general, result in the original simplex mesh or triangulation. This may be formally proved by examining the Euler equations relating the number of vertices, edges and faces in each representation. One can, however, establish geometric equivalence between a simplex mesh with planar polygons and its dual triangulation. Although there



A B

Figure B.1: Simplex Mesh Triangulation One may move from a simplex mesh to a triangulation without altering the topology of the represented surface. However, geometry may only be preserved if the simplex mesh is composed of planar polygons. Given a simplex mesh, we may derive a triangulation in one of 2 ways: A) using the centroid of each polygon to define the vertices of the *dual triangulation* or B) using the *centroid triangulation* to decompose each (possibly non-planar) polygon into a sequence of triangles.

is no unique triangulation for general simplex meshes, such a description is required for practical tasks such a surface rendering and volume calculations. Two methods are provided for determining a triangulation from a given simplex mesh:

Dual Triangulation The duality relationship associates a triangle with each simplex mesh vertex and provides a way of connecting them — Figure B.1A. Although this method has the advantage of generating an optimal triangulation (in the sense that the minimum number of triangles are used), it results in a consistent under-estimation of mesh curvature [30].

Centroid Triangulation An alternative triangulation involves the decomposition of each polygon into a set of triangles — Figure B.1B. This is accomplished by joining each pair of vertices which define the polygon to its centroid. This approach is used when we desire a more accurate triangulation and was the representation we utilised. Unfortunately, the number of triangles involved may be substantially higher, since triangles are not shared across neighbouring polygons.

The simplex mesh representation is extremely versatile. One can represent a range of topologies within a unified framework. Two classes of mesh *transformation operations* may be defined: Eulerian and Global. The former may be used to refine the mesh (through addition or decimation of points), while the latter may be used to effect topological changes, such as the addition of holes.

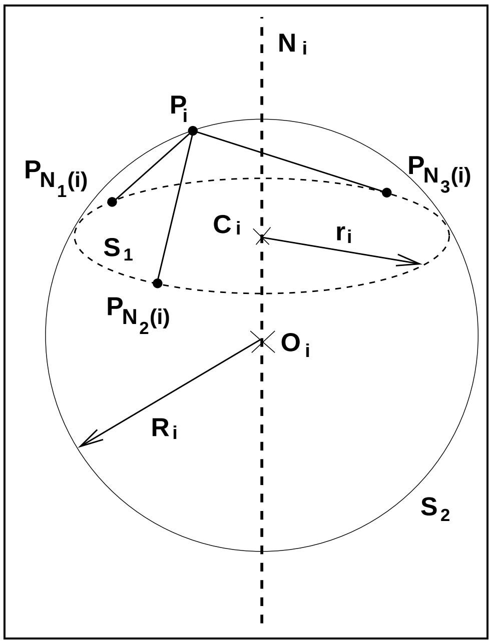


Figure B.2: Simplex Angle The simplex angle at a mesh vertex, P_i, provides a measure of local deviation from a plane viz. the one defined by the three neighbouring mesh vertices — $P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)}$. The circle, S_1, passing through these three points has radius r_i and centre C_i, while the sphere, S_2, which passes through all four points has radius R_i and is centred at O_i. The simplex angle is independent of the position of the neighbouring points on the S_1 and is zero when P_i is on the plane defined by these points.

B.1 Geometric Properties

At each vertex, P_i on an (oriented) simplex mesh, one may define a normal vector using the plane passing through the 3 neighbouring points, $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$:

$$N_i = \frac{P_{N_1(i)} \times P_{N_2(i)} + P_{N_2(i)} \times P_{N_3(i)} + P_{N_3(i)} \times P_{N_1(i)}}{||P_{N_1(i)} \times P_{N_2(i)} + P_{N_2(i)} \times P_{N_3(i)} + P_{N_3(i)} \times P_{N_1(i)}||} \tag{B.1}$$

A given vertex together with its 3 neighbours defines a tetrahedral mesh element. Given these points, one may compute a quantity known as the *simplex angle*, which provides a means of quantifying the local discrete curvature about the vertex.

Definition B.2 (Simplex Angle) *Let S_1 be the circumscribed circle at the three neighbouring mesh vertices $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$. This circle has radius r_i and centre C_i. Let S_2 be the circumscribed sphere at the four vertices $(P_i, P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$, with associated origin O_i and radius R_i (See Figure B.2). The simplex angle, ϕ_i, at vertex P_i is defined by*

$$\begin{aligned} \sin(\phi_i) &= \frac{r_i}{R_i} * sgn(\overrightarrow{P_i P}_{N_1(i)} \cdot N_i) \\ \cos(\phi_i) &= \frac{||O_i C_i||}{R_i} * sgn(\overrightarrow{O_i C_i} \cdot N_i) \end{aligned} \tag{B.2}$$

Here $||O_i C_i||$ represents the distance from the centre of the sphere to the plane and $\phi_i \in [-\pi, \pi]$. sgn(x) returns ± 1 depending on the sign of its argument.

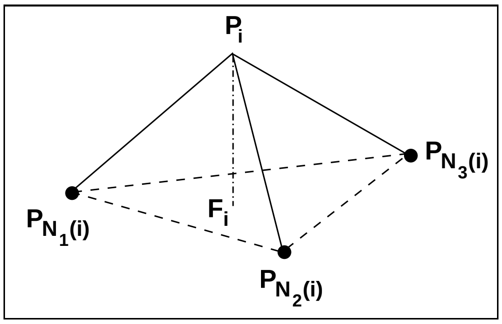


Figure B.3: Metric Parameters The orthogonal projection, F_i, of the vertex P_i onto the plane passing through its 3 neighbours may be described through 3 metric parameters (for a 2-simplex mesh). Only 2 need be specified since they must collectively sum to unity. These parameters provide "barycentric coordinates" for the projected point and offer a means of enforcing some local regularity on the mesh: we require that these parameters always be positive. This precludes the existence of skewed mesh elements and is a form of local regularisation.

The simplex angle is independent of the positions of the points $(P_{N_1(i)}, P_{N_2(i)}, P_{N_3(i)})$ on the circle S_1 and of the position of P_i on the circumscribed hemisphere of S_2. Furthermore, it is zero only when P_i lies on the plane defined by its 3 neighbours.

B.2 Metric Parameters

In order to characterise the local geometry on a simplex mesh, a set of *metric parameters*, $\epsilon_i = \{\epsilon_{1i}, \epsilon_{2i}, \epsilon_{3i}\}$, are introduced at each vertex. These parameters describe how each vertex is placed with respect to its 3 neighbours. The orthogonal projection of P_i onto the plane spanned by its 3 neighbours produces, F_i — Figure B.3 — which may be computed as

$$F_i = \epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + \epsilon_{3i} P_{N_3(i)}, \tag{B.3}$$

where $\epsilon_{1i} + \epsilon_{2i} + \epsilon_{3i} = 1$.

The point P_i may then be represented as

$$P_i = F_i + L(r_i, d_i, \phi_i) N_i, \tag{B.4}$$

where r_i and N_i have their previous meanings and d_i is the distance between F_i and the centre of the circumscribed circle, C_i. The function $L(.)$:

$$\begin{aligned} L(r_i, d_i, \phi_i) &= \frac{(r_i^2 - d_i^2)\tan\phi_i}{\epsilon\sqrt{r_i^2 + (r_i^2 - d_i^2)\tan^2\phi_i} + r_i} \\ \epsilon &= 1 \text{ if } |\phi_i| < \frac{\pi}{2} \\ \epsilon &= -1 \text{ if } |\phi_i| > \frac{\pi}{2}, \end{aligned} \tag{B.5}$$

provides the height of the point above the plane and is primarily a means of expressing the influence of the simplex angle.

The metric parameters provide a "barycentric" description of the projected point. These parameters are constrained to be positive in order to increase mesh stability. In practice this does not limit the usefulness of this description [30]. The metric parameters together with the simplex angle provide a unique description of a shape, up to a similarity. Since we have no intention of representing the mesh using these parameters, this observation is of no importance to us.

The computation of metric parameters is simple:

We have

$$\epsilon_{1i} P_{N_1(i)} + \epsilon_{2i} P_{N_2(i)} + (1 - \epsilon_{1i} - \epsilon_{2i}) P_{N_3(i)} = F_i, \tag{B.6}$$

which leads directly to

$$\epsilon_{1i} \frac{(P_{N_2(i)} - P_{N_3(i)}) \cdot (P_{N_1(i)} - P_{N_3(i)})}{\|P_{N_2(i)} - P_{N_3(i)}\|^2} + \epsilon_{2i} = \frac{(F_i - P_{N_3(i)}) \cdot (P_{N_2(i)} - P_{N_3(i)})}{\|(P_{N_2(i)} - P_{N_3(i)}\|^2}, \tag{B.7}$$

and

$$\epsilon_{1i} + \epsilon_{2i} \frac{(P_{N_1(i)} - P_{N_3(i)}) \cdot (P_{N_2(i)} - P_{N_3(i)})}{\|P_{N_1(i)} - P_{N_3(i)}\|^2} = \frac{(F_i - P_{N_3(i)}) \cdot (P_{N_1(i)} - P_{N_3(i)})}{\|(P_{N_1(i)} - P_{N_3(i)}\|^2}. \tag{B.8}$$

These equations are easily inverted. We may obtain the orthogonal projection of F_i as

$$F_i = P_i - (P_i - P_{N_1(i)}) N_i, \tag{B.9}$$

where the sign of the second term assumes that the normal is direct towards P_i.

B.3 The Simplex Mesh and Segmentation

The simplex mesh formalism outlined in [30] has a number of strategies for controlling smoothness and shape. The mesh update is

based on a Newtonian force model. Specifically, the movement of each node is governed by the following differential equation

$$m \frac{d^2 P_i}{dt^2} = -\gamma \frac{dP_i}{dt} + F_{\text{int}} + F_{\text{ext}} \tag{B.10}$$

where m is the mass of the node (unity will be assumed) and γ is a damping coefficient. Each node is subjected to an internal force, F_{int}, which express intrinsic bending constraints, and an external force, F_{ext}, which represents the influence of attractors in the environment. The equations of motion are discretised in time as

$$P_i^{t+1} = (1 - \gamma)(P_i^t - P_i^{t-1}) + P_i^t + F_{\text{int}}(t) + F_{\text{ext}}(t). \tag{B.11}$$

B.3.1 Internal Forces

The internal forces are expressed in terms of so-called "shape functionals" which characterise the shape of the local tetrahedral element surrounding each vertex. Each element has a rest shape, which it naturally adopts when no external stresses are applied. In this state, it possesses a simplex angle, ϕ_i^*. If we assume that the vertex P_i occupies the position P_i^* when in this state, then the energy, $\mathcal{S}_i$, of the local tetrahedron may be as expressed as

$$\mathcal{S}_i = \frac{\alpha_i}{2}|P_i^* - P_i|^2, \tag{B.12}$$

where $0 \leq \alpha_i \leq 0.5$ may be used to increase the strength of the internal force in the update equation. This value must be less than 0.5 to ensure stability.

Under this scheme, the gradient of this energy defines F_{int}:

$$F_{\text{int}} = \frac{\partial \mathcal{S}_i}{\partial P_i} = \alpha_i(P_i^* - P_i). \tag{B.13}$$

This definition is consistent with an "elastic" model of internal force. Combining Equations B.3,B.4 and B.13, we arrive at the final form of F_{int}:

$$F_{\text{int}} = \alpha_i(\epsilon_{1i}(P_{N_1(i)} - P_i) + \epsilon_{2i}(P_{N_2(i)} - P_i) + \epsilon_{3i}(P_{N_3(i)} - P_i) + L(r_i, d_i, \phi_i^*)N_i), \tag{B.14}$$

where ϕ_i^* determines the particular type of mesh constraint to be enforced. Four are identified:

1. **Null Constraint** $\phi_i^* = \phi_i$. This is not really a constraint — the surface can bend freely around P_i.

2. **Smoothing** $\phi_i^* = 0$. Under this constraint the mesh vertices will tend to smooth the surface regardless of local curvature of geometry.

3. **Mean curvature continuity**

$$\phi_i^* = \arcsin\left(\sum_{j \in N^r(P_i)} \frac{e_{ij} * r_i * \sin\phi_j}{r_j}\right)$$
$$\sum_{j \in N^r(P_i)} e_{ij} = 1, \quad 0 < e_{ij} < 1, \tag{B.15}$$

   where $N^r(Pi)$ represents a neighbourhood of size r about P_i. This constraint smoothes curvature and is controlled through the parameter r.

4. **Shape** If we set $\phi_i^* = \phi_i^0$ we can provide a "template" which represents the shape to which the mesh will revert when all external forces are removed.

In the absence of external forces, the smoothing constraint will cause the mesh to shrink in on itself, unless the simplex mesh is a planar object, in which case $\phi_i^* = 0$ is already satisfied.

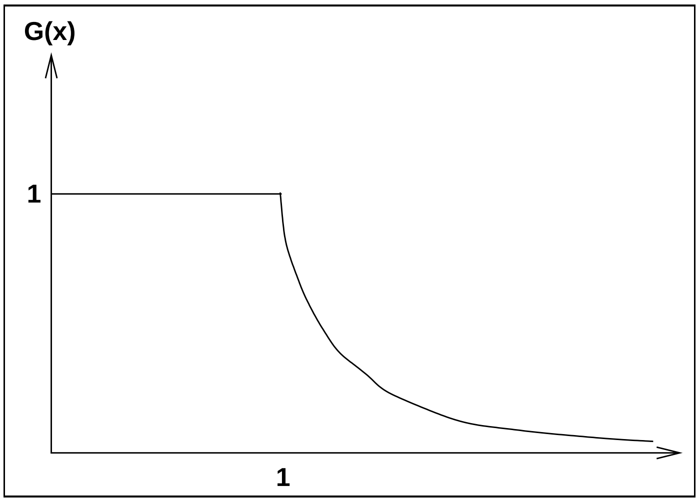


Figure B.4: Stiffness Function This function provides a means of reducing the effect of distant data during segmentation. As long as the argument is less than 1, the data force remains unchanged. After this point, the force decays under an inverse square law.

B.3.2 External Forces

If we are to use the simplex mesh for segmentation, we must have a means of generating external forces which attract our model to the structures of interest. In this framework, a search algorithm is used to identify the closest feature point, $M_{Cl(i)}$, to each mesh vertex P_i. The external force is then computed as

$$F_{\text{ext}} = \beta_i G\left(\frac{\|M_{Cl(i)} - P_i\|}{D}\right)((M_{Cl(i)} - P_i) \cdot N_i)N_i, \tag{B.16}$$

where N_i is the normal vector at P_i and $G(x)$ is a "stiffness function" which provides a means of limiting the influence of distant data — Figure B.4. The function used exhibits inverse squared behaviour outside the unit interval. Within the interval the function does not modify the force. The parameter D determines the distance at which this transition occurs. It is initially computed as 20% of the radius of the sphere which encompasses the data. As the segmentation proceeds, this value is gradually reduced, enabling more subtle deformations to emerge.

The detection of the closest data point depends on the kind of information we have at our disposal. In those instances where iso-surface or point data is available, efficient algorithms may be devised to select the required points. For volumetric medical data, gradient maxima searches along 1D lines normal to the mesh surface are employed [30, 52]. This approach is analogous to the 2D snake search procedure employed in — Chapter 3. In the simplex mesh framework, the only provision for the utilisation of *a priori* knowledge consists of region-based continuity constraints (as explained in the next section).

The nominal vertex displacement is projected onto the vertex normal and processed via Equation B.16 to yield the external force. The normal projection ensures a smoother surface update, particularly when sparse point data is being processed. By modifying the parameter β_i the relative importance of the external force in the update equations can be controlled.

Appendix C

Mesh Measures

The volume and surface area of a (closed) non-intersecting triangulated surface are easily calculated.

C.1 Surface Area

The surface area is simply the sum of the individual surface areas contributed by each of the N_T triangles. More specifically,

$$S = \sum_{i=1}^{N_T} \frac{\|(\mathbf{x}_{i,1} - \mathbf{x}_{i,0}) \times (\mathbf{x}_{i,2} - \mathbf{x}_{i,0})\|}{2}$$

where $\mathbf{x}_{i,j}$, $j = 0, 1, 2$ is the j vertex on the ith triangle.

C.2 Volume

Gauss's Divergence Theorem shows that we may compute the volume enclosed by any piece-wise smooth surface by evaluating the following surface integral:

$$V = \frac{1}{3} \int\int_S \mathbf{x} \cdot \mathbf{n} \, dS.$$

where $\mathbf{n}$ is the surface normal at $\mathbf{x}$.

In the case of a non-intersecting triangular mesh, we have a surface, $\{\cup_{i=1}^{N_T} T_i\}$. Each T_i is a triangular facet, with a constant normal, $\mathbf{n}_i$. The plane spanned by this triangle is given by

$$(\mathbf{x} - \mathbf{x_{i,0}}) \cdot \mathbf{n_i} = \mathbf{0}.$$

We are now in a position to evaluate the surface integral:

$$V = \frac{1}{3}\int\int_{\cup_{i=1}^{N_T} T_i} \mathbf{x}\cdot\mathbf{n}\,dS. \tag{C.1}$$
$$= \frac{1}{3}\sum_{i=1}^{N_T}\int\int_{T_i} \mathbf{x}_{i,0}\cdot\mathbf{n}_i\,dS$$

since surface is composed of planar facets on which the dot product is constant.

$$V = \frac{1}{3}\sum_{i=1}^{N_T}(\mathbf{x}_{i,0}\cdot\mathbf{n}_i)\int\int_{T_i} dS \tag{C.2}$$
$$= \frac{1}{3}\sum_{i=1}^{N_T}\mathbf{x}_{i,0}\cdot\frac{(\mathbf{x}_{i,1}-\mathbf{x}_{i,0})\times(\mathbf{x}_{i,2}-\mathbf{x}_{i,0})}{\|(\mathbf{x}_{i,1}-\mathbf{x}_{i,0})\times(\mathbf{x}_{i,2}-\mathbf{x}_{i,0})\|}\frac{1}{2}\|(\mathbf{x}_{i,1}-\mathbf{x}_{i,0})\times(\mathbf{x}_{i,2}-\mathbf{x}_{i,0})\|$$
$$= \frac{1}{6}\sum_{i=1}^{N_T}(\mathbf{x}_{i,0}\times\mathbf{x}_{i,1})\cdot\mathbf{x}_{i,2}.$$

after using the cross-product formula for the area of a triangle and simplifying.

Thus we see that the volume enclosed by a non-intersecting triangular mesh of arbitrary topology may be computed from a simple sum of vector triple products.

C.3 Transforming the PCA Eigensystem

Given a set $\{\mathbf{x}_i\}_{i=1}^{M}$ of N dimensional points, computing the associated PCA reduces to solving the eigensystem

$$\mathbf{\Sigma u} = \lambda\mathbf{u} \tag{C.3}$$

where

$$\mathbf{\Sigma} = \sum_n(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^T. \tag{C.4}$$

If N is large, then the computational complexity of this problem becomes excessive, since the solution process is $O(N^3)$. To transform the system, we define $P = [(\mathbf{x}_1 - \overline{\mathbf{x}})\dots(\mathbf{x}_M - \overline{\mathbf{x}})]$ which yields

$$\mathbf{\Sigma} = PP^T. \tag{C.5}$$

We may now pre-multiply by P^T to obtain

$$P^T P(P^T\mathbf{u}) = \lambda(P^T\mathbf{u}) \tag{C.6}$$

which is equivalent to solving the eigensystem $\Sigma^* v = \lambda v$ where $\Sigma^* = P^T P$ and $v = P^T u$.

The set of eigenvectors obtained from this system may be "interpolated" to yield the eigenvectors of Equation C.3 by pre-multiplying each with P. The eigenvalues will be the same in both cases.

The rank of the covariance matrix determines the number of principal components (eigenvectors) which can be determined. If the number of training examples is not equal to the N, then the covariance matrix will be rank deficient for one of the two eigensystems. We always choose the eigensystem which has full rank (the smaller of the two) and interpolate the solution as shown above.

Appendix D

Mahalanobis PCA criterion

We are given an N-dimensional PCA, defined by a mean, $\boldsymbol{\mu}$, and a set of N modes of variation, $\{\mathbf{e_i}\}_{i=1,\ldots,N}$. The Mahalanobis distance (from the mean) is defined as

$$M(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \tag{D.1}$$

where S^{-1} is the inverse of the covariance matrix used to generate the eigensystem defining the PCA:

$$\mathbf{S}\mathbf{e_i} = \lambda_i \mathbf{e_i}, \;\; \mathbf{e_i}^T \mathbf{e_i} = 1, \;\; i = 1, \ldots, N, \tag{D.2}$$

where the eigenvectors are ordered by eigenvalue, in descending order. An arbitrary vector, $\mathbf{x}$, may be expanded in the PCA system as

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{E}\mathbf{b}, \tag{D.3}$$

where $\mathbf{E} = [\mathbf{e_1} \cdots \mathbf{e_N}]$ and $\mathbf{b}$ is a vector of shape parameters (basis coefficients).

Using these equations we have:

$$\begin{aligned} M(\mathbf{x}) &= (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= (\mathbf{E}\mathbf{b})^T \mathbf{S}^{-1} \mathbf{E}\mathbf{b} \\ &= \mathbf{b}^T ([\frac{\delta_{ij}}{\lambda_j}]_{ij}) \mathbf{b} \\ &= \sum_{j=1}^{N} \frac{{b_j}^2}{\lambda_j}. \end{aligned} \tag{D.4}$$

In general, a reduced set of eigenmodes will be retained, and the expansion is truncated at term l, $\; l < N$. In this case, one cannot compute the basis coefficients, b_i, for $i > l$. In [24], the eigenvalues associated with these modes are set to $\lambda_i = \frac{1}{2}\lambda_l, \; i > l$. This leads to

$$M \approx \sum_{i=1}^{l} \frac{{b_i}^2}{\lambda_i} + \frac{2}{\lambda_l} \sum_{i=l+1}^{N} {b_i}^2. \tag{D.5}$$

The sum of squared differences, R^2, between the original vector, $\mathbf{x}$, and its approximation, $\hat{\mathbf{x}}$, yields:

$$\begin{aligned} R^2 &= (\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}}) \\ &= (\mathbf{E}(\mathbf{b} - \hat{\mathbf{b}}))^T\mathbf{E}(\mathbf{b} - \hat{\mathbf{b}}) \\ &= (\mathbf{b} - \hat{\mathbf{b}})^T(\mathbf{b} - \hat{\mathbf{b}}) \\ &= \sum_{i=1}^{N}(b_i - \hat{b}_i)^2 \\ &= \sum_{i=l+1}^{N}(b_i - \hat{b}_i)^2 \end{aligned} \tag{D.6}$$

Finally, we arrive at the following:

$$M \approx \sum_{i=1}^{l} \frac{{b_l}^2}{\lambda_l} + \frac{2R^2}{\lambda_l} \tag{D.7}$$

Appendix E

Cosine Foreshortening for Curved Surfaces

Given a sequence of parallel planes, one can show that the distance between the planes scales linearly as the angle at which the measurement is made varies. More specifically, if we know the closest (perpendicular) distance between two planes, then the distance measure along a ray at an angle of θ radians will be scaled by $\frac{1}{\cos\theta}$ — Figure E.1. This is *cosine foreshortening.*

We may think of this transformation as a *warping* of one length into another, effected through some interval contraction/dilation function, $w(l)$, where l is the distance along the sampling line.

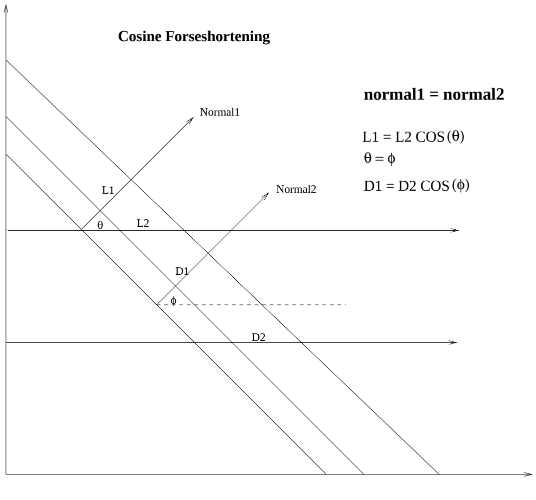


Figure E.1: Cosine foreshortening For a sequence of parallel planes, the distances between any pair of planes will be linearly contracted/dilated relative to the closest (normal) distance, depending on the angle at which they are measured.

For the simple case shown in Figure E.1, the warping function would be given by

$$w(l) = \frac{l}{\cos\theta}, \quad l \in [0, L], \tag{E.1}$$

where L is the length of the sampling line in the boundary normal direction. This rule holds for an arbitrary number of parallel planes: all lengths will be contracted or dilated by a factor of $\cos\theta$. In this scenario, a single normal is shared by each surface.

For an arbitrary set of curved parallel surfaces, this simple behaviour only holds in a differential sense. For a finite sampling ray, the foreshortening depends on the distance along the ray. This follows directly from the fact that the surface normal may vary at every point on the surface.

It is possible to derive a closed form expression for the foreshortening along such a sampling ray, provided one is is willing to accept a compromise. By approximating the local curved parallel surfaces by a series of concentric spheres, a simple expression may be computed, which provides some useful insights into nature of this warping.

The construction we employ is shown in Figure E.2. The sphere is assumed to be intersected by a number of planes, representing the MR image sampling planes. The in-plane sampling lines are oriented along normals to the plane-sphere intersection curve (which is a circle). Because of the symmetry of the construction, we need only consider a single plane which passes through both poles and the in-plane sampling ray. This reduces the problem from 3D to simple 2D plane geometry. As Figure E.2 shows, the line joining each end point of the in-plane sampling ray to the sphere surface, generates two different normal vectors. Since we are only considering the situation in the plane, the normal orientation may be described by an angle θ, measured w.r.t. the sampling plane (represented as a horizontal line in the 2D figure). Because the boundary is a circle, the normals may be linearly interpolated to give a normal at any point between the 2 end points. This is equivalent to defining

$$\theta(t) = \theta_i + t[\theta_f - \theta_i], \quad t \in [0,1], \tag{E.2}$$

where θ_f and θ_i, represent the plane angles of the normals at the final and initial points, respectively, of the sampling line.

One can see from the figure that as the extent of the in-plane boundary profile increases, the assumption of constant normal orientation breaks down. On a small enough scale, we can still use the constant angle assumption to generate a differential description of the local foreshortening[1]:

$$dl' = dl \ \cos\theta(l) \tag{E.3}$$

[1]We're using the "inverse" foreshortening, since for most of our work we wish to transform non-normal sampling into normal sampling.

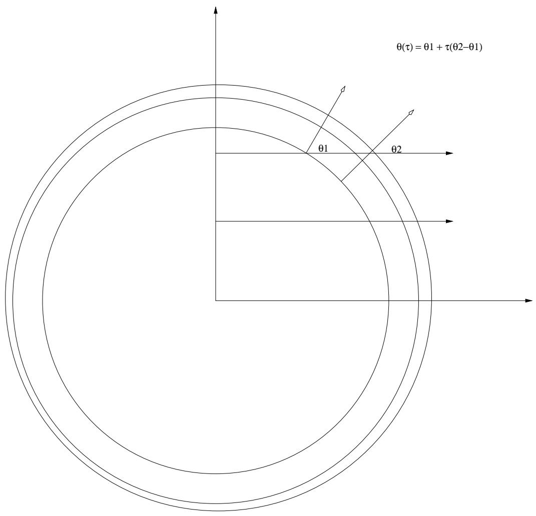


Figure E.2: Cosine foreshortening on the sphere By approximating a sequence of parallel surfaces by a local spherical patch, it is possible to derive a closed form solution for the foreshortening along a sampling ray. In our framework, the object is intersected by a number of parallel planes. The lines along which we sample in-plane data are boundary normals to the plane-sphere intersection curve within each slice. By considering the plane which passes through this line and the poles of the sphere, we may reduce the problem to one which requires only 2D planar geometry. Within this plane, the rays passing through each end point from the sphere (circle) centre provide us with two, different surface normals. We are interested in computing the foreshortening along the in-plane sampling ray, as a function of these delimiting angles.

To get the warping function, $w(l)$, we can integrate over the range of θ covered by surface normals which radiate from the endpoints of our profile sampling line (cf Figure E.2):

$$\begin{aligned} w(t) &= \int_0^t dl' \\ &= \int_0^t \cos\theta(l)dl \\ &= \int_0^t \cos(\theta_i + l\ [\theta_f - \theta_i])dl \\ &= \int_0^t [\cos\theta_i \cos l(\theta_f - \theta_i) - \sin\theta_i \sin l(\theta_f - \theta_i)]dl \\ &= \left[\frac{\cos\theta_i \sin l(\theta_f - \theta_i)}{\theta_f - \theta_i} + \frac{\sin\theta_i \cos l(\theta_f - \theta_i)}{\theta_f - \theta_i}\right]_0^t \\ &= \frac{\cos\theta_i \sin t(\theta_f - \theta_i)}{\theta_f - \theta_i} + \frac{\sin\theta_i \cos t(\theta_f - \theta_i)}{\theta_f - \theta_i} - \frac{\sin\theta_i}{\theta_f - \theta_i}. \end{aligned} \tag{E.4}$$

The warping function is defined over the interval $[0, 1]$ and specifies the mapping from one sampling direction to another over a normalised interval. In the definition of the integral above, we used the normal line attached to θ_i as our standard frame. In fact, because of the symmetry in the model, the actual warping is independent of the normal profile chosen (within the region delimited by our angles).

There are clearly regions where the model will break down: those in which the sampling plane does not intersect the full complement of nested surfaces. This will occur at the brain extremities. If one looks at the limiting behaviour of the warping function, we observe the following:

- $\lim_{\theta_f \to \theta_i} w(l) = l \ \cos\theta_i$
- $\lim_{\theta_f \to \theta_i, \theta_i = 0} w(l) = l.$
- $w(l)|_{\theta_i = \frac{\pi}{2}} = \frac{\cos l(\theta_f - \frac{\pi}{2})}{\theta_f - \frac{\pi}{2}}.$

The first limit tells us that as that as the normals converge, then we can revert to the standard cosine foreshortening rule — as we would expect. In addition, as both angles tend to zero, we recover the identity i.e. the surface normal and sampling direction coincide. When the sampling direction and the surface normal are orthogonal — which will happen at the poles of the sphere — then the third limit comes into play.

E.1 An Example

How does warping affect a profile? Consider the example presented in Figure E.3A). The original in-plane profile was resampled uniformly over it's length, which was normalised to unity, before undergoing the warping. The delimiting normals made angles of 1.4 and 1.0 radians, respectively, with the sample plane. The profile was warped onto the surface normal passing through the first end point (although it could equally well have been applied to the second point).

For comparison, a direct cosine warp is shown, with the foreshortening angle being taken as the average of the angles subtended by the end point normals. Note the significant difference in the peak separation; the non-linearities are more subtle — cf. Figure E.3B).

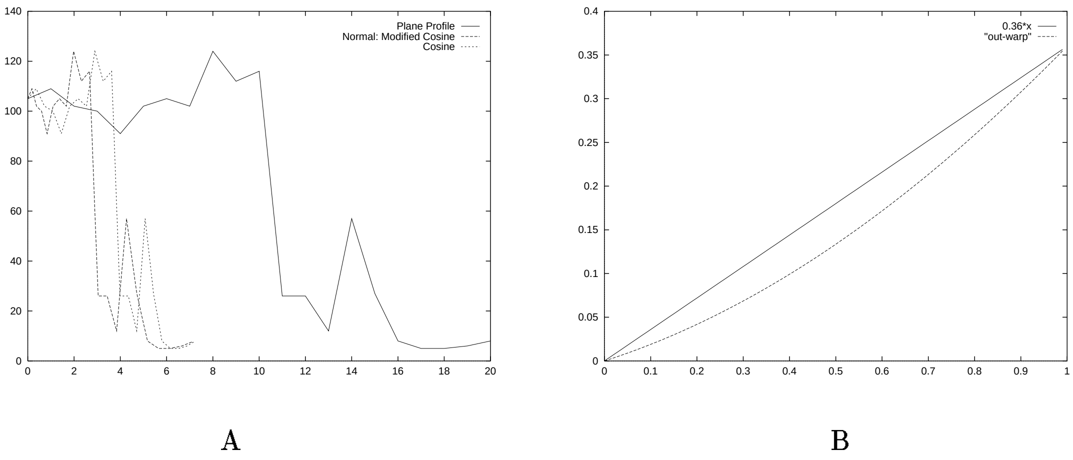


Figure E.3: Foreshortening Example A) A plane-sampled intensity profile and its foreshortened versions. Observe the significant discrepancy between the cosine warp and the modified cosine warp. It is clear that without any kind of correction, that the scale of the plane-sampled data can be dramatically different from that of the normal data. B) A Graph of the warping function $w(l)$ between 1.4 and 1.0 radians, respectively. Although the nonlinearity is subtle in this case, it becomes more pronounced when the angular separation increases. The value 0.36 corresponds to the $\cos(1.2)$, the average cosine foreshortening over this interval.

Appendix F

T2 Boundary Model Database

The 12 boundary models shown in Figure F.1 were used in the boundary detection phase of the segmentation scheme.

Model	ϵ	Edge Number
M1	0.3	1
M2	0.3	2
M3	0.3	1
M4	0.3	2
M5	0.3	1
M6	0.3	2
M7	0.3	1
M8	0.3	2
M9	0.3	2
M10	0.2	1
M11	0.2	1
M12	0.2	1

Table F.1: Model Parameters The value of ϵ and the edge index are shown for each model. The tolerance was the same for all models except the coarse models introduced to cope with boundary occlusion and foreshortening severe effects. A lower value was assigned to these models, since they were more likely to match undesirable features if not suitably constrained.

Each profile is shown to scale, and has an associated "tolerance", which corresponds to the parameter ϵ used in Chapter 5. This value determines the set of absolute constraints imposed on each model. If the value is low, the model can only exhibit low vertical variation; a high value means the model can exhibit significant vertical variation from instance to instance. For very large values, the associated absolute constraints become superfluous, and the model is only subject to relative segment constraints.

This will almost certainly result in erroneous boundary detection, particularly in regions where the boundary is occluded. The values of ϵ, as well as the edge number for each model, are shown in Table F.1.

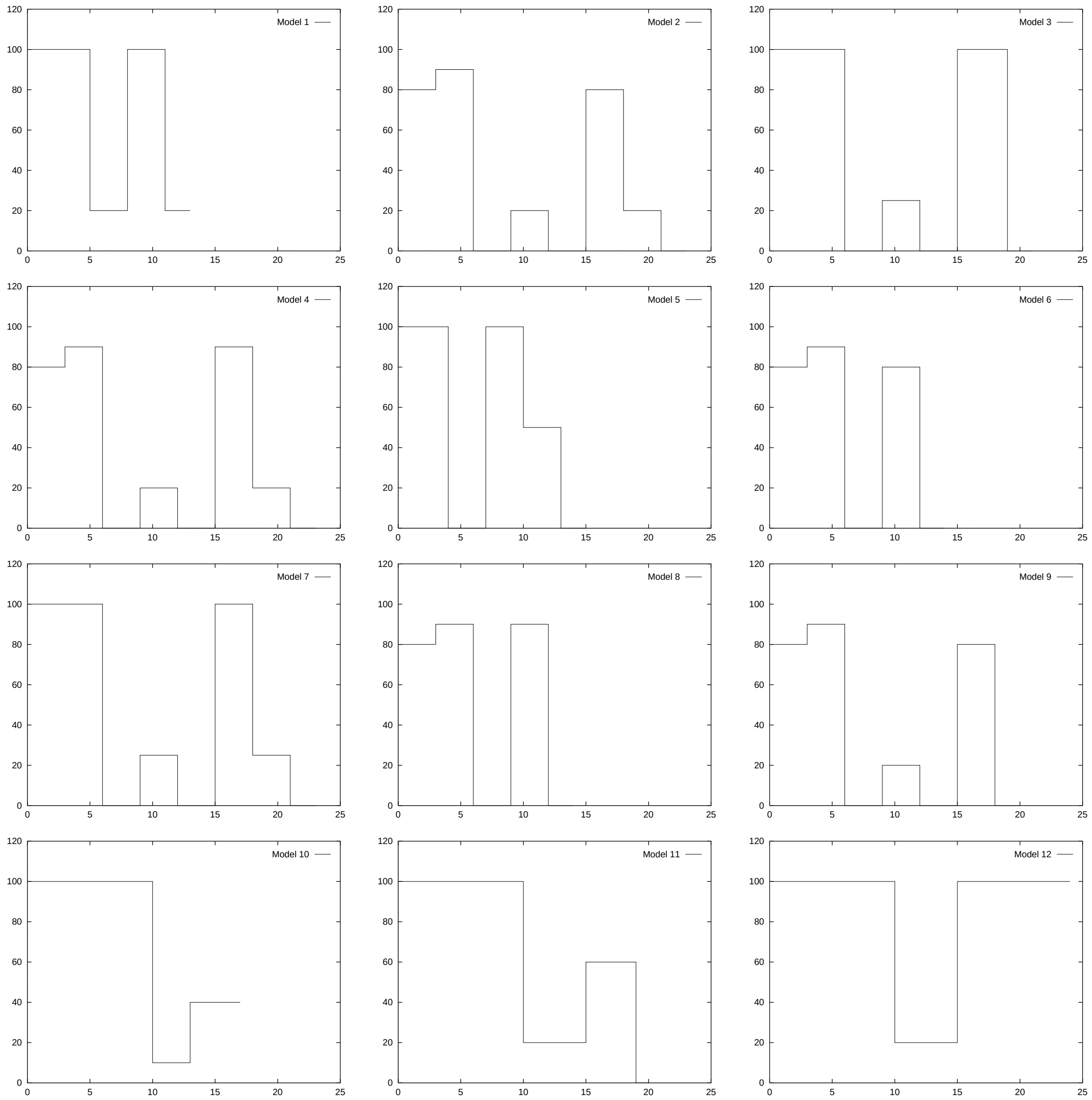


Figure F.1: Model Database The models shown here were used in the boundary detection phase of the segmentation scheme.

Bibliography

[1] N. Andreasan et al. Techniques for measuring sulcal/gyral patters in the brain as visualized through magnetic resonance imaging: brainplot and brainmap. *Proc. Natl. Acad. Sci.*, 90:93–97, January 1994.

[2] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[3] E. Bardinet, N. Ayache, and L. Cohen. Fitting of iso-surfaces using super-quadrics and free-form deformations. In *Proc. of the IEEE workshop on biomedical image analysis*, 1994.

[4] R. Bartels, J. Beatty, and B. Barsky. *An introduction to splines for use in computer graphics and geometric modelling.* Morgan-Kaufmann, San Mateo, California, 1987.

[5] P. Besl and N. McKay. A method for registration of 3d shapes. *IEEE PAMI*, 14:239–256, February 1992.

[6] R. M. Bilder et al. Absence of regional hemispheric volume asymmetries in first-episode schizophrenia. *American Journal of Psychiatry*, 151(10):1437–1447, Oct 1994.

[7] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Pres, 1995.

[8] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *Intl. Journal of Comp. Vis.*, 11(2):127–145, 1993.

[9] H. Blum. Biological shape and visual science (part i). *J. Theor. Biol.*, 38:205–387, 1973.

[10] H. Blum and R. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[11] F. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 449–511, May 1989.

[12] M. Brady and G. Scott. Parallel algorithms for shape representation. In I. Page, editor, *Parallel architectures and computer vision*, pages 97–118. Oxford Science Publications, 1988.

[13] C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, 61(2):154–170, March 1995.

[14] C. Brechbüler, G. Gerig, and G. Székely. Compensation of spatial inhomogeneity in mri based on a parametric bias estimate. In Karl H. Hoehne and Ron Kikinis, editors, *Proceedings of VBC'96*, volume 1131 of *Lecture Notes in Computer Science*, pages 141–146. Springer-Verlag, 1996.

[15] A. Brett, A. Hill, and C. J. Taylor. A method of 3d surface correspondence for automated landmark generation. In *BMVC*, pages 709–718. BMVA Press, 1997.

[16] E. Bullmore et al. Fractal analysis of the boundary between white matter and cerebral cortex in magnetic resonance images: a controlled study of schizophrenia and manic depressives. *Psychological medicine*, 24, 771–781 1994.

[17] E. Bullmore et al. Cerebral hemispheric asymmetry revisted: effects of handedness, gender and schizophrenia measured by radius of gyration in magnetic resonance images. *Psychological Medicine*, 25(2):349–363, 1995.

[18] H. Choi, D. Haynor, and Y. Kim. Partial volume tissue classification of multichannel magnetic resonance images. *IEEE Tran. Med. Img.*, 10(3):395–407, September 1991.

[19] R. Cipolla. *Active visual inference of surface shape.* PhD thesis, University of Oxford, Department of Engineering science, 1991.

[20] I. Cohen and L. Cohen. A hybrid hyperquadric model for 2-d and 3-d data fitting. *To appear: computer vision and image undertanding*, 1995.

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

[22] L. Cohen and I. Cohen. A finite element method applied to new active contour models and 3d reconstruction from cross sections. In *Proc. of the 3rd international conference on computer vision (ICCV'90)*, pages 587–591, 1990.

[23] T. Cootes and C. Taylor. Active shape model search using local grey-level models. In J. Illingworth, editor, *BMVC*, pages 639–648. BMVA Press, 1993.

[24] T. Cootes, C. Taylor, and D. Cooper et al. Active shape models: Their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[25] T. Cootes et al. Training models of shape from sets of examples. In *BMVC*, pages 9–18. Springer-Verlag, 1992.

[26] T. Cootes et al. The use of active shape models for locating sructures in medical images. *Image and Vision Computing*, 12(6):355–366, 1994.

[27] T.J. Crow, J. Ball, and S. Bloom et al. Schizophrenia as an anomaly of development of cerebral asymmetry. a post-mortem study and a proposal concerning the genetic basis of the disease. *Archives of General Psychiatry*, 46:1145–1150, 1989.

[28] T.J. Crow, R. Brown, and R. Bruton et al. Loss of sylvian fissure asymmetry in schizophrenia: findings in the runwell 2 series of brains. *Schizophrenia Research*, 6:152–153, 1992.

[29] J. Declerck, J. Feldmar, M. Goris, and F. Betting. Automatic registration and alignment on a template of cadiac stress and rest spect images. *IEEE Trans. Med. Imaging*, 16(6):727–737, December 1997.

[30] H. Delingette. Simplex meshes: A general representation for 3d shape reconstruction. Technical Report 2214, INRIA, Sophia Antipolis, March 1994.

[31] L. DeLisi et al. Loss of normal cerebral asymmetries in first-episode schizophrenia. *To appear: Archives General Psychiatry*, December 1994.

[32] I. Faux and M. Pratt. *Computational geometry for design and manufacture.* Ellis-Horwood, 1979.

[33] J. Feldmar. *Rigid, Non-rigid and Projective Registration of Three Dimensional Medical Images.* PhD thesis, INRIA, December 1995.

[34] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, June 1981.

[35] D. Gadian. *NMR and its applications to living system.* Oxford Science Publications. OUP, 1995.

[36] G. Gerig. *Multidimensional image analysis with applications to medical image data: (habilitation monograph).* PhD thesis, ETH, Zurich, 1992.

[37] L. Griffin, A. Colchester, S. Roll, and C. Studholme. Hierarchical segmentation satisfying constraints. In E. Hancock, editor, *Proceedings of the British Machine Vision Conference*, pages 135–144. BMVA Press, 1994. Sheffield.

[38] A. Gueziec. Large deformable splines, crest lines and matching. In *Proc. of the 4th international conference on computer vision (ICCV'93)*, pages 650–657, 1993.

[39] R. Guillemaud and M. Brady. Estimating the bias field of mr images. *IEEE Trans. Med. Img.*, 16(3):238, 1997.

[40] R. Guillemaud, P. Marais, and A. Zisserman et al. A 3-dimensional midsaggital plane for brain asymmetry measurement. *Schizophrenia Research*, 18(2-3):183–184, 1996.

[41] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP: Image Understanding*, 58(3):302–326, 1993.

[42] M. Hurn and K. Mardia et al. Bayesian fused classification of medical images. *IEEE Trans. Med. Img.*, 15(6):850–858, December 1996.

[43] J. Ivins and J. Porrill. Statistical snakes: active region models. In *Proceedings of the British Machine Vision Conference (BMVC'94)*, volume 2, pages 377–386, 1994.

[44] M. Kass, A. Witkin, and Terzopoulus. Snakes: Active contour models. In Proc. First Intl. Conf. on Comput. Vis., pages 259–268, 1987.

[45] R. Kikinis et al. Temporal lobe sulco-gyral pattern anomalies in schizophrenia: an in vivo mr three-dimensional surface rendering study. *Neuroscience Letters*, 182:7–12, 1994.

[46] Z. Liang, J. MacFall, and D. Harrington. Parameter estimation and tissue segmentation from multispectral mr images. *IEEE Trans. Med. Imaging*, 13(3):441–449, September 1994.

[47] P. Lipson and A. Yuille et al. Deformable templates for feature extraction from medical images. In O. Faugeras, editor, *Proc. 1st European Conf. on Comput. Vis.*, volume 427 of *Lecture Notes in Computer Science*, pages 413–417. Springer-Verlag, 1990.

[48] F. Maes, D. Vandermeule, P. Suetens, and G. Marchal. Computer-aided interactive object delineation using an intelligent paintbrush technique. In *Proc. of Computer Vision Virtual Reality and Robotics in Medicine (CVRMED'95)*, volume 905 of *Lecture Notes in Computer Science*, pages 77–83, April 1995.

[49] F. Maes et al. Automatic image partitioning for generic object segmentation in medical images. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Computational Imaging and Vision*, volume 3, pages 215–226. Kluwer Academic Publishers, June 1995.

[50] P. C. Marais, R. Guillemaud, and M. Sakuma et al. Visualising cerebral asymmetry. In *Proc. of Visualisation in Biomedical Computing*, volume 1131 of *Lecture Notes in Computer Science, 1996*, pages 411–416. Springer-Verlag, 1996.

[51] T. McInerney and D. Terzopolous. Topologically adaptable snakes. In *Proc. of the 5th international conference on computer vision (ICCV'95)*, pages 840–845, 1995.

[52] J. Montagnat and H. Delingette. Volumetric medical images segmentation using shape constrained deformable models. In *Proc. of Computer Vision Virtual reality and Robotics in Medicine (CVRMED'97),*, March 1997.

[53] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *Proc. of the 4th international conference on computer vision (ICCV'93)*, pages 275–279, 1993.

[54] A. Rinck. Magnetic resonance in medicine. Blackwell Scientific Publications, 1993.

[55] S. Röll and A. Colchester et. al. Intensity-based object extraction from 3d medical images including a correction for partial volume errors. In *BMVC*, pages 195–204. BMVA Press, 1994.

[56] R. Ronfard. Region-based strategies for active contour models. *Intl. J. of comp. vis.*, 13(2):229–251, 1994.

[57] L. E. Scales. *Introduction to Non-Linear Optimization.* Macmillan Computer Science Series. Macmillan, 1985.

[58] R. Sedgewick. *Algorithms.* Addison Wesley, 1988.

[59] M. Shenton et al. Abnomalities of the left temporal lobe and thought disorder in schizophrenia. *The new england journal of medicine*, 327(9):604–612, Aug 1992.

[60] H. Soltanian-Zadeh, J. Windham, and A. Yagle. Optimal transformation for correcting partial volume averaging effects in magentic resonance imaging. *IEEE Tran. Nucl. Sci.*, 40(4):1204–1212, August 1993.

[61] L. Staib and J. Duncan. Deformable fourier models for surface finding in 3d images. In R. Robb, editor, *Proceedings of VBC'92*, pages 90–104, 1992.

[62] G. Székely, A. Kelemen, C. Brechbüler, and G. Gerig. Segmentation of 2d and 3d objects from mri volume data using constrained elastic deformations of flexible fourier contour and surface models. *Medical Image Analysis*, 1(1):19–34, 1996.

[63] D. Terzopolous and A. Witkin. Symmetry-seeking models for 3-d object reconstruction. In *Proc. of the 1st international conference on computer vision (ICCV'87)*, pages 269–276, 1987.

[64] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7), 1991.

[65] N. Thacker, A. Jackson, X. Zhu, and K. Li. Accuracy of tissue volume estimation in nmr images. In *Second British Conference on Medical Image Understanding and Analysis*, pages 137–140, Leeds, England, July 1998.

[66] G.A. Thibodean and K.T. Patton. *Anatomy and Physiology.* Mosby, St Louis, 3rd edition, 1996.

[67] M. Vasilescu and D. Terzopoulos. Adapative meshes and shells. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR'92)*, pages 829–832, 1992.

[68] K. Vincken, A. Koster, and M. Viergever. Probabilistic segmentation of partial volume voxels. *Pattern Recognition Letters*, 15:477–484, May 1994.

[69] L. Wei-Chung, L. Cheng-Chung, and C. Chin-tu. Dynamic elastic interpolation for 3d medical image reconstruction from serial cross sections. *IEEE Trans. Med. Imaging*, 7(3):225–232, September 1988.

[70] W. Wells, W. Grimson, R. Kikinis, and F. Jolesz. In-vivo intensity correction and segmentation of magnetic resonance image data. *AAAI Med. Imaging*, pages 199–202, April 1994.

[71] R. Whitaker. Algorithms for implicit deformable models. In *Proc. of 5th international conference on computer vision (ICCV'95)*, pages 822–827, 1995.

[72] D. L. Wilson. *An Improved Planning Protocol for the Endovascular Treatment of Intracranial Aneurysms.* PhD thesis, University of Oxford, Department of Engineering Science, 1998.

[73] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, December 1994.

[74] A. Zijdenbos and B. Dawarit et al. Automatic detection of intracranial contours in mri images. *Comput. Med. Imag. and Graphics*, 18(1):11–23, 1994.

[75] R. Zipursky et al. Widespread cerebral gray matter volume deficits in schizophrenia. *Archives General Psychiatry*, 49:195–205, March 1992.