Compact Representations for Fast Nonrigid Registration of Medical Images

by

Samson J. Timoner

B.S., Caltech, Pasadena, CA (1997)
M.S., MIT, Cambridge, MA (2000)

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[June 2003]
May 2003

© Massachusetts Institute of Technology 2003. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2003

Certified by .....
W. Eric L. Grimson
Bernard Gordon Professor of Medical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

BARKER

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 07 2003
LIBRARIES

Compact Representations for Fast Nonrigid Registration of Medical Images

by

Samson J. Timoner

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2003, in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

We develop efficient techniques for the non-rigid registration of medical images by using representations that adapt to the anatomy found in such images.

Images of anatomical structures typically have uniform intensity interiors and smooth boundaries. We create methods to represent such regions compactly using tetrahedra. Unlike voxel-based representations, tetrahedra can accurately describe the expected smooth surfaces of medical objects. Furthermore, the interior of such objects can be represented using a small number of tetrahedra. Rather than describing a medical object using tens of thousands of voxels, our representations generally contain only a few thousand elements.

Tetrahedra facilitate the creation of efficient non-rigid registration algorithms based on finite element methods (FEM). We create a fast, FEM-based method to non-rigidly register segmented anatomical structures from two subjects. Using our compact tetrahedral representations, this method generally requires less than one minute of processing time on a desktop PC.

We also create a novel method for the non-rigid registration of gray scale images. To facilitate a fast method, we create a tetrahedral representation of a displacement field that automatically adapts to both the anatomy in an image and to the displacement field. The resulting algorithm has a computational cost that is dominated by the number of nodes in the mesh (about 10,000), rather than the number of voxels in an image (nearly 10,000,000). For many non-rigid registration problems, we can find a transformation from one image to another in five minutes. This speed is important as it allows use of the algorithm during surgery.

We apply our algorithms to find correlations between the shape of anatomical structures and the presence of schizophrenia. We show that a study based on our representations outperforms studies based on other representations. We also use the results of our non-rigid registration algorithm as the basis of a segmentation algorithm. That algorithm also outperforms other methods in our tests, producing smoother segmentations and more accurately reproducing manual segmentations.

Thesis Supervisor: W. Eric L. Grimson
Title: Bernard Gordon Professor of Medical Engineering

Readers: Jacob White, Professor of Electrical Eng. and Computer Science MIT.
William M. Wells III , Associate Professor Harvard Medical School.
Ron Kikinis, Associate Professor Harvard Medical School.

Acknowledgements

Let me begin by expressing my appreciation for the guidance provided by Eric Grimson, Sandy Wells, and Ron Kikinis over the last several years. Each has provided a different and important outlook on this research and has pushed me in interesting and clinically important directions. Most importantly, each has provided a research environment, filled with intelligent people, in which I have greatly enjoyed working.

Jacob White, my last committee member, provided support in a very different way: he has allowed me to attend his group meetings for years and thereby learn how to think about numerical methods. The concepts of spending a lot of computation time setting up a problem so that one can spend a lot less computation time solving a problem come directly from listening to him and his students. I have absorbed many ideas about "sparsifying" problems from his group. While most of these ideas don't show up explicitly in my thesis, they have been very important in shaping the way I think about trying to solve problems quickly.

I am thrilled to have had the opportunity to work with so many interesting and intelligent people at MIT and Brigham and Women's hospital. I have had numerous useful and interesting conversations with researchers including Alex Norbash, Steve Haker, Simon Warfield, John Fisher, Christian Shelton, Erik Miller, just to name a few.

This work could not be completed without the contributions of my colleagues. My most important collaborators are Polina Golland and Kilian Pohl. Polina Golland collaborated on the classification work in Chapter 4. Kilian Pohl and I jointly developed the automatic segmentation algorithm in Chapter 5.

Furthermore, I enjoyed the opportunity to work with clinical researchers. I collaborated with Dr. Martha Shenton and Dr. James Levitt investigating the correlation of shape of anatomical structures in the brain with the presense or absense of schizophrenia. I worked with Dr. Lennox Hoyte investigating the movement of female pelvis muscles. I worked with Dr. Arya Nabavi investigating intra-operative brain shift. I worked with Dr. Claire Tempany, detecting the movements of the prostate data.

I am grateful for the support of the Fannie and John Hertz Foundation; they have paid me a generous stipend for the last five years. The National Science Foundation (NSF) Engineering Research Center (ERC), under Johns Hopkins Agreement #8810-274, has supported this research. And, National Institutes of Health (NIH) grant 1P41RR13218 supports collaboration between Brigham and Women's Hospital and MIT.

MIT might have been a very lonely place without the support and companionship of my friends. Of the many people who have added cheer to my life in the last few years, I particular wish to acknowledge Professor E. Dickson, Dr. K Atkinson, Dr. C. Stauffer, and Dr. C Shelton.

Last but not least, this thesis would have had many more grammatical errors if it were not for my readers: Steve Haker, Charles Surasky, Biswajit Bose, Gerald Dalley, Kinh Tieu, Chris Christoudias, Henry Atkins, Polina Golland, and Kilian Pohl.

Contents

1 Introduction **17**

1.1 Motivating Concepts 20

1.2 Compact Representations of Anatomical Structures 21

1.3 Non-rigid Matching of Anatomical Structures 22

1.3.1 Morphological Studies 24

1.3.2 Segmentation 25

1.4 Non-rigid Registration of Images 26

1.5 Contributions 28

1.6 Conclusions 30

2 Forming Tetrahedral Meshes **31**

2.1 Goals 33

2.2 Desirable Properties of Tetrahedral Meshes 33

2.3 Prior Work 36

2.4 Mesh Formation 37

2.5 Multiple Resolution Meshes 38

2.6 Fitting a Mesh to a Surface 40

2.7 Mesh Improvement 41

2.7.1 Mesh Smoothing 42

2.7.2 Edge Collapse 44

2.7.3 Edge Flipping 44

2.8 Methods 45

2.8.1 Mesh Improvement 46

2.8.2 Compact Representations 48
2.8.3 Summary 50
2.9 Results 50
2.9.1 Smoothing 52
2.9.2 Compact Meshes 55
2.9.3 Other Multi-resolution Technologies 57
2.10 Discussion and Conclusion 58

3 Free Form Shape Matching **61**
3.1 Related Work 63
3.2 Choice of Representation 64
3.3 Methods 66
3.3.1 Representation 67
3.3.2 Matching Framework 67
3.3.3 Image Mesh Agreement Term 69
3.3.4 Solving Technique 71
3.3.5 Matching Parameters 74
3.3.6 Validation 74
3.4 Amygdala-Hippocampus Complex Dataset 75
3.5 Thalamus Dataset 76
3.6 Discussion and Conclusion 79

4 Application: Morphological Studies **83**
4.1 Background: Performance Issues in Morphological Studies 84
4.1.1 Choice of Shape Descriptor 84
4.1.2 Statistical Separation 88
4.1.3 Visualization of Differences 90
4.2 Methods 91
4.3 Results: Amygdala-Hippocampus Study 92
4.3.1 Comparison of Representation 93
4.3.2 Effects of Volume Normalization 95

4.3.3 Effects of Alignment Technique 95
4.3.4 Visualization of Differences 96
4.4 Results: Thalamus Study 101
4.5 Discussion 103
4.6 Conclusion 105

5 Application: Segmentation **107**
5.1 Overview of Segmentation Techniques 109
5.2 Segmentation Techniques Used 111
5.2.1 Atlas Matching 111
5.2.2 Inhomogeneity and Tissue Estimation 112
5.2.3 Deformable Models 114
5.3 Methods 115
5.3.1 Deformable Model 116
5.3.2 Error in Weight Estimates 120
5.3.3 Feedback of Shape Information 120
5.4 Experiments 121
5.5 Results 122
5.5.1 Algorithm Sensitivities 123
5.5.2 Segmentation without a Shape Prior 125
5.5.3 Segmentation with a Shape Prior 125
5.5.4 Validation 127
5.6 Discussion 129
5.7 Conclusion 131

6 Non-Rigid Registration of Medical Images **133**
6.1 Key Concepts 134
6.1.1 Previous Work 137
6.2 Methods 138
6.2.1 Initial Representation 138
6.2.2 Adaptive Representation of a Vector Field 140

6.2.3 Particulars of Image Matching 143
6.2.4 Determining the Deformation Field 145
6.2.5 Locally Maximizing the Objective Function 147
6.2.6 Comparison to Chapter 3 149
6.2.7 Summary 149
6.3 Results 150
6.3.1 Shape Matching 150
6.3.2 Brain Images 151
6.3.3 Female Pelvis Images 156
6.3.4 Computational Analysis 159
6.4 Discussion 159
6.5 Conclusion 161

7 Conclusion **163**
7.1 Future Directions of Research 164
7.2 Conclusion 166

Bibliography **167**

List of Figures

1-1 Rigid alignment example 18

1-2 Four left thalami 19

1-3 Transferring information to an intra-operative image. 20

1-4 Surface of a voxel and mesh representation of the entire prostate . . . 21

1-5 Four left thalami and correspondences found on their surfaces 23

1-6 Shape differences between thalami from controls and diseased individuals 24

1-7 3D models of the thalamus found using an EM-MRF algorithm, manual segmentation, and a new method that uses shape information 26

1-8 Images of a pelvis, brain and abdomen. 27

1-9 Non-rigid matching of intra-operative brain data 29

2-1 Surfaces of voxel- and mesh-representations of the prostate 32

2-2 A T-vertex 34

2-3 Poorly-shaped tetrahedra 35

2-4 Cube subdivision into tetrahedra 38

2-5 Subdividing tetrahedra 39

2-6 The most common ways a tetrahedron is intersected. 41

2-7 Mesh smoothing 42

2-8 Edge collapse 44

2-9 Edge swap 45

2-10 Example sphere meshes 51

2-11 Example amygdala-hippocampus meshes 51

2-12 Example Heschyl's gyrus mesh 51

2-13 Quality plot of a mesh 53
2-14 Mesh smoothing results: Heschyl's gyrus 53
2-15 Mesh smoothing results: sphere 54
2-16 Mesh smoothing results: complex 54
2-17 Quality plot of compact mesh 55
2-18 Thin slice through compact mesh representations 56
2-19 Example of template meshing 59
2-20 Multi-resolution meshes 59

3-1 Surface of manually segmented amygdala-hippocampus complex . . . 62
3-2 Illustration of volumetric attraction 66
3-3 Compact sphere representation 66
3-4 Representations used in shape matching. 67
3-5 Illustration of the sparsity structure of the elasticity matrix 69
3-6 Illustration of challenges in obtaining convergence of the non-rigid matcher 73
3-7 Location of the left amygdala-hippocampus complex in the brain . . . 75
3-8 Correspondences of six matched amygdala-hippocampus complexes . 77
3-9 80% Hausdorff distance between the deformable model and the target 78
3-10 Illustration of errors in non-rigid shape matching 78
3-11 Location of the left thalamus complex in the brain. 79
3-12 Thalamus correspondences 80
3-13 Hausdorff distances in the thalamus matching study 80

4-1 Distance map representation of a complex 85
4-2 Illustration of the challenge of aligning shapes 87
4-3 Similarity matrix between shapes 93
4-4 Effect of alignment methods on classification accuracy. 96
4-5 Shape differences of right amygdala hippocampus complex 98
4-6 Shape differences of left complex 99

4-7 Amygdala-hippocampus complex shape differences found using a linear classifier . . . 100
4-8 Shape differences of the thalamus . . . 102

5-1 Confounding effects of segmentation of MR . . . 108
5-2 Slice through an atlas of white matter . . . 111
5-3 The main steps in the segmentation algorithm . . . 116
5-4 Deformable tetrahedral mesh . . . 117
5-5 Convergence of updated tissue probabilities . . . 121
5-6 Eigenvectors of the thalamus deformable model . . . 122
5-7 Variance captured by PCA of the right thalamus . . . 123
5-8 Challenges in matching deformable models to the results of the EM-MRF algorithm. . . . 124
5-9 3D models of the thalamus found using an EM-MRF algorithm, manually, and with the addition of shape information . . . 126
5-10 Comparison between an automatic segmentation of the thalamus and a manual segmentation without shape prior . . . 126
5-11 Comparison between an automatic segmentation of the thalamus using shape information and a manual segmentation . . . 126
5-12 Dice similarity comparison . . . 127
5-13 Hausdorff similarity comparison . . . 128
5-14 A brain segmentation . . . 129

6-1 Block matching example . . . 135
6-2 Images of a pelvis, brain and abdomen. . . . 136
6-3 Adapting mesh to anatomy . . . 139
6-4 Adaptive flow chart . . . 140
6-5 Adaptive representation of a vector field . . . 141
6-6 Representation of a sinusoidal vector field . . . 143
6-7 Thin slice through a mesh of an ellipse . . . 144
6-8 Non-rigid matching of two ellipses . . . 150

6-9 Non-rigid matching of segmented prostate data 151
6-10 Non-rigid matching of intra-operative brain data 152
6-11 Mesh adaptation for the brain image 153
6-12 Comparison of manually labeled fiducials in a brain image sequence . 155
6-13 Female pelvis non-rigid matching sequence 157
6-14 Points at which the pelvis warping was validated. 158

List of Tables

2.1 The effect of edge collapse on example meshes 57

4.1 Comparison of classifiers based on distance maps and displacement fields 94
4.2 The effect of volume normalization on classifer accuracy 94
4.3 Cross training accuracy for the thalamus 103

6.1 Errors in female pelvis non-rigid registration 158

Chapter 1

Introduction

Advances in non-invasive imaging have revolutionized surgery and neuroscience. Imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT) scanning yield three-dimensional images of the insides of living subjects. This information allows physicians to diagnose and plan treatment without opening the patient or to carefully plan surgery to avoid important anatomy. Additionally, researchers can use these images to investigate anatomical structures in living patients, rather than in cadavers. In fact, not only can scientists investigate the structure of anatomy, but through techniques such as functional magnetic resonance imaging (fMRI), they can infer the function of anatomy.

Neuroscientists and clinicians have created a demand for medical image processing tools. One example is the desire for tools that automatically register three-dimensional images such as those in Figure 1-1. Physicians sometimes look for changes between two images by overlaying them. However, because of differences in patient positioning during each image acquisition, images generally need to be aligned before being compared. Physicians would like to avoid the somewhat tedious process of manual registration. Instead, they prefer an automatic way to register volumetric images.

In many cases, even after alignment, simply overlaying two images is not sufficient to compare them. For example, a common procedure to detect tumors in the liver is to inject an agent into a patient that preferentially enhances tumors in MRI images.

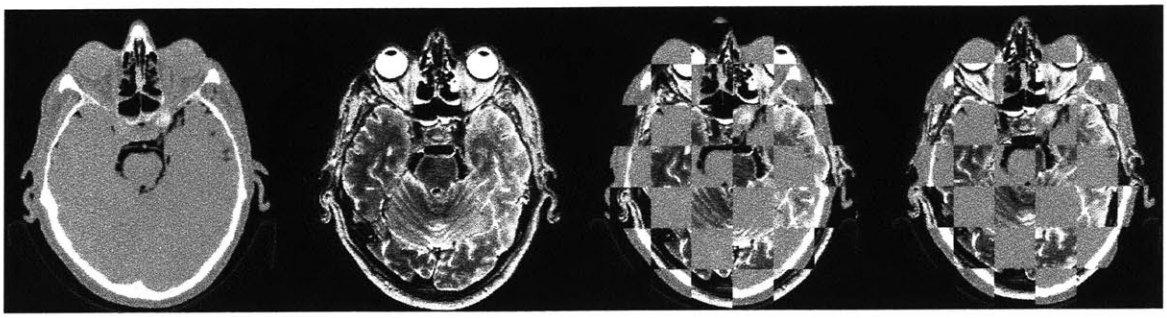


Figure 1-1: From left to right: A slice through a three-dimensional CT of a human head; a slice through a three-dimensional MRI (T2) of the same subject; a checkerboard visualization of the two images overlaid; a checkerboard visualization of the two images overlaid after a manual registration. Without registration (center right) the nose and ears and eyes are clearly not registered properly. It is therefore difficult to compare the images by overlaying them. The manual rigid registration (right) makes comparing the images much easier.

To decide if a region contains tumor, doctors will compare two images in time and see if that region becomes brighter. However, as the liver expands and contracts upon breathing [Ros02], the regions move and deform. Thus, overlaying two images of the same liver at different points in the breathing cycle makes comparing the two images difficult. In order to readily make that comparison, one would like to non-rigidly register the images. Non-rigid registration is the process of registering two images while allowing the images to deform into each other.

Not only is non-rigid registration useful in comparing images of the same subject, it is useful in comparing images of different subjects. Physicians are interested in comparing images of different subjects to find differences in anatomy due to biological processes such as aging or disease. For example, there is evidence that the size and shape of some anatomical structures in the brain correlate with the presence or absence of schizophrenia [SKJ+92]. One way to compare anatomical structures across subjects is to use non-rigid registration. An example of anatomical structures to be compared is shown in Figure 1-2.

Non-rigid registration has another important application other than comparing images; it is also used to map information from one image to another. This second application is of particular importance for intra-operative imaging. Intra-operative

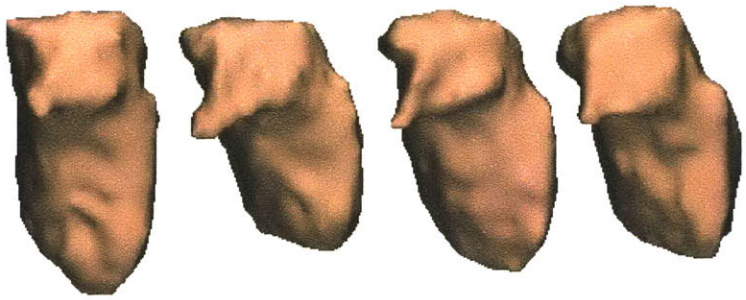


Figure 1-2: The surfaces of four left thalami. The examples were taken from subjects in a study trying to correlate the presence of schizophrenia with the shape of the thalamus. Note the differences in size between the thalami, as well as the difference in the shape of the "hook-shaped" part of the thalamus (the lateral geniculate nucleus). The left thalamus is a structure in the left part of the brain.

imaging is the process of acquiring new images during surgery. These images give surgeons updated information about the location of deforming tissues. Unfortunately, intra-operative images are often of lower quality than pre-surgical images[1]. Additionally, while physicians have time to annotate pre-operative data with surgical path planning and segmented structures, they rarely have time to stop surgery and annotate intra-operative data. An example of the need to map information onto intra-operative data is shown in Figure 1-3 for the case of a prostate surgery. The major zones of the prostate have carefully been identified in an image taken weeks before surgery (left). Physicians are would like to target the peripheral zone in the prostate because they believe it is the source of cancer growth. Unfortunately, the intra-operative image (right) does not have sufficient contrast to view the separate zones. This surgery would benefit from updated information on the location of the peripheral zone; that information could be obtained using non-rigid registration.

While surgeons could benefit greatly by non-rigidly mapping augmented, high-contrast, pre-operative data to intra-operative data, in practice non-rigid registration is rarely used during surgery. Most non-rigid registration algorithms require several

[1]The lower quality results partly from the fact that during surgery, physicians are unwilling to wait the long periods necessary to obtain high contrast images. Also important is that in order to make surgery and imaging compatible, intra-operative imaging equipment can be less sensitive than typical imaging equipment. For example, intra-operative MRI imagers use significantly lower magnetic fields than normal imaging magnets, resulting in lower contrast images.

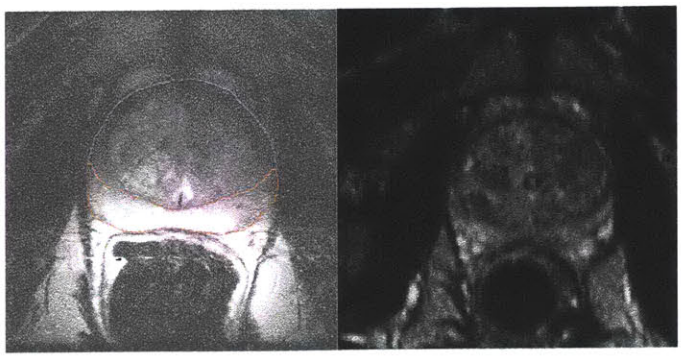


Figure 1-3: Left: pre-operative image of a male pelvis. The prostate central zone has been outlined in blue; the prostate peripheral zone has been outlined in red. Right: image taken just before surgery commences. In the right image, the peripheral zone is difficult to see. Physicians would benefit by knowing its location. They could obtain that information by non-rigidly registering the pre-operative image (left) to the inter-operative image (right).

hours to complete, making them unusable during surgery.

In summary, neuroscientists and clinicians have created a demand for medical image processing tools. One such tool is non-rigid registration which is useful for comparing images across subjects, in comparing images of the same subject, and in general mapping information from one image to another. It is our goal to make contributions to the field of non-rigid registration of three-dimensional images. In particular, we will develop methods to compare anatomical structures across subjects and to make a non-rigid registration algorithm that is fast enough to be used during surgery.

1.1 Motivating Concepts

To create efficient non-rigid registration algorithms, we note that in many fields of research, the use of a representation that is related to the structure of a problem leads to compact descriptions and efficient algorithms. For example, variations in camera images can be dominant in any direction. Freeman found that by using filters that could be aligned to the direction of variation, he could make far more compact descriptions of the image and more effective algorithms based on that representation [Fre92]. As another example, when solving Maxwell's equations, the potential due to

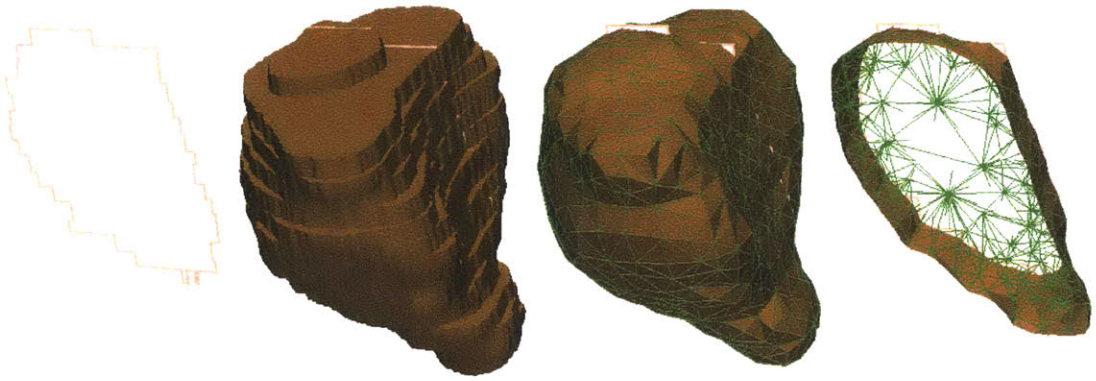


Figure 1-4: Left: Slice through a voxel representation of a prostate. Center Left: Surface of a voxel representation of a prostate. The region encompasses 80,000 voxels. Center Right: Surface of a tetrahedral mesh representation of a prostate using 3500 tetrahedra. Right: thin slice through the mesh. The representation using tetrahedra is roughly 20 times smaller and better represents the expected smooth surface of the anatomy.

a source of charge varies quickly close to the source, and slowly far away. Multipole methods effectively use representations that are adapted to this rate of variation [Phi97]. By using such representations, multi-pole methods for solving Maxwell's equations can be made much more computationally efficient than other methods.

In medical images, the primary structure is due to anatomy. We believe that using representations that conform to the shapes of anatomical images will lead to compact representations of data, as well as efficient algorithms. We therefore begin by developing compact representations of anatomical objects using tetrahedra. Those tetrahedra will facilitate the creation of efficient finite element based methods to non-rigidly align medical images.

1.2 Compact Representations of Anatomical Structures

Images of anatomical structures frequently have uniform intensity interiors and smooth surfaces. A typical medical object is segmented and then described using tens of thousands of voxels, with each voxel storing redundant data (a voxel is the name given to a point in a three-dimensional image). Such regions are inefficiently described

using a uniform grid of points. In fact, not only is the description inefficient, but the resulting surface of such a representation is described as an unrealistic jagged edge (see Figure 1-4). We seek a volumetric representation that more accurately represents smooth surfaces while representing uniform regions efficiently. We propose using volumetric meshes to describe medical objects. Volumetric elements have the ability to compactly represent large uniform regions. The surfaces of such elements can be chosen to correspond to the surfaces of medical objects, so that smooth surfaces can be well described. The interior of medical objects can be filled with a small number of volume elements. For these reasons, volumetric meshes should represent medical data more efficiently than uniform grids of voxels.

We develop methods to create tetrahedral meshes to describe anatomical structures. We choose tetrahedra because they are the simplest three-dimensional volumetric element; compared to other volumetric elements with more nodes, tetrahedra are relatively easy to manipulate. However, as we describe in Chapter 2, there are numerous challenges to overcome to obtain a good tetrahedral representation of an object. Figure 1-4 compares a voxel based representation of a prostate and a representation using a mesh of tetrahedra we created. The voxel based representation uses 80,000 voxels. The tetrahedral mesh based representation is much smaller; it uses only 3500 tetrahedra.

1.3 Non-rigid Matching of Anatomical Structures

Having developed compact representations of an anatomical structure, we proceed to use those representation to develop fast algorithms to non-rigidly register anatomical structures from different subjects. As tetrahedra facilitate the creation of efficient algorithms based on finite element methods (FEM), we develop FEM based methods to find a registration.

Finding a good non-rigid registration between two medical shapes can be challenging because medical shapes typically have smooth surfaces. As shown for the thalamus in Figure 1-2, it is not obvious where points in one smooth surface should

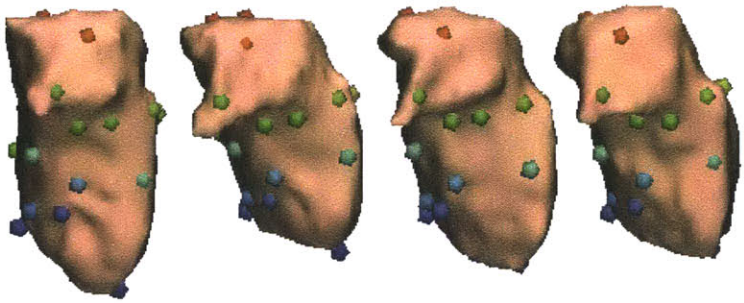


Figure 1-5: The surfaces of four left thalami with colored spheres indicating correspondences.

lie on a second such surface. To overcome this challenge, we register anatomical shapes while minimizing an elastic energy. That is, during the matching process, we treat one of the objects like a linear elastic material. We then try to find a transformation from the first shape to the second that maximizes overlap and minimizes a linear elastic energy. Intuitively, in a good match, high curvature regions in one object should match to high curvature regions in a second object. Matching by minimizing a linear elastic energy should accomplish this goal. Matching a sharp portion of one surface against a sharp portion of another surface results in a lower energy configuration than flattening a region to match against a less sharp adjacent region.

Unfortunately, such linear-elastic matching leads to a set of ill-conditioned equations that are difficult to solve. However, while the equations may be poorly conditioned, the problem is not. We present a fast, linear-time algorithm that overcomes the ill-conditioned nature of the equations. Because we use compact tetrahedral representations during this process, the resulting algorithm generally finishes in less than 30 seconds on a desktop machine. Using the non-rigid registration algorithm, we can take anatomical shapes such as the thalami shown in Figure 1-2, and find correspondences on the surface such as those shown Figure 1-5.

Once anatomical shapes have been registered across subjects, we use the resulting transformations in two applications. First, we use the transformation as a way to compare shapes in a morphological study. Second, we create a segmentation algorithm that uses the measured variability in a set of structures to identify new examples of structures.

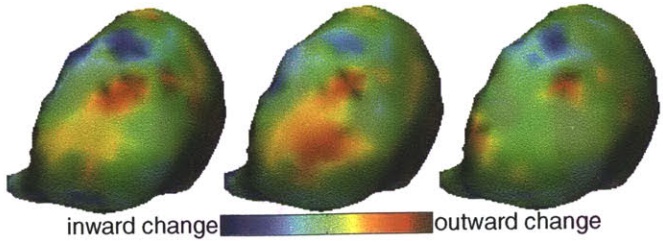


Figure 1-6: Shape differences of the left thalamus between normal and first episode schizophrenic subjects. The thalami are from three normal subjects. The color coding indicates how to change the surface of the thalami to make them be more like the left thalamus in schizophrenic subjects.

1.3.1 Morphological Studies

As we have already described, researchers are interested in comparing anatomical structures to correlate shape with biological processes, such as the effects of aging and disease. We show that correspondences found using the method discussed in the last section are effective for use in a study of shape.

We develop methods to find shape differences between two groups. We choose a reference shape and then represent each other shape using the displacement field from the reference to the other shape. We then form a classifier using support vector machines to separate the two groups of shapes. Finally, using the methods developed by Golland *et al.* [GGSK01], we take derivatives of the classification function to determine shape differences.

In this work, we use these methods to find shape differences in anatomical structures in the brain between groups of normal subjects and groups of schizophrenic subjects. In particular, we examine the amygdala-hippocampus and the thalamus. For the amygdala-hippocampus, we not only show that displacement fields are effective at determining shape differences, but they are more effective than methods based on another representation [Gol01]. For the thalamus, we find shape differences that were hitherto unknown. Those shape differences are shown in Figure 1-6.

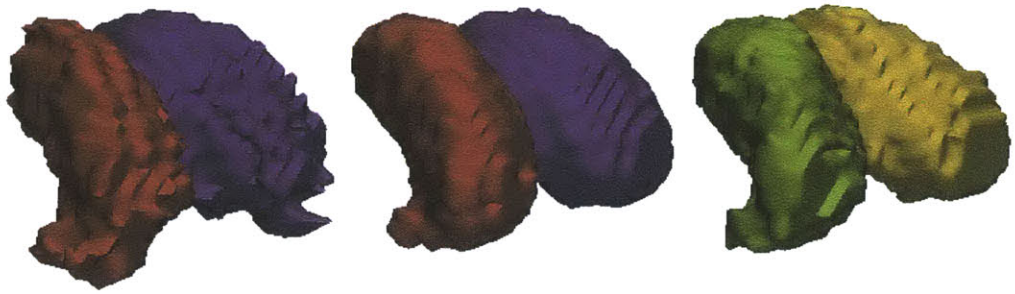
1.3.2 Segmentation

The morphological studies that we perform are based on manual segmentations of shape. That is, a researcher identified voxels in the image that she believed contained the tissue of interest. Manual segmentation is a tedious, time consuming and error prone task [Dem02, KSG+92]. For studies involving tens of cases, manual segmentation may be a viable option. However, for studies involving hundreds or thousands of cases, manual segmentation can be impractical.

For large studies, it is desirable to use automatic methods to identify anatomical structures. Unfortunately, automatically identifying tissue types in medical images has proven challenging. That challenge is partly due to noise and image inhomogeneities, and partly due to the nearly invisible borders between some anatomical structures.

Automatic, intensity based segmentation methods have been very successful in dealing with some of those challenges. In particular, such methods are able to handle imaging noise and image inhomogeneities. However, these methods have difficulties finding low contrast borders. Deformable models are a second class of methods that, conversely, often work well with nearly invisible boundaries. These methods create a generative model based on training representations of shape, and try to segment by fitting that generative model to a new image. But, these methods also have drawbacks. First, it is challenging to accurately model the variability in many structures. Second, these methods are typically trained on images from one imager. They therefore have difficulties with other types of images, or even other imagers of the same type.

We develop a new method that combines the deformable model and intensity based methods. We use an intensity based method to capture all information in the image that is known about the various tissue classes to be segmented. That method produces an image of probabilities showing which voxels are most likely in which tissues. We then fit a deformable model to those tissue probabilities. The resulting fit can be given back to the intensity based method so that method can incorporate



(a) Prior Method (b) New Method (c) manual segmentation

Figure 1-7: 3D models of the right (maroon) and left (violet) thalamus generated by (a) existing intensity based algorithm, (b) the new algorithm and (c) a manual segmentation. Note that the thalami produced with a shape prior (b) are significantly smoother than those produced without a shape prior, which have protrusion and sharp edges.

all information.

To accomplish that fit, we create a deformable model based on the correspondences found by our non-rigid registration algorithm. We also create a new non-rigid registration algorithm to fit the deformable model to the image of probabilities produced by the intensity based method.

We show that adding shape information into an intensity based method has significant advantages over a such a method alone. We segment the thalamus within an MRI of the brain and find that our new method yields boundaries which are significantly smoother than an intensity based method. Also, the new method yields boundaries which are significantly closer to those of manual segmentations. A comparison of the three segmentations can be found in Figure 1-7.

1.4 Non-rigid Registration of Images

In the research discussed up to this point, we adapted representations to anatomical structures that had been previously identified. In this section, we describe methods to non-rigidly register images, where anatomical structures have not previously been identified.

To make a fast non-rigid registration algorithm, we make the key observation that the information most useful for non-rigid registration is not uniformly distributed in

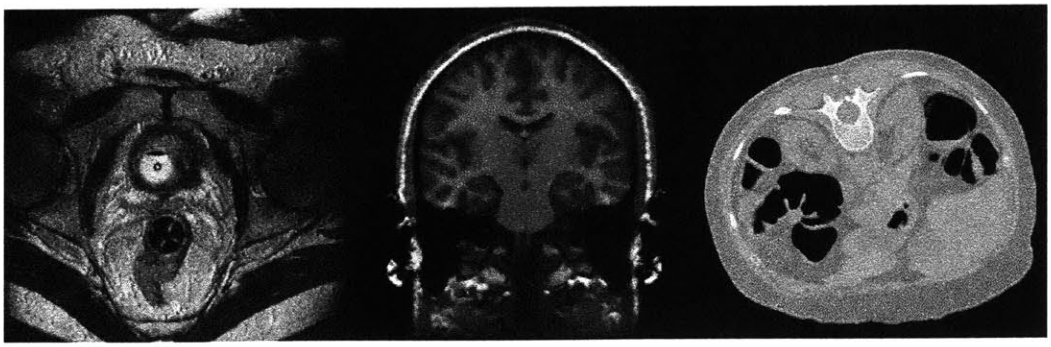


Figure 1-8: From left to right: Axial MRI of male pelvis, coronal MRI of a brain, and axial abdomen CT. These three images are examples of medical data appearing nearly uniform over large regions. In the pelvis, the muscle and fat has roughly uniform intensity. The white matter in the brain is nearly uniform intensity. In the CT image, the interior of the colon (in black), the liver (lower right), as well as other soft tissue all appear nearly uniform.

an image; it is generally concentrated at the boundaries of anatomical structures. As shown in Figure 1-8, large fractions of the image have uniform intensity (or texture). One would expect that by finding the displacement field in those regions, one could accurately capture the majority of the displacement field, which could be interpolated into regions of uniform intensity. Thus similarly to previous methods we automatically adapt representations to the anatomy in the image.

We also make a second key observation: the displacement fields between images need not be represented at the same resolution everywhere. Displacement fields between pre-operative images and intra-operative images are generally continuous and smooth (though there are exceptions which can be dealt with explicitly such as in [MSH+02].) and slowly varying in large portions of the image. For example, a cursory review of brain warping papers suggests that displacement fields often change very quickly near an incision, but much more slowly far away it [FMNW00, HRS+99, MPH+00, SD99]. Because displacement fields are mostly smooth, regions of slowly varying displacements can be accurately described using a small number of vectors that are interpolated within those regions. Thus, one way to create a compact representation of a displacement field is to identify regions of slowly-varying displacements and to represent those regions with as few vectors as possible.

To take advantage of the two observation made, we develop a representation of

a displacement field using the nodes of a mesh of tetrahedra. The nodes of the tetrahedral mesh can be created using different densities in different parts of the image, and selectively adapted as needed. Furthermore, the surfaces of the tetrahedra can be aligned to the surfaces of anatomical structures in the image. If segmentations are available, it is straightforward to include the surfaces of such regions directly in the mesh. If segmented regions are not available, the nodes of the mesh can be moved to places of large intensity variations where information for non-rigid registration is maximized.

We create a non-rigid registration algorithm based on this representation. The resulting algorithm has a computational cost that is dominated by the number of nodes in the mesh (often roughly 10000), rather than the number of voxels in an image (nearly 10 million). For many non-rigid registration problems, we can find a transformation from one image to another in five minutes. As example of the results of the algorithm is shown in Figure 1-9. Note how the mesh is concentrated in the region of largest deformations.

1.5 Contributions

In this thesis, we are concerned with using the structure of anatomical images to make efficient non-rigid registration algorithms. To that end, we contribute the following algorithmic developments in this thesis:

1. Methods to compactly represent medical shapes with tetrahedra.
2. Methods to quickly non-rigidly register anatomical shapes.
3. Methods to use the results of the non-rigid algorithm to form a representation of shape to be used in a morphological study.
4. A segmentation algorithm using the correspondences found with the non-rigid registration algorithm.
5. An adaptive non-rigid registration algorithm that adapts to both anatomy and to the displacement field so that the dominant computational cost need not scale linearly with the number of voxels in an image.

We use those methods to create the following clinical results:

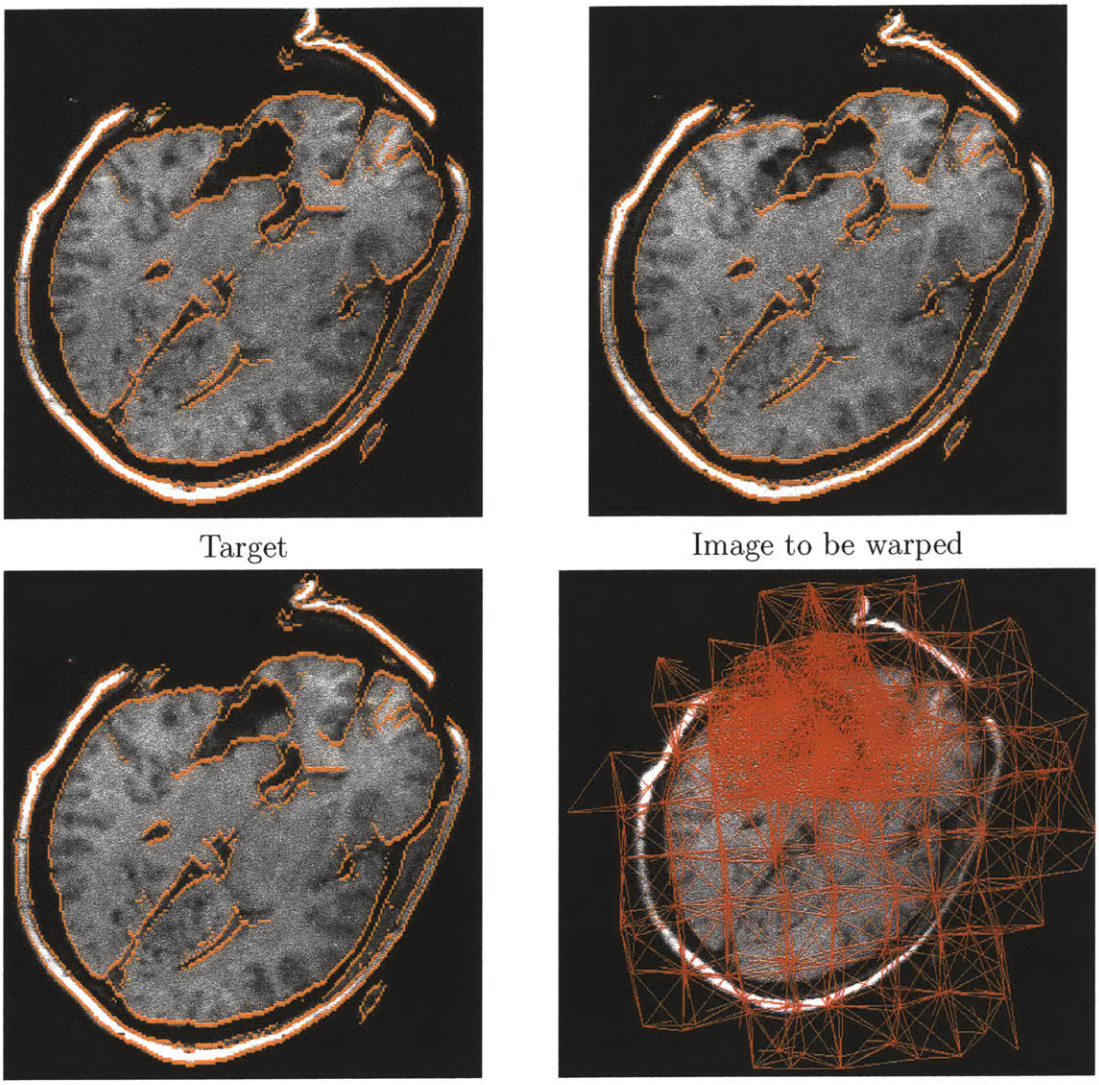


Figure 1-9: Top left: MRI of a brain taking during surgery with edges highlighted. Top right: MRI taken of the same brain later in the surgery, with the edges from left image overlayed. Bottom left: the result of warping the top right image onto the top left image, with the edges of the top-left image overlaid. The non-rigid registration algorithm has effectively captured the deformation. The bottom right shows a mesh used in the registration process. Note how the mesh is very coarse in most areas, and fine in areas near the incision where most of the deformation is.

1. Correlating shape differences between the presence and absence of schizophrenia with the shape of the thalamus and the shape of the amygdala-hippocampus complex.
2. An automatic brain segmentation algorithm.
3. A non-rigid registration algorithm that is fast enough to be used during surgery.

1.6 Conclusions

We start by making representations that adapt to anatomy which we then use to create efficient algorithms to match anatomical structures. Using the results of that algorithm, we create effective methods to find shape differences in a morphological study and an effective segmentation algorithm.

We also recognize that displacement fields between images are often slowly changing in large portions of the image so that dense representations of a displacement field need not be used in the entirety of the image. Using these observation and adapting our representations to anatomy, we create a fast non-rigid registration algorithms.

Overall, we use compact representation that adapt to the structure of the non-rigid registration problem in order to make fast and effective algorithms.

Chapter 2

Forming Tetrahedral Meshes

The main objective of this thesis is to create compact descriptions of medical objects in order to make computationally efficient non-rigid registration algorithms. In this chapter, we focus on finding compact representations of anatomical structures. Anatomical structures often have uniform intensity interiors and smooth surfaces. A typical medical object is described using tens of thousands of voxels, with each voxel effectively storing redundant data. Not only is the representation inefficient, but the resulting surface of such a representation is an unrealistic jagged edge. Figure 2-1 shows an example of the prostate central and peripheral zone being described using this representation.

We seek a description of a region that more accurately represents smooth surfaces and represents uniform regions efficiently. We propose using volumetric meshes to describe medical objects. Volumetric elements have the ability to compactly represent large uniform regions. The surfaces of such elements can be chosen to correspond to the surfaces of medical objects, so that smooth surfaces can be well described. The interior of medical objects can be filled with a small number of volume elements. For these reasons, volumetric meshes should represent medical data more efficiently than uniform grids of voxels.

We have chosen to use meshes of tetrahedra. Tetrahedra are a good choice to represent medical data since it is particularly straightforward to use them to describe smooth surfaces. Also, tetrahedra are the simplest three dimensional volumetric

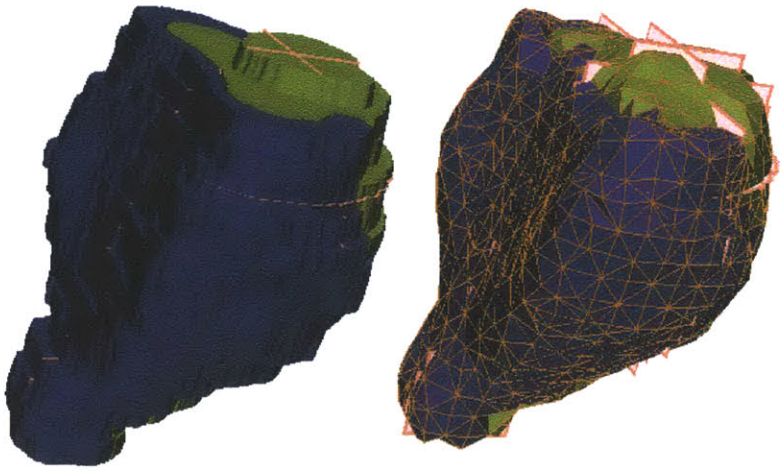


Figure 2-1: Left: Surface of voxel representation of a prostate central (green) and peripheral zone (blue). The region encompasses eighty thousand voxels. Right: the same region represented after smoothing the left surface and filling with tetrahedra. The brown lines represent the surface edges of the mesh. The region is described by seven thousand tetrahedra. The tetrahedra-based representation is much smaller than the voxel-based representation and better describes the expected smooth surface of the prostate.

element; compared to other volumetric elements with more nodes, tetrahedra are relatively easy to manipulate. Figure 2-1 shows a prostate represented using a tetrahedral mesh; the resulting representation uses a factor of 10 fewer nodes than the voxel based representation and better describes the smooth surface of the prostate.

Tools to fill shapes with tetrahedra are typically designed to process surfaces from Computer Aided Design (CAD) programs. Medical data has two key properties not typically found in CAD data. First, the surfaces of medical structures are often irregular, including non-convex regions. Second, it is usually the case that any set of nodes that describe a medical object's surface are equally acceptable descriptions of that surface. Therefore, it is not important that the nodes of a tetrahedral mesh exactly overlap those of the original surface, as long as the surface is well described. Here, well described means that the volume overlap between the tetrahedral mesh and the surface interior is near 100%, and the distance from the surface to the surface of the mesh is small enough.

In this chapter, we begin by describing the goals of a tetrahedral mesher for medical data. We then review desirable properties of tetrahedral meshes. We discuss

the mesh formation methods we will use, and mesh improvement methods. We then present resulting tetrahedral meshes, showing that the quality of the meshes is above our target threshold.

2.1 Goals

In this chapter, we create a toolkit of meshing techniques. We incorporate a subset of those techniques into two different algorithms to satisfy the needs of Chapter 3 and Chapter 4.

For Chapter 3, we will create deformable models from tetrahedral meshes to deform a medical shape from one subject into the same medical shape in another. Thus, we will solve a set of partial differential equations (PDE) on the tetrahedral mesh. As we have no prior information on the variability of one part of the surface of a structure over another part, we desire a roughly uniform description of the surface of the structure. That is, we desire the faces of the tetrahedra that determine the surface to be approximately the same size. To create an efficient algorithm, we will also desire the interior of the structure to be as sparse as possible. Thus, for Chapter 3, we will create a approximately uniform description of the surface at a resolution specified by the programmer, while making the interior described as sparsely as possible.

In Chapter 4, we will desire not only a roughly uniform tiling of the surface, but a uniform description of the interior. In that chapter, we will use the displacement field of the deformable model as a representation of shape. As we desire a dense sampling of the displacement field inside the shape, we will create meshes with roughly uniform node density in the interior. Thus, a second algorithm in this chapter will create a roughly uniformly description of the interior and surface of a structure.

2.2 Desirable Properties of Tetrahedral Meshes

Tetrahedral meshes are generally used to solve partial differential equations (PDE). Typically, the tetrahedra are finite elements that represent small chunks of material.

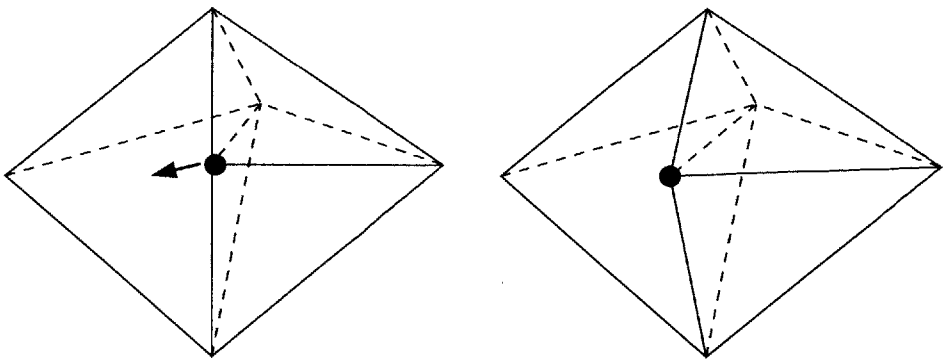


Figure 2-2: Left: Three tetrahedra. The dot indicates a t-vertex: a vertex on the edge of the left tetrahedron but not one of the corners of that tetrahedron. If that vertex moves a small amount as indicated by the arrow, the leftmost tetrahedron ceases to be a tetrahedron as shown in the right image.

In that role, there are several desirable properties of the tetrahedra. To review those properties, consider the problem of solving for the final position of a block of rubber that is being compressed. To be accurate, it is clear that the volume elements describing the rubber must describe the entire block. Holes in the rubber would ignore interactions between small elements of rubber. Thus, the first desirable property of a tetrahedral mesh is that it must completely fill the region being simulated.

A second desirable property of tetrahedral meshes is that no edge crossings are allowed. An edge crossing forms what is know as a T-vertex; such a vertex is shown in Figure 2-2. If the vertex in the figure moves, and all other vertices remain fixed, the left most tetrahedron will cease to be a tetrahedron. There are ways to address this problem [GKS02], but in this chapter we will avoid the additional complications of such methods by not allowing such vertices.

A third desirable property of tetrahedral meshes is that the tetrahedra be nearly equilateral. Tetrahedra that are particularly skewed slow down and produce errors in the solutions given by partial differential equation (PDE) solvers [Ber99, FJP95]. The reason for these problems is that the equations corresponding to skewed tetrahedra can be very poorly conditioned. Bern and Plassman classify "poorly-shaped" tetrahedra into the five classes as shown in Figure 2-3 according to dihedral angle and solid angle (Dihedral angles are the angles between triangles in a tetrahedron). In

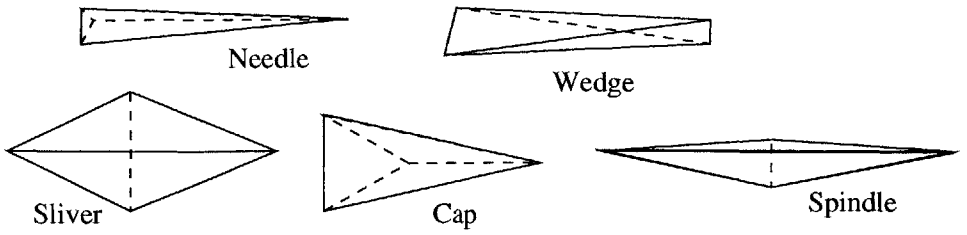


Figure 2-3: Examples of the five classes of poorly-shaped tetrahedra [BPar] . A "needle" has the edges of one triangle much smaller than the other edges. A "wedge" has one edge much smaller than the rest. A "sliver" has four well-separated points that nearly lie in a plane. A "cap" has one vertex very close to the triangle spanned by the other three vertices. A "spindle" has one long edge and one small edge.

each case, a small motion of one vertex relative to the average length of a tetrahedron edge can cause the tetrahedron to have zero volume or "negative volume". Returning to the rubber simulation example, this situation corresponds to a small piece of the rubber compressing to zero volume and then continuing to compress, an unphysical result.

To avoid numerical problems, it is important to assure the quality of the tetrahedra in a mesh. There are several useful quality metrics [Ber99, LDGC99] which measure the skewness of tetrahedra. The metrics are 0 for a degenerate tetrahedron and 1 for an equilateral tetrahedron. We have chosen to use the following quality metric [LDGC99]:

$$\text{Quality} = \text{sign}(\text{Volume}) \cdot \frac{3^7 \ \text{Volume}^4}{(\sum_{i=1}^{4} \text{Area}_i^2)^3} \tag{2.1}$$

where 3^7 is a normalization factor and the sum in the denominator is over the squared areas of each face of the tetrahedron and the sign of the volume is the sign of the determinant of the scalar triple product used to calculate the volume of the tetrahedron. This measure is one of several that becomes small for all five types of poorly-shaped tetrahedra, and becomes 1 for an equilateral tetrahedron. The measure also has the property that it becomes negative if a tetrahedron inverts.

For a given PDE on a particular domain, it is generally desirable to find a solution using as few computations as possible. It is therefore desirable to create tetrahedral meshes that lead to as few computations as possible when solving a PDE. When solving a finite-element based PDE in 3 spatial dimensions, each edge in the mesh

will result in two 3x3 non-zero sub-matrices in the solution matrix. The sub-matrices relate the displacements of the two nodes in the mesh. As we will be seeking fast algorithms, we would like to design meshes with as few edges as possible for a given node density. Similarly, for a given number of nodes, we often would like to minimize the number of tetrahedra in the mesh. When forming the solution matrix, calculations must often be done for every tetrahedron.

However, the number of non-zero entries in a matrix and the size of the matrix are not the only properties that determine the speed of solving a system of equations. Poorly shaped tetrahedra result in poorly conditioned matrices that can make iterative matrix solvers noticeably slower than they would be for well-shaped tetrahedra [BPar]. It is typically worthwhile to add additional tetrahedra and additional nodes in order to have well shaped tetrahedra.

In summary, we seek to create a mesh that (1) completely fills the region of interest, (2) has no T-vertices, (3) contains high quality tetrahedra, and (4) leads to as few computations as possible in a finite element simulation.

2.3 Prior Work

There are several publicly available, open-source tetrahedral meshers [Gei93, Owe98, SFdC+95], most of which are specialized for particular types of problems. Medical shape data has two important properties. First, surfaces are typically derived by running marching cubes [LC87] on a segmented volume and then smoothing and decimating the resulting surfaces [GNK+01, SMLT98]. The resulting nodes are not particularly important; that is, any set of nodes that describes the surface of the object is sufficient. Therefore, it is not critical that nodes of the surface mesh are also nodes in the volumetric mesh. A second property of medical shapes is that their surfaces can be complicated and non-convex. This causes problems for some less-sophisticated Delaunay meshers that are unable to handle this property.

Ferrant [Fer97] found that the meshers he tested on medical data were unable to produce desirable meshes. These meshers were unable to handle the non-convexity of

the medical data, produced many poorly-shaped tetrahedra, or produced many very small tetrahedra. He therefore wrote his own mesher by cutting tetrahedra using a marching tetrahedron algorithm.

We tested Ferrant's mesher on several anatomical structures and found the mesher lacking in some respects. In earlier versions of Ferrant's mesher, many of the tetrahedra were strongly skewed. In later versions, Ferrant *et al.* [Fer97] introduced a Laplacian smoothing term, which greatly improved the shape of the tetrahedra in the cases he tested. Unfortunately, Laplacian smoothing can often worsen the quality of meshes [ABE97, Dji00, Owe98]. We therefore did not further pursue the Ferrant mesher. Instead, we pursue methods to create our own mesher.

2.4 Mesh Formation

The goal of the tetrahedra mesher is to fill space with tetrahedra, conforming to a surface while creating as few skewed tetrahedra as possible (see Figure 2-3). There are three general methods for forming tetrahedral meshes [Owe98]: Delaunay tetrahedralization of pre-selected points, advancing front based methods, and oct-tree based methods. Each method has advantages and disadvantages.

Delaunay triangulation attempts to find the tetrahedralization of a given set of points such that the sphere through the four points of each tetrahedra (the circumsphere) contains no other points inside it. Generally, the resulting meshes are of high quality, though they often contain sliver-shaped tetrahedra [BPar]. The major difficulty for such methods is choosing the location of the points to be meshed; the decision process can be complicated.

Advancing front methods are very similar to Delaunay methods. In Delaunay methods, all of the nodes are placed first and then meshed. In advancing front methods, new nodes are added to the mesh one at a time by placing them outside the boundary of an existing mesh and adding tetrahedra. These methods face the same challenges of Delaunay methods in that deciding where to place points can be difficult.

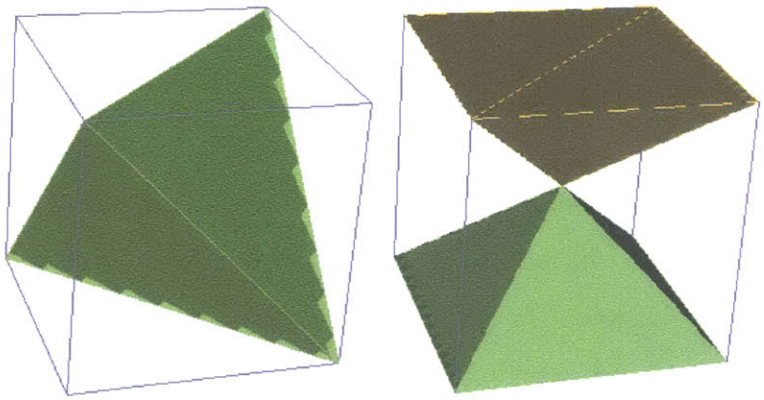


Figure 2-4: Two ways to subdivide a cube into tetrahedra. On the left, a cube is subdivided into five tetrahedra, of which only the center is shown. On the right a cube is subdivided into twelve tetrahedra by adding a point in the center. Only four tetrahedra are shown, the remaining eight can be created by adding an edge on each face.

Oct-tree based methods involve dividing space into cubes and sub-dividing the cubes into tetrahedra. Using these methods, it is straightforward to form a mesh. The challenge lies in making the mesh conform to the structure of interest. We pursue oct-tree methods because they are simplest to implement. Also, oct-tree methods will allow the user to specify at what resolution he wishes to represent a surface.

We build meshes by filling space with cubes and then sub-dividing the cubes into tetrahedra in a consistent way. We divide all cubes into five or all cubes into twelve tetrahedra as shown in Figure 2-4. For subdivision into five tetrahedra, an alternating pattern of the subdivision shown, and one rotated 90 degrees relative to an edge of the cube is necessary for a consistent mesh. For twelve tetrahedra per cube, no such tiling is necessary.

2.5 Multiple Resolution Meshes

It is often important to have meshes with a spatially varying density of nodes. For example, for meshes containing multiple shapes one might want one shape to contain more tetrahedra than another. For some shapes, one might want smaller tetrahedra near the surface and larger tetrahedra in the interior. Or, one may wish more tetrahedra in regions of high curvature of a surface. In this section, we briefly describe

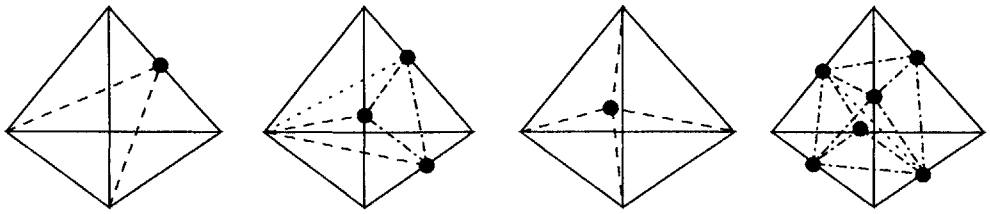


Figure 2-5: Common methods of subdividing a tetrahedron. From left to right: edge bisection which splits a tetrahedron in 2; face trisection which splits a tetrahedron in 4; adding a point in the middle, which splits a tetrahedron in 4; and bisecting all the edges which splits the tetrahedron in 8. Note that only the third method does not affect the neighboring tetrahedra.

methods for making meshes with different sized tetrahedra. A review of such methods is found in [SFdC+95].

There are several ways of forming multi-resolution tetrahedral meshes within oct-tree mesh generation. The first involves subdividing cubes before converting them to tetrahedra [SFdC+95]. The challenge in this method is making sure that after dividing the different size cubes into tetrahedra, the resulting tetrahedra at the surfaces of the cubes share faces rather than crossing in T-junctions. This result can be obtained using complicated case-tables or using local Delaunay triangulation. We do not pursue these methods, with one exception. It is straightforward to divide some cubes into five and some cubes into twelve tetrahedra. This ability allows us to generate meshes with roughly a factor of 2 variation in node and tetrahedra density.

A second method to subdivide meshes is to place nodes and then locally re-mesh. Usually, remeshing is done using Delaunay methods. A third technique to making multi-resolution meshes is to modify an existing mesh by subdividing the mesh. These methods include: edge bisection, face subdivision, placing a point in the middle of a tetrahedra, and subdividing all the edges of a tetrahedron (Figure 2-5). The main challenge with using these methods is that repeated application of a method can lead to arbitrarily low quality tetrahedra [AMP00]. Or conversely, an algorithm can over-refine a region in order to guarantee good-quality tetrahedra [AMP00, SFdC+95].

We have implemented three simple but effective methods to subdivide tetrahedra (Figure 2-5). The first two are bisection and edge collapse. To subdivide via the bisection method, we serially bisect the longest edge in the region subject to the

constraint that the resulting tetrahedra are of quality above a threshold. Bisecting the longest edge of a tetrahedron does not guarantee good quality tetrahedra. But in practice, the method is useful. The reverse process is edge-collapse. As in bisection, we serially collapse the shortest edge in a region subject to the constraint that the resulting tetrahedra are of high quality. Edge collapse is further described in Section 2.7.2.

The final method is octasection, which subdivides all the edges of a tetrahedron. In this method, many tetrahedra are subdivided simultaneously to achieve large variations in tetrahedral density. The major disadvantage of this method is that neighboring tetrahedra of tetrahedra that have been subdivided are also subdivided. For repeated iterations of octasection, these neighboring tetrahedra can become very low quality. This problem has led to the "red", "green" tetrahedron subdivision method. Tetrahedra that are octasected are labeled red. Tetrahedra that are subdivided because only some of their edges were subdivided are labeled green. In subsequent iterations of this method, if a green tetrahedron needs to be further subdivided, the last subdivision of that tetrahedron is undone and replaced by octasection and then the new subdivision takes place. The red-green method is generally stable, though it leads to large increases in tetrahedral density.

2.6 Fitting a Mesh to a Surface

After forming an initial mesh, we adapt that mesh to an object by intersecting the edges of the mesh with the surface of the object. If the object is specified as labeled voxels, intersections are found by checking for edges in which the two nodes of the edge overlap different labels. If the surface is specified as a polygonal surface mesh, we place the surface mesh in a object oriented bounding box (OBB) tree [GLM96] and intersect the OBB tree with an oct-tree of the edges of the mesh.

Once intersections are found, one subdivides the tetrahedra, introducing new nodes at intersections. There are $2^6 = 64$ different ways the six edges of the tetrahedra can be intersected. Taking into account symmetries, there are 11 configurations

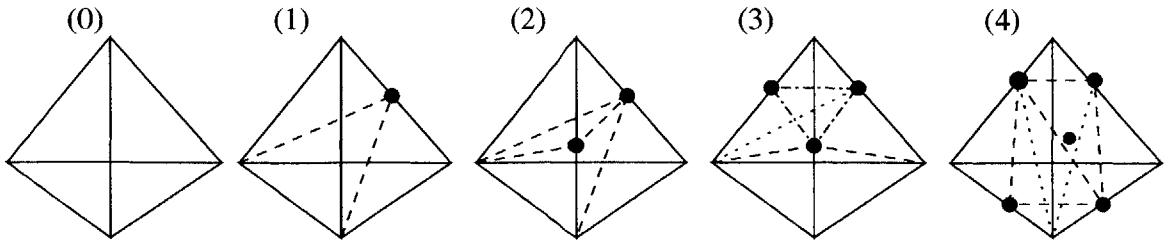


Figure 2-6: The most common ways that a tetrahedron is intersected. Each displayed point indicates an intersection. From left to right: (0) Nothing need be done for an un-cut tetrahedron. (1) A tetrahedron with 1 intersection is split into two tetrahedra. (2) A tetrahedron with 2 intersections is split into three tetrahedra. (3) A tetrahedron with 3 intersections is split into four tetrahedra. (4) A tetrahedron with 4 intersections is split into two prisms. A point is added in the center of each prism so that a total of 24 tetrahedra are created. It is straight forward to see that if the intersection points are near the vertices of the initial tetrahedron, the tetrahedron resulting from subdivision will be skewed.

of intersections [RM98]. We show the 5 most common configurations in Figure 2-6. Splitting a tetrahedron with one, two or three intersections is straightforward; though, one must be careful to do it in a consistent manner so as not to introduce T-vertices. For example, in the case of two intersections, Figure 2-6 shows that on one face, the two intersections and two corners of the tetrahedra form a quadrilateral, to which a diagonal is added. Similar faces exist in the three and four intersection case. We introduce the shorter diagonal to the quadrilateral as it produces better quality tetrahedra and is suggested by the Delaunay criterion.

2.7 Mesh Improvement

After a mesh is adapted to a surface, some fraction of the tetrahedra on the boundary are typically skewed. There are three methods generally used to improve the quality of the tetrahedra in a mesh [ABE97, FOG96]: inserting or removing points, local re-meshing, and mesh smoothing to relocate grid points.

To proceed, it will be necessary to have a metric of the quality, or lack of skewness

of a tetrahedron. We use the quality metric of a tetrahedron presented in Section 2.2:

$$\text{Quality} = \text{sign}(\text{Volume}) \cdot \frac{3^7 \ \text{Volume}^4}{(\sum_{i=1}^{4} \text{Area}_i^2)^3}, \tag{2.2}$$

where the sum in the denominator is over the squared areas of each face of the tetrahedron. The metric is 0 for a degenerate tetrahedron and 1 for an equilateral tetrahedron.

2.7.1 Mesh Smoothing

The simplest mesh improvement technique is Laplacian smoothing. With this method, each node point is moved towards the center of all its neighbors. Laplacian smoothing is local; each node is only affected by its neighbors. Laplacian smoothing also requires few computations. Alas, this algorithm can actually decrease the quality of the tetrahedral mesh [FJP95, FOG96]. Looking at Figure 2-7, one can understand some of the problems with Laplacian smoothing. The point in the figure is roughly at the center of its neighbors, where Laplacian smoothing would move it. Unfortunately, the green tetrahedron is nearly flat in the resulting configuration; its quality is therefore close to zero.

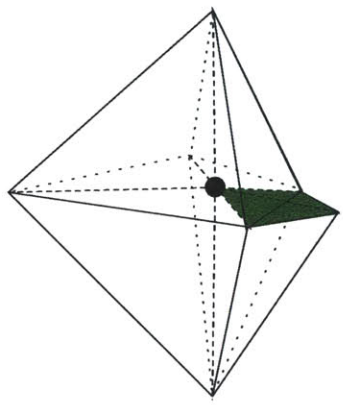


Figure 2-7: A point to be smoothed and all the tetrahedra which have it as a vertex. The green tetrahedron has much lower quality than the remaining tetrahedra; all of its nodes lie in a plane (it is a sliver). Laplacian smoothing would leave the point roughly where it is. However, a smoothing algorithm's goal should be to increase the quality of the green tetrahedron to become similar to the rest.

To improve the smoothing algorithm, while still keeping it a local method, one considers the quality of the tetrahedra. At each point p, one would like to move the point to increase the quality of the tetrahedra containing p. Specifically, it is the minimum quality one would like to increase. One possible function to maximize is:

$$\hat{\vec{x}}_p = \arg\max_{\vec{x}_p} \Big[\min_{t \in \mathrm{Tet}_p} \mathrm{Quality(t)} \Big] \tag{2.3}$$

where $\vec{x}_p$ is the position of point p,and Tet_p are the tetrahedra with p as a vertex. Thus, an alternative mesh improvement algorithm moves each node to the location that maximizes the minimum quality of the surrounding tetrahedra. This method was proposed by Freitag and Olliver-Gooch, and more generally by Amenta *et al.* [ABE97]; we call it the "*local optimum search* method" because it searches for an optimum to Equation 2.3. Like Laplacian smoothing, the method is local; each node is moved based only on its neighbors. Unlike Laplacian smoothing the *local optimum search* algorithm has the property that each node movement strictly improves the minimum quality of the tetrahedra that possess that node as a vertex. This property is very useful because it guarantees the minimum quality of tetrahedra in the mesh can never decrease.

We have found two difficulties with the *local optimum search* method. First, the qualities of tetrahedra must be evaluated many times during an iteration, so that each iteration is computationally expensive. More importantly, we find that the method converges to a locally optimal mesh that is often of much lower quality than we have achieved using other methods. In particular, after two or three iterations, the *local optimum search* method has often mostly converged. Because the algorithm is local, the small number of iterations only allows interactions between nodes that are close to each other in the mesh; neighborhoods of tetrahedra that are far away do not interact. For a smoothing algorithm to work well, one would expect that it should possess the ability to allow long range iterations.

Based on this expectation, we hypothesize a "noisy" smoothing algorithm might perform better than the *local optimum search* method. Such a smoothing algorithm

would explore more of the search space of possible node positions. We therefore designed a new method that moves a node in a direction to increase the quality of the lowest quality surrounding tetrahedron, rather than to find the maximum-minimum quality as suggested by Equation 2.3. By making approximations to the direction and how far a node is moved, we expect to explore more of the search space of node positions before convergence. This new technique is described in Section 2.8.2.

2.7.2 Edge Collapse

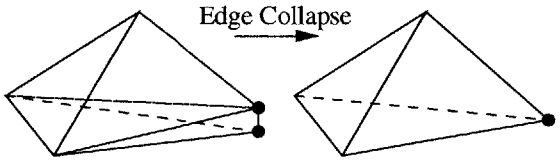


Figure 2-8: Example of edge collapse. The points, indicated with dots, are collapsed into one point causing a tetrahedron to be deleted. In this case, a "wedge" is deleted.

Edge Collapse is another important algorithm which we pursue. As shown in Figure 2-8, it is often possible to delete poor quality tetrahedra simply by collapsing them out of the mesh. Edge collapse has the effect of removing one node from the mesh. We consider collapsing edges of the lowest quality tetrahedra after smoothing. We find the lowest quality tetrahedra, and decide if collapsing an edge will decrease or increase the minimum quality of the tetrahedron affected. Edge collapse is known to sometimes cause surrounding tetrahedra to invert; however, this situation is prevented by our quality criterion.

2.7.3 Edge Flipping

Rather than removing points, one can attempt to change the connectivity of a region. Several researchers have tried Delaunay re-triangulation of a small region [KO98]. Others have tried local face swaps or edge swaps [ES96]. We implemented edge swapping for two simple cases: an edge owned by 4 tetrahedra and an edge owned by 3 tetrahedra, as shown in Figure 2-9. Each swap is performed only if the swap

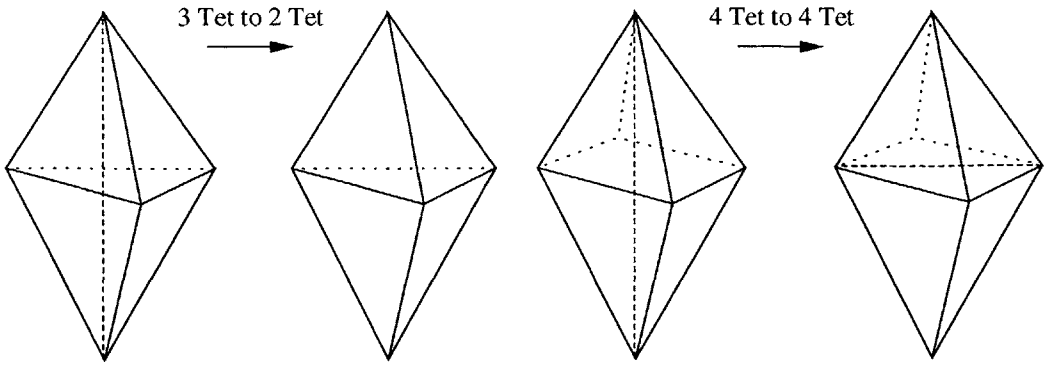


Figure 2-9: Two examples of swappable configurations. On the left three tetrahedra are transformed to two tetrahedra. On the right, the inner edge of four tetrahedra is moved to create four different tetrahedra.

increases the quality of the resulting tetrahedra. In practice, we use edge-flipping techniques only on the lowest quality tetrahedra.

2.8 Methods

We summarize the two algorithms we present in this chapter, and introduce a new mesh smoothing algorithm. The first algorithm is a single resolution method. For the first algorithm, we begin by filling space with cubes of a size set by the user. The edge lengths of the cube set the scale of the edge lengths that will be used in the mesh, including on the surface of the object to be meshed. The intersections of the mesh with a surface, or the boundaries of a labelmap are then determined. Often intersections occur very near one corner of a tetrahedron so that when subdividing tetrahedra (Figure 2-6) to conform to the edge intersections, the resulting tetrahedra are typically tiny and skewed. Instead, when possible, we move nodes of the mesh to the cutting surface. We have found that using this method reduces the number of tetrahedra that need to be subdivided by 60 to 90%, which substantially increases the average quality of the cut tetrahedra. After the intersections are found, we proceed to improve the quality of the tetrahedra.

2.8.1 Mesh Improvement

We create a new smoothing algorithm to improve the quality of the tetrahedra in the mesh after intersection with a surface. To motivate the smoothing algorithm, we define the following energy, E, on the mesh:

$$E = -\sum_{p \in \text{vertices}} \min_{t \in \text{Tet}_p} \text{Quality(t)} \tag{2.4}$$

where $\text{Quality}(t_{qmin})$ is the quality of the lowest quality tetrahedron that contains vertex p. We minimize this energy using a modified Newton iteration, moving one vertex in each iteration:

$$\vec{\delta}_{x_p} = \alpha * L^2 * \frac{d\left[\min_{t \in \text{Tet}_p} \text{Quality(t)}\right]}{d\vec{x}_p} \tag{2.5}$$

where α is a parameter in the smoothing algorithm between 0 and 1, L is the average length of the lowest quality tetrahedron, and the derivative is evaluated numerically. One can view this iteration as a modified Newton solver where the inverse of the Hessian Matrix is approximated by αL^2. Or, one can also view the iteration as gradient descent, where the step size is proportional to a local squared length; squared length is appropriate so that the units on both sides of the equation are the same.

Like the *local optimum search* method, Equation 2.5 is a local iteration. Unlike that method, the qualities of the tetrahedra that possess a node need only be calculated once per iteration. Also, the new method does not guarantee that the minimum quality of tetrahedra in the mesh can never decrease. In fact, the new iteration is subject to over-shooting. When the derivative term is sufficiently high, the movement in the point can be arbitrarily large. In order to avoid over-shooting, we use the following update rule:

$$\vec{x}_p = \vec{x}_p + \vec{\delta}_{x_p} \frac{\beta L}{||\vec{\delta}_{x_p}|| + \beta L}. \tag{2.6}$$

The extra multiplier has the effect of preventing the node from moving more than

βL, where β is a number between 0 and 1. We call this iteration "*local gradient-based smoothing* ". We have chosen this name because Equation 2.4 is a local method that moves nodes more in regions where the gradient of the minimum quality tetrahedron is largest. Such an algorithm roughly fits into the framework of force directed methods present by Djidjev [Dji00]. In these methods, nodes with large forces applied to them are moved further in each iteration than nodes with small forces applied to them.

Convergence of the *local gradient-based smoothing* algorithm can be determined by monitoring the average quality of the lowest quality tetrahedra in the mesh. Because the algorithm is somewhat of a noisy search, examining only the lowest quality tetrahedron in the mesh can be a noisy measure. A more robust measure is the average of the smallest few tetrahedra. We have chosen to use the average of the five lowest tetrahedra. When that average ceases to increase, the method is converged.

Our overall mesh improvement method consists of smoothing using the *local gradient-based smoothing* technique. We then use edge collapse methods, followed by edge swapping methods. As the great majority of tetrahedra have high enough quality so that the gains using edge collapse and edge swapping are minimal, we use these last methods only on tetrahedra whose quality are below a quality of interest. We have found a quality threshold of 0.1 to be far more than sufficient for our simulations. This quality generally corresponds to a tetrahedron that has a smallest edge that is approximately one forth the size of its largest edge.

Colleagues with experience solving PDEs using tetrahedral meshes have recommended a minimum tetrahedron quality; simulations with tetrahedra below this value tend to lead to inaccurate results and collapsed tetrahedra. A quality (using Equation 2.1) of about 0.015 is the suggested threshold[1], though qualities far higher than this are preferred.

[1]Ashraf Mohammed at Johns Hopkins University doing simulations with ABAQUS originally suggested this number.

2.8.2 Compact Representations

Our second algorithm is identical to the first except that it uses multi-resolution methods to create a compact representation of a volume. The goal of this algorithm is to create a mesh with a sparsely represented interior while maintaining the quality of the mesh. We obtain this result using edge collapse methods. A review of the literature finds several allusions to edge collapse as a method to reduce the number of tetrahedra in a mesh, but we have not found any actual implementations. We therefore describe one here.

To edge collapse a list of tetrahedra, one collapses the smallest tetrahedra first. However, collapsing tetrahedra stretches neighboring tetrahedra. We call the minimum quality of the "stretched" tetrahedra affected by an edge collapse, *collapse quality.* It is desirable to collapse those tetrahedra that lead to the largest *collapse quality* first.

We therefore maintain a list of candidate tetrahedra to be collapsed primarily sorted by volume. However, if the volumes of two tetrahedra are within 10% of each other, then we sort by *collapse quality.* Thus, when the algorithm attempts to edge collapse a list, it collapses the smallest tetrahedra that are expected to lead toward high quality meshes first. To avoid unnecessary calculations, at the start of the algorithm all tetrahedra are set to have a *collapse quality* of 1, which indicates *collapse quality* is unknown. (One is an upper bound on the *collapse quality.*)

The tetrahedron at the top of the list (lowest volume, highest *collapse quality*) is removed from the top of the list. If the shortest edge of that tetrahedron can be collapsed with a *collapse quality* greater than the target threshold, the tetrahedron is collapsed. If not, the *collapse quality* is cached for that tetrahedron and more tetrahedra are successively removed from the list until a tetrahedron can be collapsed. Once a tetrahedron is collapsed, the cached *collapse quality* of each tetrahedron that was stretched is no longer accurate; it is reset to 1. All tetrahedra that were not collapsed out of the mesh are placed back on the list. As these tetrahedra have been tested for edge collapse already and edge collapse produced a low *collapse quality,*

these tetrahedra are generally inserted into the the middle of the list, rather than the top. Thus, the next tetrahedra that the algorithm will attempt to collapse will generally not be from those that were just tried.

When collapsing an edge, two nodes in the mesh become one. It is possible to place the resulting node at any location. In practice, we find that it is effective to place the resulting node at either end point, so that no other candidate locations need be tried. However, after many iterations, edge collapse leads to skewed tetrahedra so that another edge collapse is not possible without a *collapse quality* less than the threshold. A simple way to fix this problem is to smooth the mesh. To prevent this problem from happening, after every 100 collapses, we smooth the remaining tetrahedra for two iterations. A small amount of smoothing has shown to be effective at solving this problem.

The algorithm terminates when no tetrahedra can be collapsed and when smoothing is no longer helpful toward this end. We summarize the edge collapse algorithm as follows

```
done = False
While (Not done)
 {
  I) Form the list of candidate tetrahedra to be collapsed
      1) Set the collapse quality of all tetrahedra to 1
      2) Sort tetrahedra by volume, smallest first.
         If the volume of two tetrahedra are within 10%
         of each other, secondarily sort by collapse quality,
         largest first.
  II) While ((the list is not empty) and (number of collapses less than 100))
      1) Get tetrahedron.
      2) If collapse quality is greater than threshold,
         try to collapse smallest edge
      3) If successful
         i) Reset collapse quality of all tetrahedra effected.
         ii) Put all tetrahedra back on main list
      4) If not successful
         i) Cache collapse quality.
 III) Smooth tetrahedra for 2 iterations.
  IV) If no tetrahedra have been collapsed in step (II) on two
      successive iterations, done = True
 }
```

2.8.3 Summary

We present two algorithms in this chapter. In the first method:,

1. The user chooses a cube size.
2. Cubes are divided into 5 tetrahedra.
3. The tetrahedra are intersected with the mesh.
4. The resulting mesh is smoothed using the *local gradient-based smoothing* method.
5. Edge collapse is performed on the lowest quality tetrahedra
6. Edge flipping is used to improve the quality of the lowest quality tetrahedra.

The second method is identical to the first, except that after the formation of the mesh, edge collapse is used to remove tetrahedra from the mesh.

2.9 Results

We first consider whether the single resolution method presented here can sufficiently well describe the surfaces of medical data. In each of Figure 2-10, Figure 2-11 and Figure 2-12, space was filled with cubes at a set resolution, subdivided into tetrahedra, intersected with a polygonal surface and then smoothed. Figure 2-10 shows the surface of tetrahedral meshes of a sphere at two different resolutions. Figure 2-11 shows the surface of meshes of an amygdala-hippocampus complex at two different resolutions. Figure 2-12 shows a mesh of Heschyl's gyrus. The complex was chosen because it is a medical shape of average difficulty to mesh, in that it is not convex, and has only a few regions of large curvature. Heschyl's gyrus, conversely, is an example of a difficult to mesh medical shape because it is very thin. The sphere is an example of a shape that should be relatively easy to mesh due to its uniform curvature.

The tetrahedral mesh volumes overlap the surface volumes to better than 95% at the lower resolutions, and 99% at the higher resolutions. The medical surfaces were produced using marching cubes [LC87] on a segmented volume, followed by a smoothing and triangle decimation algorithm. The volume changes due to the smoothing and decimation algorithm were significantly larger than those due to meshing.

One issue with this method is that it uses the same density of tetrahedra everywhere, rather than adapting to the curvature of the surface mesh. The central image

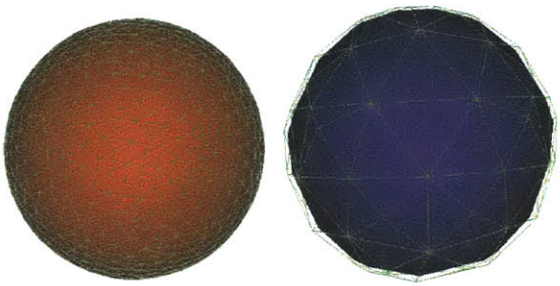


Figure 2-10: The surfaces of a low resolution and a high resolution tetrahedral mesh of a sphere.

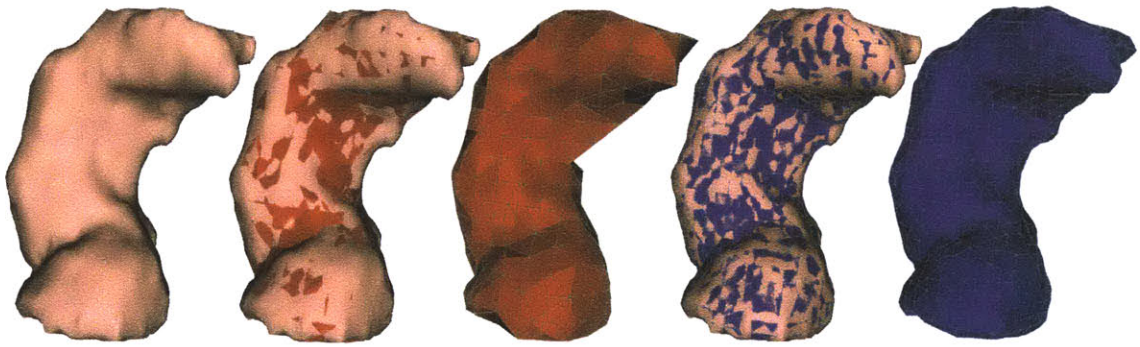


Figure 2-11: From left to right: the initial shaded surface mesh of an amygdala-hippocampus complex, the surface of a coarse mesh overlaid with the original surface, the edges of the surface of the mesh, a fine mesh overlaid the original surface, the fine mesh alone.

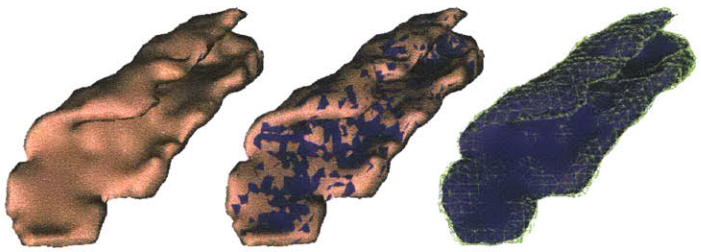


Figure 2-12: From Left to Right: the initial surface mesh of Heschyl's gyrus from a healthy patient; the surface of a tetrahedral mesh overlaid on the original surface; the edges of the tetrahedral mesh overlaid on a smoothed version of the original surface. In order to make the edges visible in the right most image, the surface was smoothed.

in Figure 2-11 illustrates this issue. There is a sharp corner in that mesh which does not appear in the original surface. One of the dangers of choosing a mesh to be represented at a particular scale is that features of a smaller scale cannot be accurately represented. This difficulty can be handled by selectively further subdividing tetrahedra based on curvature.

A second issue with this meshing method is shown well in the high-resolution sphere. There are numerous skewed triangles on the surface of that sphere. Those triangles are faces of tetrahedra; they therefore guarantee that the connecting tetrahedra will be skewed to some degree. Figure 2-13 shows quality profiles of three tetrahedral meshes. There are typically a small number of tetrahedra having a quality of approximately 0.1. This quality corresponds to a tetrahedron that has a smallest edge that is approximately one fourth the size of its largest edge. Dihedral angles in the meshes ranged from 40 degrees to 160 degrees, with a maximum ratio of lengths of 4.4. The dihedral angle and length ratios reported here are as good or better than dihedral angles and length ratios recently reported for another mesher designed for deformable bodies [MBF03].

2.9.1 Smoothing

Figure 2-14 illustrates the poor performance of the *local optimum search* method to improve the quality of the mesh. The algorithm effectively converges after two iterations. However, the quality of the resulting tetrahedra is below the desired threshold. Conversely, the *local gradient-based smoothing* method was able to significantly improve the quality of the tetrahedra, producing a mesh with qualities greater than the desired threshold, although, this new algorithm required many more iterations to converge.

Examining Figures 2-15 and 2-16, the *local optimum search* algorithm again converges after a very small number of iterations. The *local gradient-based smoothing* method required more iterations to converge. However, the quality of the resulting meshes is clearly significantly improved by using the *local gradient-based smoothing* algorithm over the *local optimum search* algorithm.

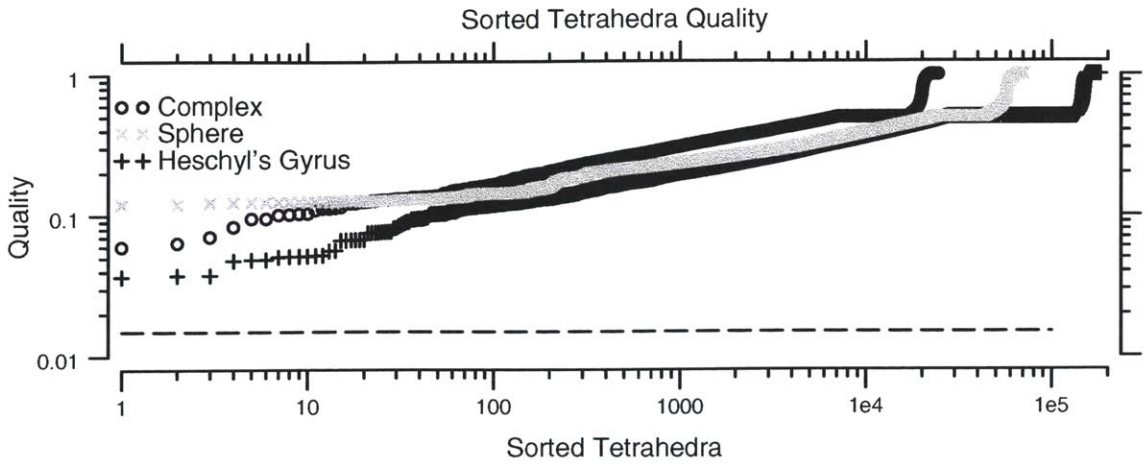


Figure 2-13: The quality of tetrahedra in three typical meshes: the amygdala-hippocampus complex, the densely represented sphere, and Heschyl's gyrus. To form these meshes, an initial mesh of cubes was subdivided into tetrahedra, cut with a surface and then acted on by mesh improvement techniques. All tetrahedra, both inside and outside the surfaces, are included in the above plot. The tetrahedra are sorted by quality. The dashed line indicates the minimum acceptable quality, 0.015.

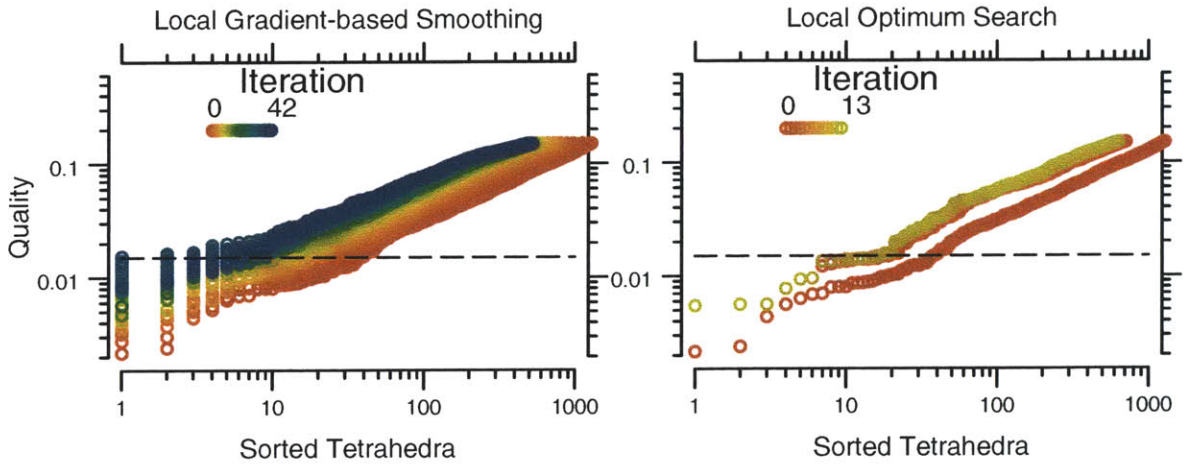


Figure 2-14: For a Heschyl's gyrus mesh, the lowest quality tetrahedra at each iteration of the gradient-based and optimal-searching algorithms. The smoothing algorithm required approximately 42 iterations to converge. The local optimal search algorithm effectively converged after 2 iterations; further iteration had almost no effect on the mesh.

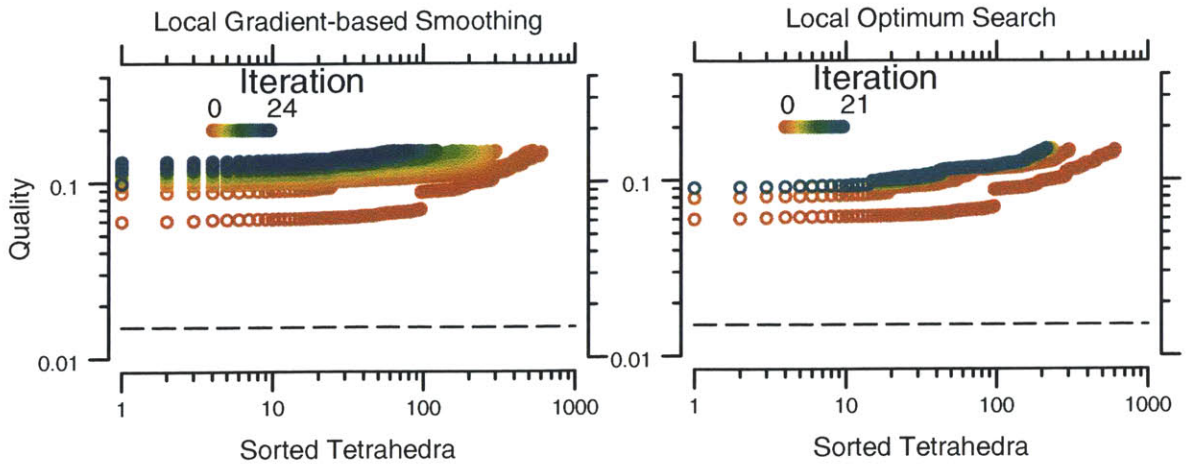


Figure 2-15: For a mesh of a sphere, the lowest quality tetrahedra at each iteration of the *local gradient-based smoothing* and *local optimum search* algorithms. The *local optimum search* algorithm effectively converged in 3 iterations. The *local gradient-based smoothing* algorithm required many more iterations, but yielded a higher quality mesh.

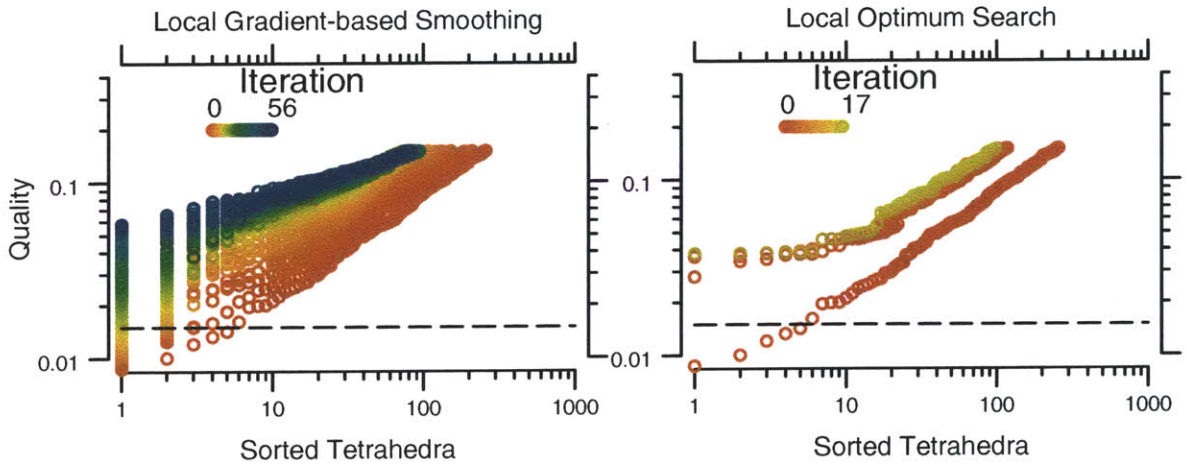


Figure 2-16: For a mesh of a amygdala-hippocampus complex, the lowest quality tetrahedra at each iteration of *local gradient-based smoothing* and *local optimum search* algorithms. The *local optimum search* algorithm effectively converged in 3 iterations. The *local gradient-based smoothing* algorithm required many more iterations, but yielded a higher quality mesh.

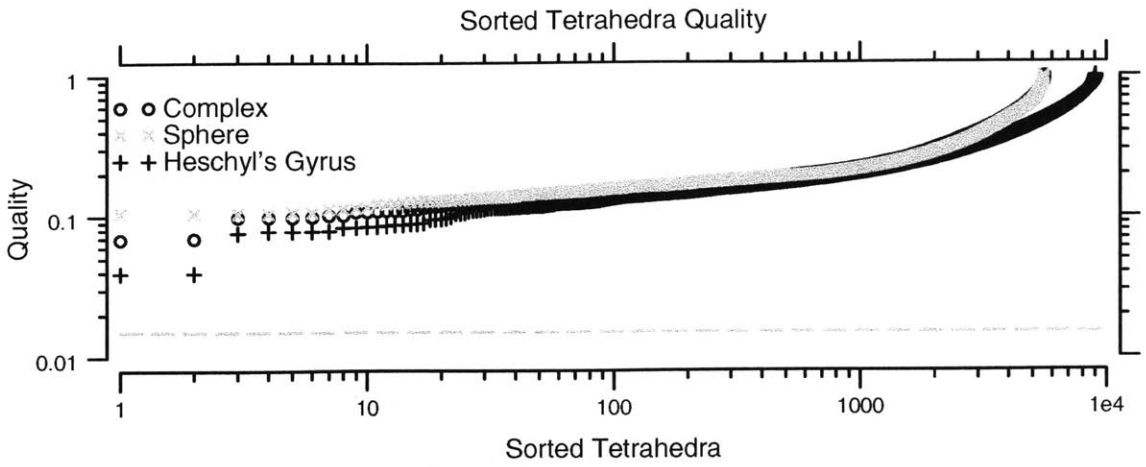


Figure 2-17: The quality of tetrahedra in three typical meshes from the previous section after edge collapse: the higher resolution amygdala-hippocampus complex, the higher resolution sphere, and Heschyl's gyrus. Unlike in Figure 2-13, which is the previous plot of tetrahedron qualities, only tetrahedra inside the meshed object are included in the above plot. The tetrahedra are sorted by quality. The dashed line indicates the minimum acceptable quality, 0.015.

There are two parameters for the *local gradient-based smoothing* algorithm, α and β. In practice, we find that numbers between 0 and 0.3 work well for both parameters. Setting the numbers too low leads to slow convergence. Setting the numbers too high leads to over-shooting, that is moving nodes too far. The results we show here use $\alpha = 0.2$ and $\beta = 0.1$, which have worked well for all the meshes we tested.

We attempted to smooth using various ordering of the nodes including the following four ways: moving all the nodes simultaneously, a different random order in each iteration, smoothing the nodes of the lowest quality tetrahedra first, ordering the nodes in a list and traversing the list in increasing and then decreasing order repeatedly. The local search method gave results almost independent of search order. In some cases, randomizing node order in every traversal through the nodes of the mesh somewhat helped the smoothing algorithm.

2.9.2 Compact Meshes

To make the meshes more compact, we use edge-collapse in the interiors of a mesh to remove as many nodes as possible while still maintaining a high quality. In Fig-

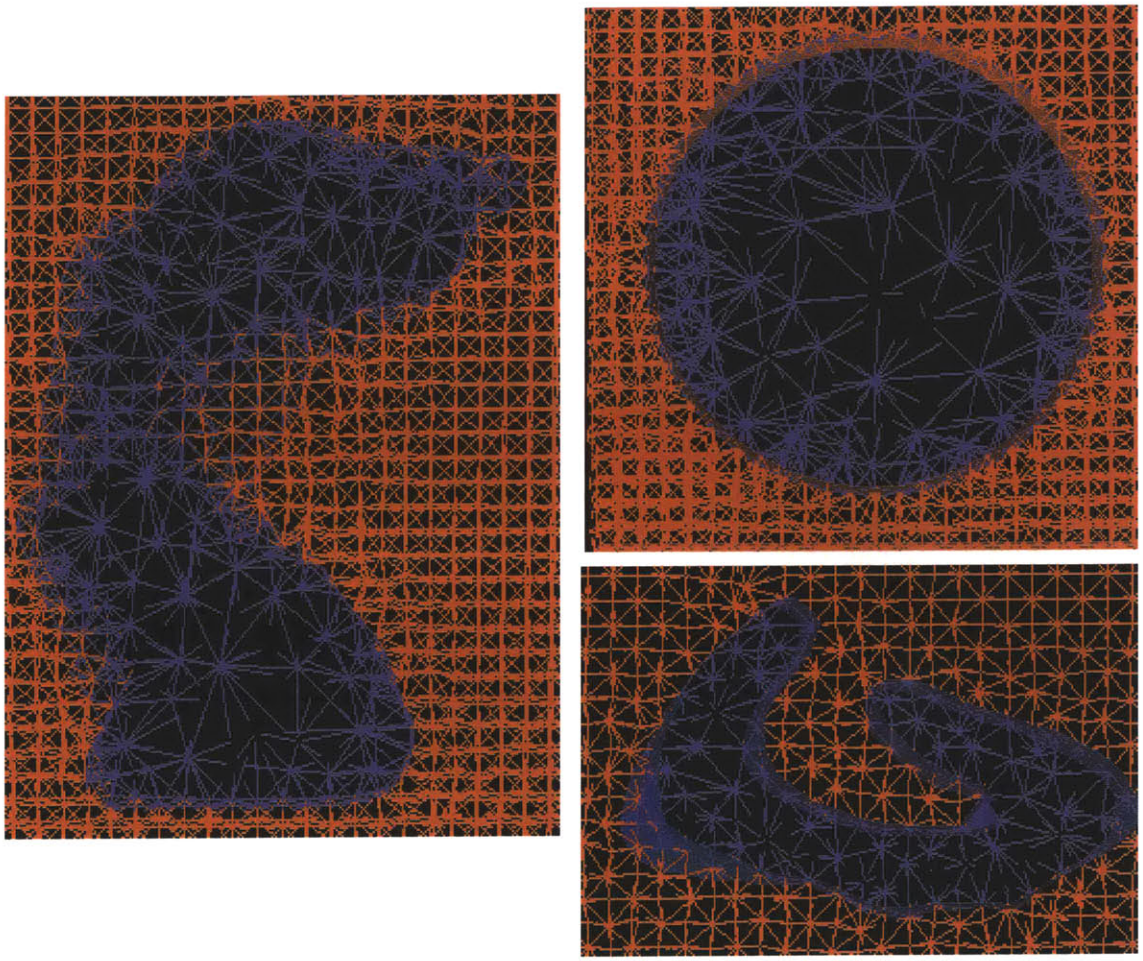


Figure 2-18: Thin slices through compact mesh representations. Only the interior (blue) of each mesh was edge collapsed so that the exterior (red) represents the original resolution. Note the large increase in node density near the surfaces

ure 2-17, we show the resulting quality profiles after collapsing the gyrus, the higher resolution complex and the higher resolution sphere from the previous section. The edge collapse algorithm guarantees no loss of quality below the quality threshold. As that threshold was set to 0.1, no tetrahedra were modified to a quality less than 0.1.

Table 2.1 shows the decrease in the number of nodes and the number of tetrahedra in each mesh. The most important result of the table is that for all three meshes, the resulting representation contained more than 75% of the nodes of the mesh in the surface. Thin slices through each mesh are shown in Figure 2-18.

	Results of Edge Collapse				
	Before Edge Collapse		After Edge Collapse		Both
Structure	Num Tet	Num Node	Num Tet	Num Node	Num Node on Surface
Sphere	21880	5033	5564	1690	1290
Complex	12719	3256	5612	1764	1360
Gyrus	14044	3903	9047	2800	2171

Table 2.1: The number of tetrahedra and the number of nodes in each mesh before and after edge collapse. Note that in all three cases, more than 75% of the nodes in the mesh lie on the surface of the mesh after edge collapse.

It is worth noting that most meshing algorithms choose octasection for creating sparse meshes. Unfortunately, octasection somewhat degrades tetrahedron quality. When the resulting skewed tetrahedra are intersected with a surface, the results are typically even more skewed. For example, using octasection to create a mesh of the sphere with a similar number of nodes to the one in Figure 2-17, produced qualities as low as 0.025 after mesh improvement. Most other researchers achieve much higher qualities using smoothing methods on the surface of the mesh. We have not implemented such methods.

2.9.3 Other Multi-resolution Technologies

In the previous section, we examined edge collapse. In this section, we briefly demonstrate other multi-resolution capabilities of the mesher. The simplest way to achieve a node density difference is to subdivide cubes into different numbers of tetrahedra. Figure 2-19 demonstrates subdividing cubes into 5 tetrahedra everywhere except near

the polygonal surface, which is subdivided into 12 tetrahedra, roughly a factor of 2 change in tetrahedron density and node density.

Figure 2-20 demonstrates edge bisection and edge octasection. The central mesh in that figure was generated by subdividing the edges of the left-most mesh in order to increase the density of tetrahedra by a factor of 4. The subdivision was done with essentially no loss of mesh quality. The right most image was generated using octasection on an entire region. In this case, the tetrahedral mesh was subdivided before intersection with the surface mesh.

2.10 Discussion and Conclusion

We created a tetrahedral mesher to represent the surfaces of medical objects with meshes of sufficient quality. We demonstrated this capability along with the ability to produce meshes of varying resolution.

We introduced a new mesh smoothing algorithm that produced meshes of significantly higher quality than the *local optimum search* method. In our experience, the *local gradient-based smoothing* method is effectively a noisy search. In regions with low quality tetrahedra, large quality gradients in the positions of nodes cause the smoothing method to yield relatively large node displacements – often somewhat over-shooting the location that would be determined by the *local optimum search* method. The effects of over-shooting propagate through a region, creating interactions between tetrahedra that are not direct neighbors. Conversely, the *local optimum search* algorithm converges to a locally optimum mesh in just a few iterations, preventing interactions from tetrahedra that are not direct neighbors.

The *local optimum search* algorithm has the advantage of guaranteeing that the minimum quality found in the mesh after searching is always equal to or greater than the minimum quality in the mesh before smoothing. The smoothing iteration we proposed has no such guarantee; it is possible to move points in such a way as to decrease the quality of the lowest quality tetrahedron in the mesh. The energy to be minimized is a global energy, and therefore it should decrease with each iteration.

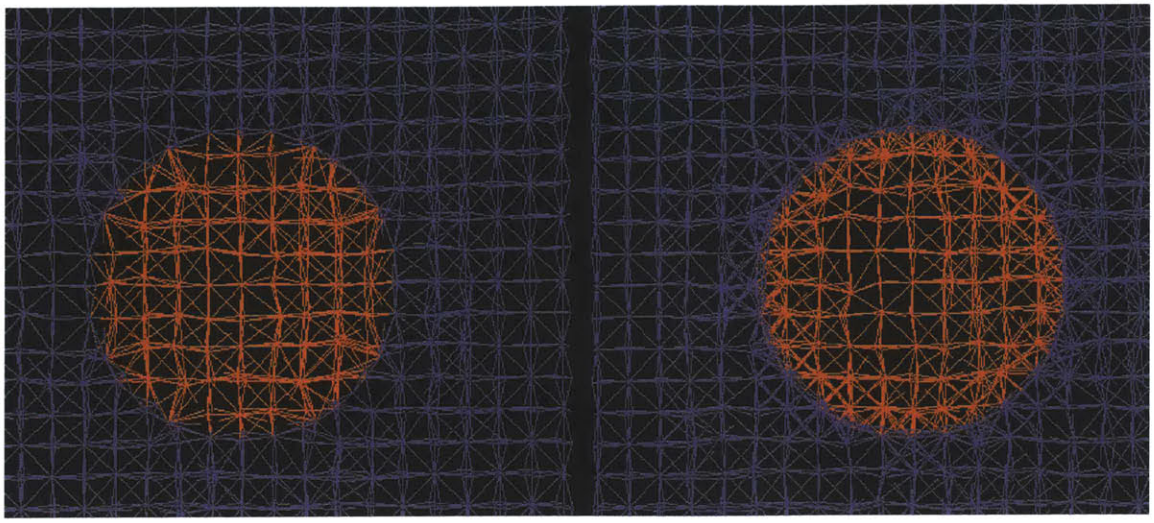


Figure 2-19: Two mesh of a sphere. The mesh on the left was generated from cubes subdividing into 5 tetrahedra. The mesh on the right was generated in the same way, except cubes intersecting the surface of the sphere are subdivided into 12 tetrahedra, roughly doubling the density of mesh nodes near and on the surface.

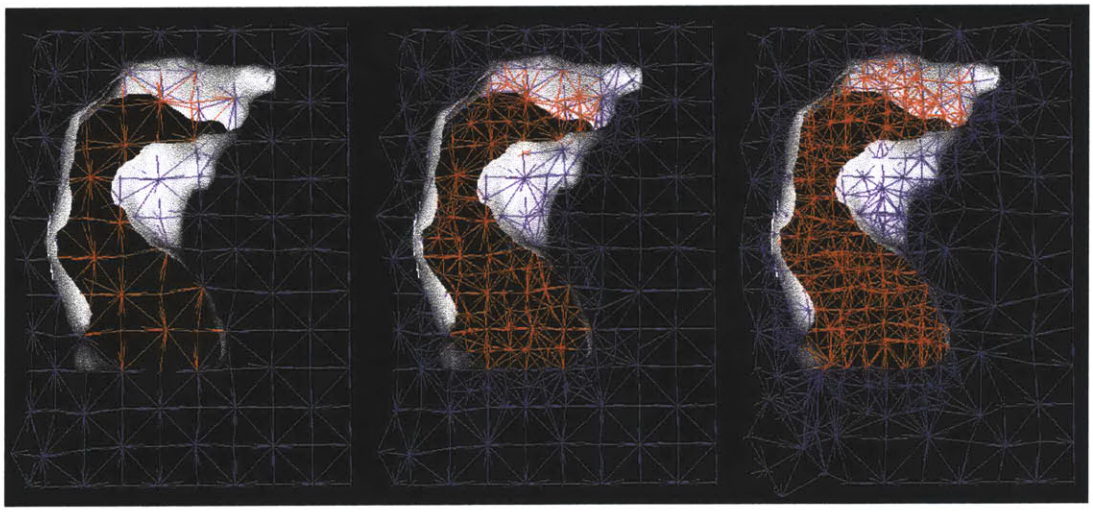


Figure 2-20: Meshes of an amygdala-hippocampus complex with varying density. From left to right: an initial mesh generated according to Section 2.9. The same mesh after edge-bisection of the complex and some neighboring tetrahedra in order to increase the number of tetrahedra by a factor of 4. The same initial mesh, subdivided using octasection before intersecting it with the surface

However, in practice, there are sufficient constraints on the motion of the nodes so that the quality of tetrahedra do not drop significantly between iterations.

We introduced an edge collapse algorithm to reduce the number of tetrahedra in the interior of meshes. The meshes that resulted from the algorithm had slightly more than 75% percent of their nodes on the surface of the mesh, and easily met our quality goals. Overall, the meshing algorithms met our goals, producing meshes above the quality threshold that we desired, and dihedral angles and ratios of the lengths of tetrahedra in a range similar to other meshers.

There are several areas of improvement for the meshing algorithms presented here. Clearly, the ability to smooth tetrahedra within the surface would be very helpful. In fact, we have recently implemented such a smoothing function, but we have not yet fully evaluated its performance.

The meshes created in this chapter will be used in the Chapter 3 and Chapter 5 for shape-matching and deformable models. The tools developed here will be essential for image matching in Chapter 6.

Chapter 3

Free Form Shape Matching

In the previous chapter, we developed compact tetrahedral representations to describe anatomical objects. In this chapter, those representations will facilitate the creation of efficient methods for the non-rigid registration of shapes.

The non-rigid matching of three dimensional shapes is a core technology for several medical image analysis applications. Such matching is used to find the correspondence between an anatomical atlas and a volumetric dataset [FCM+01, GRB93], allowing atlas information to be transferred to a patient. Non-rigid warping of shapes is also used between two labeled volumes to derive a deformation field that can be used to match corresponding gray scale images [DP94, FWN+00]. In addition, in order to correlate shape and disease, medical structures can be non-rigidly aligned enabling machine learning algorithms to classify them based on deformation fields. [DVR+96, MG98, TGG+02].

Unfortunately, finding correspondences between medical shapes can be difficult. Surface points on one shape can be matched to the surface of a different structure in an infinite number of ways. Choosing a meaningful match is difficult because medical objects typically have smooth surfaces. Often, it is not intuitive where points in one smooth surface should lie on a second surface.

Matching medical shapes is further complicated by errors in the segmentation process. For example, intensity based segmentations methods can result in disconnected regions of what should be a single object (see Chapter 5). The surfaces of

manual segmentations can suffer from features such as protrusions, inlets, and other sharp features that are likely mistakes by the person who created the segmentation [KSG+92]. These defects can appear prominently on one medical surface, but not at all on a different medical surface. An example is shown in Figure 3-1.

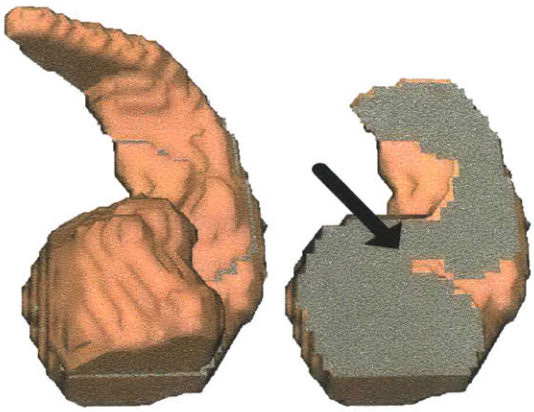


Figure 3-1: Left: Surface of voxel representation of a manually-segmented amygdala-hippocampus complex. The surface of the complex was smoothed after the manual segmenter finished identifying voxels. Right: a slice through the complex. The arrow points at an "inlet" in the segmentation. This feature does not appear in any other amygdala-hippocampus complex in the data sets examined. The inlet is likely an error due to a manual-segmenter.

One way to overcome these challenges is to find recognizable landmarks on surfaces to guide the match. Unfortunately, while many medical surfaces have recognizable undulations, they often do not have distinct points which can be accurately and robustly identified.

A separate way to non-rigidly register shapes is to minimize an energy while forming the match. This approach is known as free-form shape matching, and it is the approach we take. We model one of the shapes as a linear elastic object to be matched to the other shape. Intuitively, in a good match, high curvature regions in one object should match to high curvature regions in a second object. Matching by minimizing a linear elastic energy should accomplish this goal. That is, matching a sharp portion of one surface against a sharp portion of another surface is lower energy than flattening the region to match against a less sharp adjacent region.

We are not the first to pursue elastic models with matching [DP94, FWN+00,

WS98, GBBH95]. Elastic models have become a popular choice because they are easy to understand and because their properties resemble the mechanical properties of tissue (Real tissue properties are far more complicated. See [MC02] for example). Unfortunately, in the absence of landmarks, linear elastic matching leads to a set of ill-conditioned equations which makes the problem difficult to solve. However, while the equations may be poorly conditioned, the problem is not. We present a novel, linear-time matching algorithm that uses a linear elastic model and overcomes the associated ill-conditioned equations.

3.1 Related Work

Because of the importance of shape matching, there have been a number of different methods proposed for finding correspondences between shapes. Many of the methods use a deformable model that is warped on the basis of an image agreement term and a smoothness prior. The prior insures that the model deforms onto its target in a smooth way.

Christensen *et al.* [CRM96], for example, introduced models that deform using a viscous fluid prior. Because these models deform like fluids, they have the tremendous flexibility that could be necessary to match two arbitrary shapes. However, the flexibility of these models can be a drawback. In some applications these models may yield unphysical deformations spread over large regions [WS98].

Ferrant *et al.* implemented a shape matcher using an active surface algorithm [FWN+00]. The surface is treated like a membrane and is attracted to the closest point on the surface of the target shape. Once the surface correspondences between shapes are established, interior correspondences are formed by interpolating the surface correspondences using a linear elastic model. We find it somewhat un-natural to match a surface without using an elastic prior, and then to subsequently use elasticity in the interior, rather than simultaneously.

Davatzikos and Prince examined a method which is closer to a full elastic solver [DP94]. They establish correspondences between surfaces at a series of points. They

then use an elastic force and an image derived force to fill in the rest of the deformation field. Wang and Staib [WS98] found that constraining the deformation field at a series of points led to a "jiggling effect" near the surface. They therefore modified this method by not forcing the deformation field to be fixed at the surface points, but rather to penalize the deformation field for deviating from the set points.

Our methods are similar to those of Wang and Staib [WS98] in the use of an elastic energy smoothness prior with a statistical image agreement term. Wang and Staib note that with these two terms alone, the resulting equations are ill-conditioned and difficult to solve. They add landmark points to improve the conditioning of the equations to make them easier to solve.

However, it can be challenging to find landmark points on smooth medical surfaces. We would prefer to match shapes without the need to specify landmarks. In this chapter, we develop a method to solve the ill-conditioned equations that result from using an elastic energy smoothness prior with a statistical image agreement term.

3.2 Choice of Representation

Non-rigid shape matchers generally hold one shape fixed and deform the other shape onto it. The target shape is most often represented as a three dimensional image, with voxels inside the object labeled 1 and voxels outside the object labeled 0. In the medical image processing community, the two major choices of representations that are used for deforming shapes are deformable surface models and voxel based representations.

Deformable surface models, or snakes, are typically matched to a target surface by following gradients (boundaries) in the target image. In this process, one of the nodes of the surface mesh overlays a gradient in the target image, moves onto the maximum of the gradient, pulling its neighbor nodes into a capture range of the gradient. The capture range can be increased by smoothing the target image or by doing a local search around each node.

Deformable surface models have the disadvantage that image forces can only be

applied to the small part of the surface near image gradients. Ferrant *et al.* [FWN+00] addresses this issue by using a surface mesh that is attracted to the closest point in the surface of the target image. To move a mesh node towards the closest target point, they calculate a distance map around the surface of the target image and evaluate the gradient of the distance map at each node in the surface mesh. By using distance maps, Ferrant can apply image forces to all nodes in the mesh. Unfortunately, while distance maps can be effective, they are computationally costly to compute relative to other shape matching methods.

A second way to create image forces in many parts of the deformable model is to use a volumetric deformable model using a voxel based representation such as in the work of Thirion [Thi98] or Christensen *et al.* [CRM96]. In this type of method, there is a volumetric agreement term between the deformable image and the target image. Forces can be applied anywhere in the deformable model where there are gradients in either the initial or target image. However, often forces are only applied based on gradients in the target image.

Figure 3-2 illustrates this method in two dimensions for a blue disk being attracted to a green disk. The disk is broken into small chunks that prefer to overlap the green disk rather than the background. Those chunks that partially overlap background and partially overlap foreground feel a force to be attracted to the foreground. In this way, the blue disk is attracted into the green disk.

We seek to form an efficient algorithm using a volumetric representation of a deformable model. As discussed in the previous chapter, voxel representations inefficiently describe the volume and represent the surfaces of medical objects as unrealistic jagged edges. Therefore, instead of using voxels to represent a deformable object, we use tetrahedral elements. The faces of the tetrahedra can be aligned to the surface of the object to be deformed. The interior of the objects can be filled with a small number of volume elements.

One might expect volumetric methods to be relatively slow compared to deformable surfaces because volumetric descriptions can be much larger than surface representations. For example, if a surface mesh of a sphere uses n^2 points, a volumet-

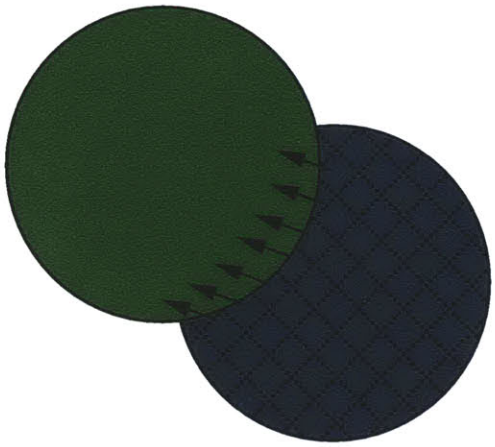


Figure 3-2: Illustration of a volumetric attraction in two dimensions for the case of a blue disk being attracted to a green disk. The blue circle is broken into small pieces which prefer overlapping the green circle to the background. Small pieces, that partially overlap background and partially overlay the green circle, feel a force to move them onto the green disk, indicated by the arrows.

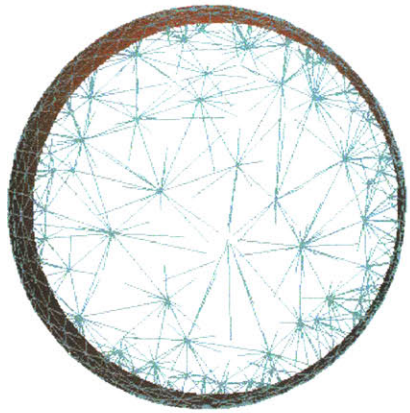


Figure 3-3: A thin slice through a tetrahedral mesh of a sphere illustrating that volumetric data can be compactly represented. The entire mesh contains 1690 nodes; the majority of nodes, 1290, are on the surface.

ric mesh might have on the order of n^3 points. In fact, the increase in the number of nodes can be much smaller than this simple three halves power law would predict because the interior of the deformable object can be represented sparsely. A thin slice of such a representation of a sphere is shown in the right of Figure 3-3. In that case, the surface of the tetrahedral mesh contains 1290 points, and the interior of the mesh contains an additional 400 points. For this case, the cost of switching from a surface mesh to a volumetric mesh is about 30% in the number of nodes.

3.3 Methods

The matching process starts with two labeled volumes, each with 1's inside the shape and 0's outside. One volume is filled with a mesh of tetrahedra. The mesh then aligns itself to the other label volume by minimizing a combination of an elastic energy term and an image agreement term.

3.3.1 Representation

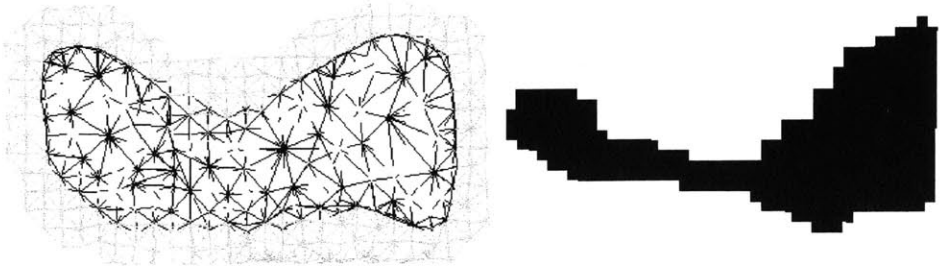


Figure 3-4: Representations used during shape matching. Left: thin slice through tetrahedral mesh. Tetrahedra inside the mesh are labeled 1, indicated by black. Tetrahedra outside the mesh are labeled 0, indicated by light gray. Right: slice through a target shape. The tetrahedra labeled 1 are attracted to the interior of the target. The tetrahedra labeled 0 are attracted to the background. The examples are sagittal slices through left amygdala-hippocampus complexes.

We begin by dividing a 3D shape into a mesh of tetrahedra using methods developed in Chapter 2. That is, a tetrahedral mesh at a chosen resolution is conformed to the surface of the shape; then as many tetrahedra are removed from the interior of the shape as possible. The resulting mesh includes tetrahedra that are inside the shape and outside the shape as shown in Figure 3-4. Each tetrahedron is labeled 1 if it is inside the initial shape or 0 if it is outside. During the matching process, the tetrahedra with label 1 will be attracted to the interior of the target shape. The tetrahedra outside the initial shape will be repelled by the target shape.

3.3.2 Matching Framework

We use a probabilistic framework to match two shapes. We choose such a framework in order to motivate a principled way to merge an image agreement term and a linear elastic energy term. Let U be the target image, V be the mesh, and $\vec{r}$ the displacement of the nodes of the mesh. We write the joint probability density of U, V and $\vec{r}$ as

$$p(U, V, \vec{r}) = p(U, V_{\vec{r}}|\vec{r})p(\vec{r}) \tag{3.1}$$

where $V_{\vec{r}}$ is the displaced mesh. We use probability densities rather than probabilities because $\vec{r}$ and interpolated intensities within U are continuous variables. One way to

write a maximum a posteriori estimator is to find $\hat{\vec{r}}$ that maximizes the logarithm of Equation 3.1,

$$\hat{\vec{r}} = \arg\max_{\vec{r}} \log p(U, V_{\vec{r}}|\vec{r}) + \log p(\vec{r}). \tag{3.2}$$

To estimate $p(\vec{r})$, note that $p(\vec{r})$ is the probability density that the mesh is in a certain configuration. In statistical physics, the probability density that a system is in a configuration is proportional to $e^{-E/\tau}$ where E is the energy of the system and τ is a temperature. We treat the mesh like a linear elastic material so that the energy of the mesh is the elastic bending energy. That energy is given by the integral of the stress strain product, $\sigma^T\epsilon$, over the mesh. Using standard finite element methods (FEM), the total elastic energy of the mesh is simply the sum of the elastic energies from each tetrahedron in the mesh. By assuming a linear displacement field within each tetrahedron, through standard means [ZT89], the integral can be linearized about a current configuration so that

$$E = \int_{\text{Volume}} \sigma^T\epsilon \, dV \approx \frac{1}{2}\vec{r}^T K \vec{r} \tag{3.3}$$

where K is an elasticity matrix. The matrix K is proportional to Young's modulus Y and also depends on Poisson's ratio ν – both of which have been assumed to be uniform in the mesh. The matrix K has the key property that it is sparse; most of its elements are 0, as shown in Figure 3-5. Note that the elements of the matrix K changes each time the location of the nodes is updated; though, the "sparseness structure" of the matrix does not change.

Using K, the probability density $p(\vec{r})$ becomes proportional to $e^{-\vec{r}^T K\vec{r}/\tau}$. Because K is proportional to Y, the exponential term has two unknown parameters, ν and $\frac{Y}{\tau}$, which will become parameters in the model. Note that large $\frac{Y}{\tau}$ corresponds to a stiffer deformable model; one which large image derived forces would be necessary to overcome.

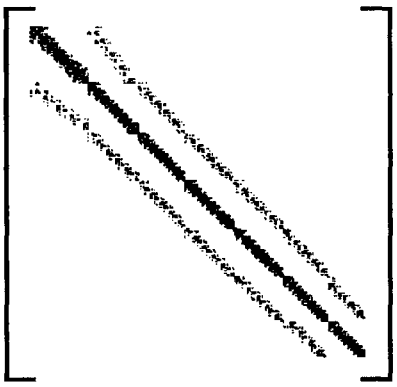


Figure 3-5: An image of the matrix K for a mesh of the amygdala-hippocampus complex. The image uses black points to indicate non-zero elements in K. Because of the connectivity of the mesh, most of the elements of K are zero.

3.3.3 Image Mesh Agreement Term

There are two principled ways to estimate $p(U, V_{\vec{r}}|\vec{r})$ which we investigated. The first approach initially seems intuitive, but in fact is not an appropriate approach. In the first method, we introduce a spatially stationary (or spatially invariant) probability density $p(u, v)$ where u and v are overlapping intensities in the image and mesh respectively. Using this method, the probability of the mesh overlapping the image in a given configuration is the integral of the probability density function over space. This method is generally not appropriate for our problem. In this method, we effectively divide space into small units and add the probability density in each unit. Probability densities add for events that are mutually exclusive; that is, for a set of events in which only one can happen at one time. As the mesh overlaps many different units of space simultaneously, this derivation does not well model the matching problem.

A more appropriate assumption is that the probability of overlap of each unit of volume is independent of the next unit. This model is too strong because we expect spatial correlations within the shapes we are matching. However, since the spatial correlation can effectively be modeled by the regularization of the mesh, the independence assumption may still be useful, especially in view of its simplicity.

We proceed with the standard assumption that the probability of overlap at each voxel is independent of every other voxel. Let u_T be the intensity of the voxels inside

tetrahedron T, and l_T be the label of a tetrahedron where both take on the values 0 or 1,

$$\log p(U, V_{\vec{r}}|\vec{r}) = \sum_T \sum_{u \in u_T} \log p(u, l_T|\vec{r}) \tag{3.4}$$

$$= \sum_T N_T \left(\frac{1}{N_T} \sum_{u \in u_T} \log p(u, l_T|\vec{r}) \right) \tag{3.5}$$

$$\approx \sum_T \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \langle \log p(u, l_T|\vec{r}) \rangle_T \tag{3.6}$$

where $\langle \cdot \rangle_T$ represents the continuous average over tetrahedron T, N_T is the number of voxels in T, $\mathcal{V}_{voxel}$ is the volume of a voxel and $\mathcal{V}_T$ is the volume of the tetrahedron. The progression from Equation 3.5 to Equation 3.6 is the change from the discrete, voxels, to continuous space, where u becomes a continuous variable and is determined through trilinear interpolation. To make that advance, we have effectively performed the thought experiment of upsampling the image many times. In this case, there are many voxels within each tetrahedron, so that continuous approximations become appropriate.

There are a number of probability densities, $p(u, l_T|\vec{r})$, that might be appropriate for a shape registration problem. As long as it is more probable that similar labels in the mesh and image overlap than different labels overlap, the probability density should be acceptable. In practice, the exact form of the probability density determines the strength of the image forces relative to the elastic forces. However, as the elastic forces have a free parameter $\frac{Y}{\tau}$ which determines their strength between the two forces, a careful choice of probability density should not be necessary. We choose the following exponential probability density:

$$P(u, l_T|\vec{r}) = e^{-|u - l_T|} * \frac{e}{e-1} \tag{3.7}$$

where the constant is a normalization factor. Using this density, tetrahedra with label 1 are attracted to intensities near 1 and tetrahedra with label 0 are attracted to intensities near 0. Substituting into the initial maximization and simplifying, the

objective function becomes

$$\hat{\vec{r}} = \arg\max_{\vec{r}} \frac{-\vec{r}^T K \vec{r}}{2\tau} + \sum_{T\in\{l_T=1\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \langle u|\vec{r}\rangle_T + \sum_{T\in\{l_T=0\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \langle -u|\vec{r}\rangle_T. \quad (3.8)$$

3.3.4 Solving Technique

One standard way to maximize a function is to use a Newton based solver to find the zeros in the gradient of the maximization function. A standard Newton solver would find that given the guess $\vec{r_k}$ in the k^{th} iteration, the change in the deformation field, $\vec{\delta r}$, in the $k+1^{th}$ iteration solves

$$\left[\frac{K}{\tau} - H_P\right] \vec{\delta r} = \frac{-K\vec{r_k}}{\tau} + \sum_{T\in\{l_T=1\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r_k}} \langle u|\vec{r_k}\rangle_T + \sum_{T\in\{l_T=0\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r_k}} \langle -u|\vec{r}\rangle_T \quad (3.9)$$

where H_P is the Hessian of the image agreement term, and the gradient and the Hessian terms can be estimated numerically. The terms on the right of the equation can be recognized as elastic forces due to the existing displacement of the mesh and image forces. The terms on the left can be recognized as the change in the elastic force and the change in the image force due to the candidate displacement.

Unfortunately, the standard Newton solver formulation has several difficulties in maximizing Equation 3.8. The first difficulties come from H_P. Because the Hessian term is estimated locally, it can have positive eigenvalues. These positive eigenvalues make the solver unstable. To understand this problem, consider a node between voxels in a region where locally the intensity quadratically increases. If that node is attracted to high intensities, the solution to the system of equations could be to move the node infinitely far in one direction. A second difficulty with the formulations is that the elasticity matrix K is also singular. For example a rigid translation of the entire mesh causes no elastic energy increase, and therefore a rigid translation is in the null space of the elasticity matrix.

To address these issues, we do not calculate H_P. (Note that changing the left hand side of the equation does not affect the maximum found. Convergence occurs when

the forces on the right hand side of the equation balance, so that $\vec{\delta r} = 0$.) Instead, we replace it by $-\epsilon I$ where ϵ is a small positive number and I is the identity matrix. This term essentially penalizes $\vec{\delta r}$. If one considers each iteration, a time step, then $\vec{\delta r}$ is a velocity, so that this term is effectively a velocity drag force. One can also think about this matrix as equivalent to adding a small constant to the diagonal of K. Adding a small constant to the diagonal of a matrix is a standard technique to improve the conditioning of the matrix. The iteration becomes

$$\left[\frac{K}{\tau} + \epsilon I\right] \vec{\delta r} = \frac{-K\vec{r_k}}{\tau} + \sum_{T \in \{l_T = 1\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r_k}} \langle u | \vec{r_k} \rangle_T + \sum_{T \in \{l_T = 0\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r_k}} \langle -u | \vec{r} \rangle_T. \tag{3.10}$$

While the ϵI term improves the conditioning of the elasticity matrix, the elasticity matrix is still somewhat poorly-conditioned. Solving a poorly-conditioned linear system is not a problem for standard Krylov subspace based iterative solvers. A solver of this sort attempts to minimize the squared error in the system of equations while minimizing the norm of the solution vector. However, because there may be no exact solution, an iterative solver will require many iterations before it converges. We therefore do not try and solve the system of equations to high precision. Since the overall algorithm is iterative, a small error in one solution step can be corrected in the next step. Instead of solving the equations to high precision, we use only a few iterations of an iterative matrix solver to solve Equation 3.10. Using this technique, the matrix solve times can be improved by more that a factor of 100. And, because there may be no exact solution, little is lost in solving the equations approximately.

As the velocity drag term in Equation 3.10 is small, the resulting solution $\vec{r}$ can be very large, significantly larger than one voxel. Because the image forces are estimated locally, those forces are no longer accurate after a displacement of roughly one voxel. Such inaccuracies can lead to instabilities in the convergence of the algorithm. We therefore limit the motion of the nodes of the mesh in any iteration to the spacing between the voxels. Should the maximum displacement of a node exceed that spacing, we scale $\vec{\delta r}$ so that the maximum displacement is exactly that spacing. This type of

method is a standard technique for solving non-linear equations and is referred to as velocity limiting.

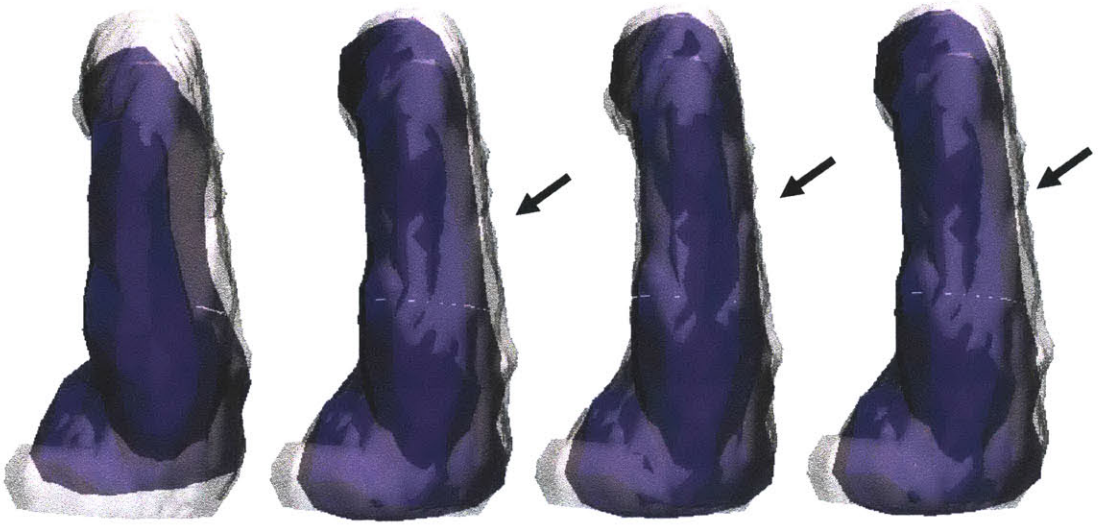


Figure 3-6: A target shape is shown in translucent white; the deformable model is in blue. From left to right: An initial alignment, after 12 iterations, after 13 iterations, after 14 iterations. The arrow shows that between iterations 12, 13 and 14 the dominate motion of the model was moving back and forth within the target.

Figure 3-6 shows a deformable model evolving using the algorithm we have just mentioned. In that match, the deformable model deformed most of the way to the target in the first 12 iterations. Afterwards, much of the changes in the model were simply moving back and forth within the target. The cause of this motion is that as the model moves to one side, the forces on that side go to zero, and increase on the other side. Thus, an oscillatory motion results.

There are two methods to obtain convergence. The first is to work with the Hessian H_p directly. One might expect that near the solution, the Hessian might lead to stable equations. Unfortunately, our experience is that the Hessian is almost always unstable. A second technique is to damp out oscillations using another drag force, a drag force on changes in velocity. We add such a drag in the form of $-\alpha(\vec{\delta r}_k - \vec{\delta r}_{k-1})$,

where α is positive. The final iteration becomes

$$\left[\frac{K}{\tau} + (\epsilon + \alpha)I\right] \delta\vec{r}_k = \frac{-K\vec{r}_k}{\tau} + \alpha\delta\vec{r}_{k-1} + \sum_{T\in\{l_T=1\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r}_k}\langle u|\vec{r}_k\rangle_T + \sum_{T\in\{l_T=0\}} \frac{\mathcal{V}_T}{\mathcal{V}_{voxel}} \frac{d}{d\vec{r}_k}\langle -u|\vec{r}\rangle_T. \tag{3.11}$$

3.3.5 Matching Parameters

It is challenging to set parameters for Equation 3.11 that will create an accurate matcher. For example, if the bending forces are too large compared to the image forces, the mesh will never move at all. If the bending forces are too small compared to the image forces, the mesh is torn apart.

When matching across populations, the elastic properties of the mesh are not true material properties; instead they are pseudo-material properties that lead to a solution with desirable smoothness characteristics. Thus, we end up choosing material properties that work well. We set Young's modulus and Poisson's ratio to the numbers of Vulcanized Rubber; we use Y of ten thousand pounds per square inch (psi), and ν of 0.4. We choose τ as 10^6, so that $Y/\tau = 0.01$. Using this form, the terms in the matrix K/τ are mostly within an order of magnitude of 0.01.

As Equation 3.11 includes both a velocity drag and an acceleration drag, it is effectively a second order temporal, partial differential equation. Such equations can generally be described as overdamped, underdamped or critically damped. We seek equations that are critically damped or overdamped in order to prevent oscillations. However, we don't want to overdamp the system so much that it never moves anywhere. We choose $\alpha = 0.01$ and $\epsilon = 10^{-4}$ which accomplishes this goal.

3.3.6 Validation

In this chapter, we will only be able to determine whether correspondences found are good ones by examining them visually. In the coming chapters, we will use the correspondences found as the basis for other algorithms. The success of those

algorithms will be a good indication of the usefulness of the matches found here.

We will be able to evaluate whether the deformable model was able to deform into its target. We use the 80% Hausdorff distance to evaluate the distance between the deformable model and the target shape [HKR93, Lev00]. To compute that distance, we find the distribution of distances from the surface of the first object to the closest point on the second object. We then repeat the process from the second object to the first object, take the fourth quintile of the non-zero values in each distribution, and use the maximum of the two numbers. We choose the 80% Hausdorff distance instead of the full Hausdorff distance in order to reduce the effect of outliers. When matching manually segmented objects, we expect such outliers due to small errors in manual segmentation.

3.4 Amygdala-Hippocampus Complex Dataset



Figure 3-7: The location of the left amygdala and hippocampus in the brain. The two structures are shown in green in the image. The hippocampus is the long structure and the amygdala is the bulbous structure. In an MRI, it is difficult to tell the two structures apart so that they are typically segmented as one large complex.

In the first set of experiments, we consider a data set of 30 left amygdala-hippocampus complexes, and separately 30 right amygdala-hippocampus complexes that were taken from a study of schizophrenic and control subjects [SKJ+92]. The complex is found

in the brain as shown in Figure 3-7. The amygdala is the bulbous shape shown in light green; the hippocampus is the longer shape shown in green.

In this study, one example structure was meshed and matched to the remaining 29 complexes. The meshes contained nearly 1500 nodes. The resulting matching algorithm converged in 20 to 50 iterations, depending on the case. This computation generally required between 30 and 90 second on a 1.5 ghz desktop computer.

Examples of a few correspondences are shown in Figure 3-8. The correspondences found appear appropriate. For example, Point 1 in the figure is consistently at the tail of the hippocampus, no matter at what angle the amygdala lies. Point 2 in the figure appears in roughly the same spot in the amygdala. Point 3 in the figure appears somewhat after the bend in the hippocampus tail.

The Hausdorff distances between the matches are shown in Figure 3-9. Most of the distances are less than 1 voxel, indicating good agreement between the final deformed model and the target shape. A few of the distances are somewhat larger. An examination of the larger Hausdorff distances found errors in the segmentations as shown in Figure 3-1. In a couple of the examples with larger Hausdorff distances, the target had features that were not found in the deformable model. Such an example is shown in Figure 3-10. That target shape has two ends to the tail of the amygdala; the deformable model has one. There is also no small crease between the amygdala and the hippocampus in that target; there is such a crease in the deformable model.

3.5 Thalamus Dataset

In a second set of experiments, we consider 53 segmented left and right thalami. The thalamus is a structure in the brain as shown in Figure 3-11. These thalami come from a study of normal and first episode schizophrenia patients that we will describe in a following chapter.

Time to convergence for the thalamus matching problem is very similar to the amygdala matching problem. For meshes of roughly 1500 nodes, the matching algorithm converged in typically 20 to 50 iterations, in less than a minute on a 1.5 Ghz



Figure 3-8: The colored spheres indicate a few points on the surfaces of six amygdala-hippocampus complexes that were found to correspond. Points 1, 2, and 3 are referenced in the text.

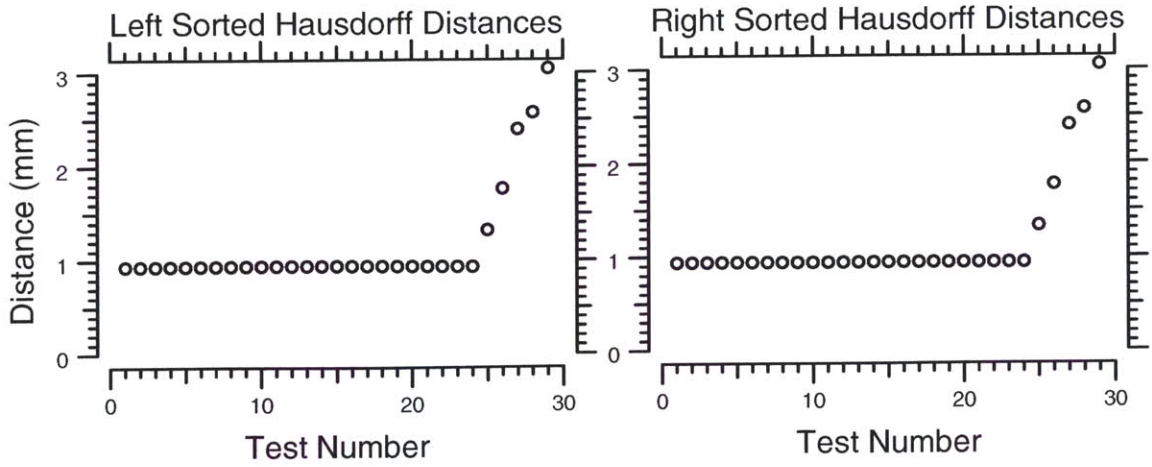


Figure 3-9: 80% Hausdorff distances between the amygdala hippocampus complexes and the warped mesh matched to them.

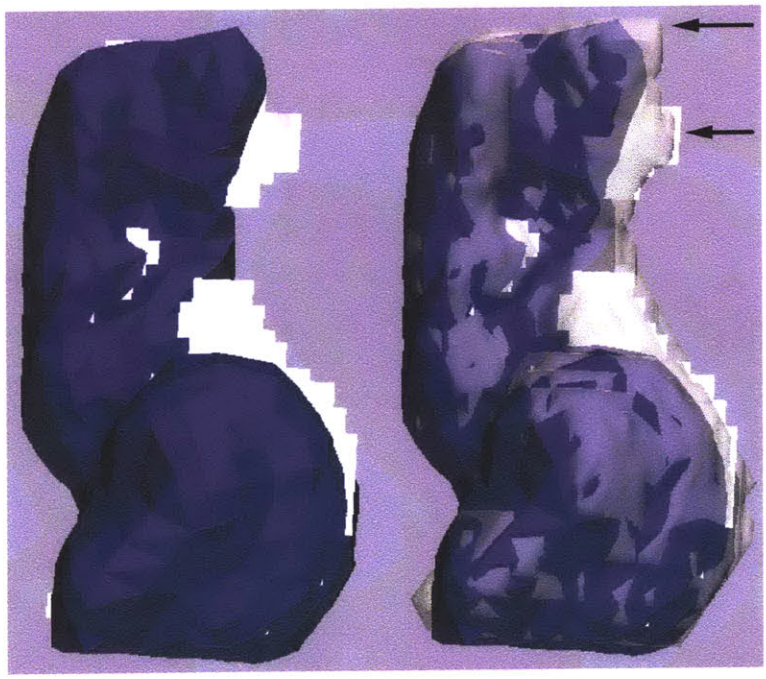


Figure 3-10: Illustrations of errors in matching for one amygdala-hippocampus complex. The blue surface represents the deformable model. The white jagged edge is the edge of a slice through the target complex. On the right image, a translucent white surface of the target image has been added. The two models are within one voxel of each other in all places except at two locations. The first error occurs at the top of the image where there are two "ends" of the hippocampus tail, as indicated by the arrows. The deformable model matched to one and not the other. The second error is in the center of the image. There, the amygdala did not expand upwards into the "bulge" in the target as it cost too must elastic energy to do so. In particular, the deformable model has a small crease between the hippocampus and the amygdala. The target has no such crease, and therefore the match in that region is difficult.

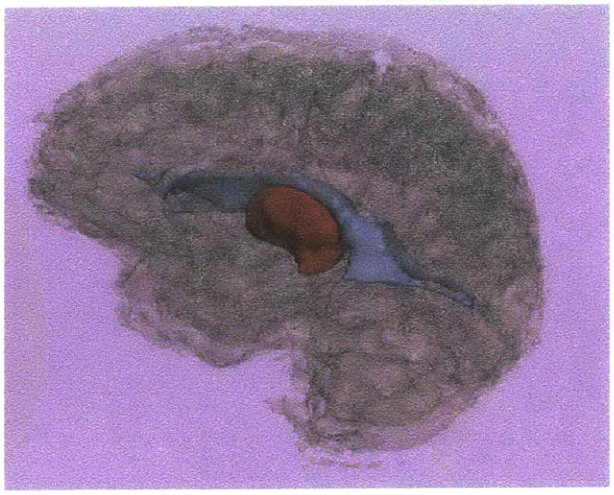


Figure 3-11: The location of the left thalamus in the brain. The structure is shown in red.

desktop machine.

Examples of a few correspondences are shown in Figure 3-12. The correspondences found appear appropriate. For example, Point 1 in the figure is consistently to the left of the hook-shaped part of the thalamus (lateral geniculate). Point 2 is consistently near the lower end of the thalamus (anterior portion).

As with the amygdala-hippocampus complex study, the thalamus study matches were generally very good, with Hausdorff distances at or below one voxel. Several distances were larger as indicated by Figure 3-13. Some of those errors were caused by "holes" in the segmentation where the segmenter left out a voxel in the middle of the structure. Other errors in Figure 3-13 came from problems in capturing protrusions that existed on some thalami. One such protrusion appears prominently in the top center thalamus in Figure 3-12, in the top right of the thalamus.

3.6 Discussion and Conclusion

We created a non-rigid matching algorithm to find correspondences between shapes without using landmarks. Our method uses a balance between linear elasticity and image agreement, and is able to overcome the ill-conditioned nature of the resulting equations. By using the compact tetrahedral representations developed in the previ-

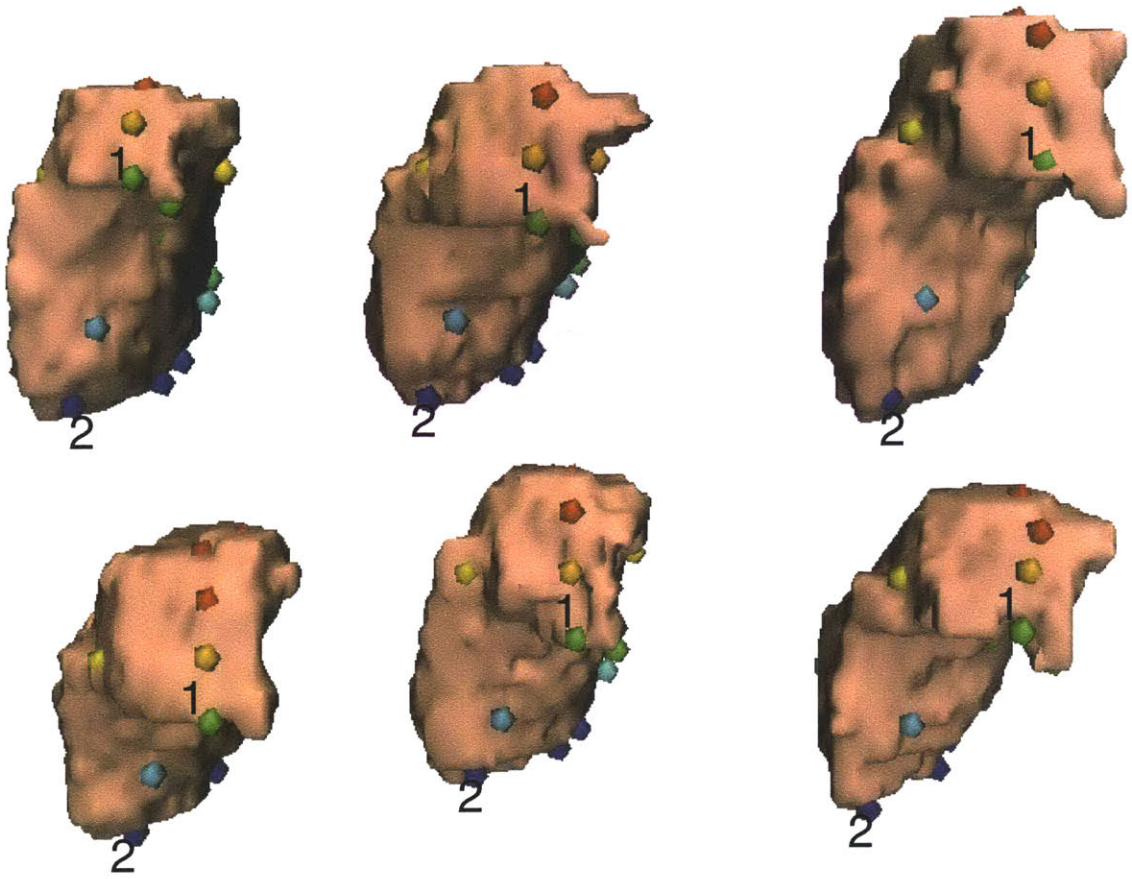


Figure 3-12: The colored spheres indicate a few corresponding points on the surfaces of six right thalami. The numbered points are referenced in the text.

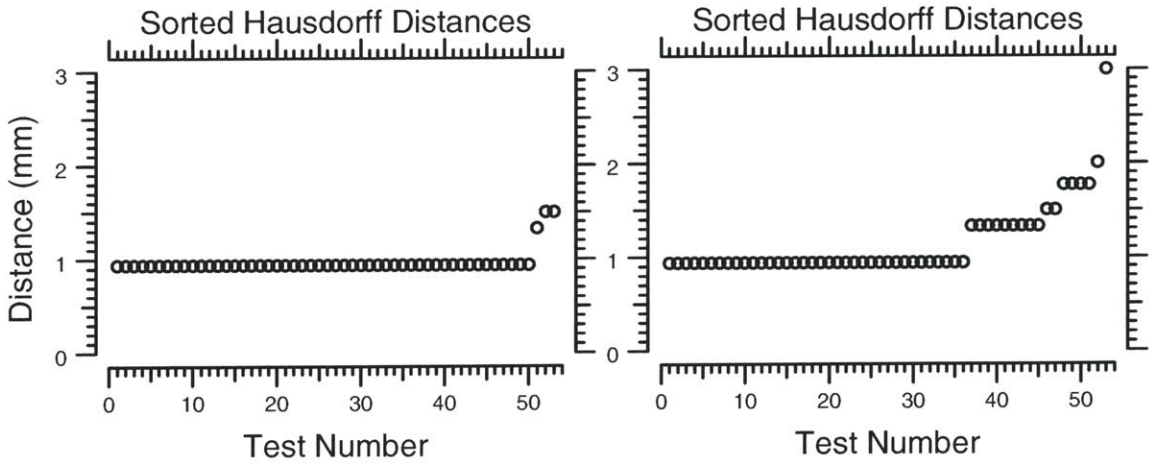


Figure 3-13: 80% Hausdorff distances between the thalami and the deformable template. The left plot is for the left thalami; the right plot is for the right thalami. Several of the larger distances result from holes in the thalamus segmentation.

ous chapter, the algorithm converges quickly, generally in less than one minute. For most cases, the method was able to deform a template into a target with errors less than one voxel. The algorithm did have problems matching to features in a target shape that were not present in the deformable model. We view this as an indication that the template we chose was not the optimal one for the data set. In fact, it is reasonable to expect that when choosing one shape as a template to match to other shapes that the template will be missing features found in the remainder of the data set. A better method would find a model that captures the variations in the examples to be matched. In fact, one could imagine forming an adaptive model that introduces variability only where the set of example shapes call for it. For example, Davies *et al.* have developed such a model [DCT01] in a minimum description framework.

In general, free-form matchers are susceptible to finding local minima in the matching equations that are not necessarily valid. We gave the example in Figure 3-10 of the target having two tails and the deformable model having one. As the model is not guided in any way, we expect that the final result in such a matching problem would strongly depend on the initial alignment. For more complicated structures which we did not examine here, we would therefore expect our matcher to yield poor results. For example, using our methods to match cortical folds between subjects would likely result in folds in the deformable model being attracted to the wrong folds in the target.

Another important limitation of our algorithm is the possibility of having a tetrahedron invert during the matching process. We chose the parameters in our model carefully so that tetrahedron collapse was not a problem in the experiments. However, it is reasonable to expect that we will find shape matching challenges where tetrahedron inversion will be a problem.

To end, we remind the reader that while the correspondences found here appear reasonable, we have not yet shown they are good ones. In the coming chapters, we will use the correspondences found here as the basis for other algorithms. If the correspondence found here are good ones, our expectation is that algorithms that use them will work well.

Chapter 4

Application: Morphological Studies

Statistical studies of shape generally compare the shape of structures selected from two different groups. Such studies are used to test hypotheses about the anatomical differences between genders [GSSL01, DVR+96, MG98] and to understand the effects of aging on anatomy [Dav96, SGA+84]. Morphological studies are also used to form connections between shape and the presence of disease [GSSL01, GGSK01]. For this application, physicians are interested in developing early diagnosis tools and in better modeling the disease.

We pursue shape studies as a method of demonstrating the usefulness of the correspondences found in the previous chapter. We will show that a shape study based on those correspondences shows improved performance over the same shape study based on other representations.

We begin by reviewing some of the issues that are important to the success of a shape study, including representation, alignment technique, volume normalization and the choice of statistical classifier. We then create a representation of shape based on the correspondences found in the previous chapter. We then use that representation in a morphological study of segmented amygdala-hippocampus complexes from fifteen schizophrenic subjects and fifteen control subjects [SKJ+92]. This study has been examined by other researchers using different representations so that the sensitivity

of the performance of the shape study to representation used can be explored. We further examine the effects of other key factors on the quality of the results including alignment technique, volume normalization and choice of statistical classifier. After demonstrating the success of our methods for the amygdala-hippocampus study, we create a study of the thalamus in eighteen first episode schizophrenia subjects and eighteen normal control subjects. We are the first to find correlation in shape of the thalamus with schizophrenia.

4.1 Background: Performance Issues in Morphological Studies

The process of finding morphology differences between classes of shapes typically involves three steps. First, a descriptor of shape is found to represent each of the input data. Second, a classifier is trained to attempt to separate the two groups of shapes. Third, differences between shapes are mapped back onto the space of the input data to visualize shape differences.

4.1.1 Choice of Shape Descriptor

There are numerous attractive representations of shape for use in a morphological study. There are generally two key factors that guide a choice of a descriptor: ease of extraction and sensitivity to noise. One might expect that a choice of shape descriptor would be guided by the ability of a descriptor to capture morphological differences between groups. To our knowledge, the results in this chapter are the first comparing the effectiveness of different shape descriptions in a morphological study. Thus, it is yet to be demonstrated which shape descriptors better capture morphological differences.

Shape Descriptors

We divide shape descriptors into two classes: those that establish explicit surface correspondences between shapes and those that do not. The correspondence-based representations are generally more difficult to form than the correspondence-free representations. This relative difficulty is partly due to the challenge of finding correspondences, which was discussed in the previous chapter. It is also stems from the ability to form a correspondence-free representations of a particular shape in isolation from the other shapes. For example, spherical harmonics can be used to implicitly represent a surface [GSSL01] and are therefore in the correspondence-free class of shape descriptor. Using spherical harmonics, the representation of each shape can be described using little or no information from the other shapes. Conversely, when correspondences are established, it is generally necessary to examine at least two input shapes at one time.

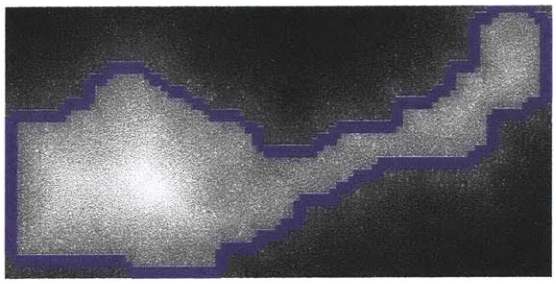


Figure 4-1: A slice through a signed distance map representation of an amygdala-hippocampus complex. Each voxel in a volume is labeled by its distance from the surface of the shape. The sign of each label is changed so that voxels inside the object have positive sign and outside the object have negative sign. Image provided by Polina Golland.

Examples of correspondence-free representations include spherical harmonics, distance maps and medial representations or skeletons. Distance maps embed a surface by labeling the voxels of a 3D-volume with the distance from the surface [GGSK01]. An example is shown in Figure 4-1. Medial representations [PFY+96] are closely related to skeletons. They parameterize shapes based on a chain or sheet of atoms that project a surface. All of the representations in this class have the property that they implicitly determine a surface so that no explicit correspondences are made.

And, these descriptors of shape have the property that once the parameters of the representation are chosen, the representation of each shape can be formed separately.

Members of the class of correspondence-based representations include polygonal surface meshes, displacement fields, and volume element based representations. Polygonal surface meshes describe each shape using the same mesh, where the vertices of the mesh on each input shape are placed at corresponding points [CTCG92]. Displacement fields are sets of vectors that transform one example shape to the other [DVR+96, TGG+02]. This representation is not only based on correspondences at the surface of an object, but is based on the correspondences in the interior. Because shapes are typically assumed to be uniform in their interior, correspondences in the interior are frequently interpolated from the surface. Volume element based representations are based on displacement fields [MG98]. In a volume element based representation, an input sample is divided into small pieces which expand or contract as the shape is deformed into other shapes. The representation of each shape is the expansion or compression of one input sample into the remaining shapes. Thus, while displacement fields describe directions of change in an example shape, volumetric elements describe compression and expansion.

In this chapter, we create classifiers based on one correspondence-free representation, distance maps, and one correspondence-based representation, displacement fields. We also briefly examine a classifier based on volume elements in Section 4.3.3.

Confounding Factors

After a representation has been extracted of each input shape, a classifier is typically used to separate the two groups. The choice of representation effects the performance of the classifier; there are three major confounding factors of a classifier that can guide the choice of a representation in a morphological study. The first such factor is shape alignment. Variations in patient position during imaging causes misalignment between data sets. This misalignment can confuse the training phase of a classifier. That is, a classifier must find differences between the data sets while ignoring affects due to alignment errors. To mitigate the effects of misalignment, one typically rigidly



Figure 4-2: Illustration of the challenges of automatic alignment. To align two shapes, one typically places the central point of the shapes at same location and then rotates the shapes into alignment. For this example of a black circle (left) aligning to a green circle with a bump (center), a desirable outcome is that the center of the two circles become co-located in alignment. But, many alignment techniques would find the result on the right, with the center of the two shapes co-located, not the center of two circles.

aligns the data before classification. In this chapter, we examine several different alignment methods in the attempt to quantify the importance of rigid alignment technique.

Misalignment can be addressed in more than one way. Correcting the misalignment is one method, although as illustrated in Figure 4-2, alignment techniques sometimes do not yield the most desirable outcome. A second method for addressing misalignment is to choose a representation that is independent of the orientation of a shape. Spherical harmonic representations can satisfy this requirement; volumetric elements can also solve this requirement. To see if this second method might be effective, we examine the volume elements representation in Section 4.3.3

Volumetric effects are a second factor that can confuse a classifier. As subjects are generally of varying size, one would expect their anatomical structures to be of different size. Thus, volume differences in anatomical structures can be unrelated to the differences between groups. However, when studying aging for example, Davatzikos finds that parts of the brain atrophy with age[Dav96]. Thus, volume differences can be directly related to the differences between groups. In this chapter, we examine the effects of normalizing the data by volume.

A third confounding factor is noise in the determination of the input surfaces. It is often the case that there are ambiguities in the exact location of a surface. Even for an individual experienced in segmenting a particular structure, the decision whether to include a particular voxel in the structure of interest can be difficult [Dem02]. Also,

both manual segmenters and automatic segmentation algorithms are subject to error. It is therefore important that a description of shape lead to a classifier that is robust to small changes in the surface measurement. For example, spherical harmonics may not be an optimal surface descriptor because a small change in the surface can entirely change the set of coefficients describing the surface.

Both displacement fields and distance maps are somewhat robust to noise. Using distance maps, small changes in the surface of an object affect only a small set of voxels in the representation [Gol01]. Displacements fields are also somewhat robust to noise in the determination of the surface. As interior correspondences between shapes are usually found by weighted averages of the displacement of the surface, interior displacements typically change a small amount due errors in the determination of the surface. Furthermore, the non-rigid matcher developed in the last chapter – which we will used to generate displacement fields – was specifically designed to mitigate the effects of errors in surface determination such as protrusions.

4.1.2 Statistical Separation

Once representations of the input shapes have been formed, they are used in a statistical separation method to attempt to capture differences between the classes of shapes. Early research in this area used technology that was originally developed for automatic segmentation [CHTH92]. In automatic segmentation, a deformable model of shape is often formed using Principal Component Analysis (PCA) on representations of example structures (this method will be described in Chapter 5). A generative model can then be created by allowing the shape to deform along the most important modes of variation. To compare two groups, one forms such a model for each group, and separates the two models using a hyperplane. Visualization of differences between classes can be examined by moving perpendicular to the hyperplane [MG98, MPK94].

The biggest concern with this method is that the number of example data sets in medical images is often in the tens, not the hundreds or thousands. Using PCA on each class effectively estimates a Gaussian probability density over each data set. It

is unclear whether an accurate estimate of the probability density can be formed with such a small numbers of examples. Furthermore, it is entirely unclear if the Gaussian assumption is appropriate.

More recently, Golland *et al.* [GGSK01] introduced an approach that avoids making explicit assumptions on probability densities over the input data. They used non-parametric support vector machine (SVM) based classification methods to perform morphological studies of shape and demonstrated that non-linear SVM classification can potentially improve the separation between groups of shapes over linear methods. Furthermore, they showed that the gradient of the classifier can be used to show differences between classes of shapes.

While non-linear methods outperform linear methods in classification accuracy, there are some advantages to linear classifiers. First, the shape differences found by linear classifiers are much easier to interpret than those found by non-linear classifiers. Second, because there are usually a small number of examples in a medical classification problem, there is a danger that a non-linear classifier will over-fit the data and find differences between the groups that will not generalize to further examples [Gol01]. Because of the advantages and disadvantages of each method, we explore classifiers using both linear and non-linear methods.

Review of Support Vector Machines

We briefly review the machinery of support vector machines we will use. Readers can see [Gol01] for more details. We begin by letting $\vec{x}$ be the representation vector of the shapes for a representation. The squared distance between two shapes, $||\vec{x}-\vec{x'}||^2$, is defined to be $(\vec{x}-\vec{x'})^T(\vec{x}-\vec{x'})$. For displacement fields, this distance is simply the square of the length(l_2 norm) of the samples of the displacement field between each shape. For distance maps, there is no simple interpretation of distance between shapes.

Support vector machines (SVM) form the basis of the classification methods we will use. Given m measurements $\{\vec{x}_k\}$, with labels $\{y_k\}$ which are 1 for one group

and -1 for the other, the SVM finds the optimal classification function

$$f_K(\vec{x}) = \sum_{k=1}^{m} \alpha_k y_k K(\vec{x}, \vec{x}_k) + b \tag{4.1}$$

where b and α_k are constants, K is the kernel function, and new examples $\vec{x}$ are classified based on the sign of f_K. The kernel function maps two examples into a high dimensional space and takes a dot product in that space. The SVM effectively finds an optimal separating hyperplane in that higher dimensional space. The separating hyperplane is optimal in the sense that the distance from the examples to the hyperplane is maximized.

For the non-linear classifier, we use a family of radial basis functions (RBF) $K(\vec{x}, \vec{x}_k) = -e^{||\vec{x}-\vec{x}_k||^2/\gamma}$ where γ is proportional to the square of the width of the kernel. We pick γ to optimize the cross validation accuracy of the classifier, which is evaluated by leave one out testing. For the linear classifier, the kernel function is simply the dot product: $K(\vec{x}, \vec{x}_k) = \vec{x} \cdot \vec{x}_k$.

4.1.3 Visualization of Differences

The goal of a morphological study may be to create a classifier to identify new example shapes as being a member of one or the other group. Such a classifier could be used to as an early diagnostic test, for example. More often, the goal of a morphological study is to explicitly represent and visualize the shape differences between groups.

A linear classifier uses a hyper-plane to separate the groups. Thus, the shape differences between groups are effectively encoded in the normal to the hyperplane. For the non-linear kernel, differentiating the pre-thresholded, RBF classifying function with respect to shape would seem to yield the answer [GGSK01]. The derivative at $\vec{x}$ is given by:

$$\frac{2}{\gamma} \sum_{k=1}^{m} \alpha_k y_k \gamma (\vec{x_k} - \vec{x}) e^{-||\vec{x}-\vec{x}_k||^2} \tag{4.2}$$

where the $\{\alpha_k\}$ are constraints determined by the SVM and $\{y_k\}$ are -1 for on group and 1 for the other. Using distance maps the derivative is not sufficient to find the

shape differences between classes. A small change to a distance map does not yield another distance map. To address this issue, one must project a derivative back onto the manifold of distance maps [GGSK01]. However, displacement fields form a vector space; a small change in a displacement field yields another displacement field. Thus for this case, differentiating the classifier is sufficient for our goals.

4.2 Methods

Having reviewed the issues and methods that are important in a morphological study, we develop methods to perform such a study based on displacement fields. To form the displacement field representation, one of the example shapes is chosen as a template and meshed with tetrahedra to facilitate the matching process. The mesh is then treated as a linear elastic material and deformed in order to match the other example shapes as described in the previous chapter and in [TGKW01]. The nodes of the tetrahedral mesh form a roughly uniform sampling of the displacement field. The vector representation of shape is simply a the displacements of the nodes of the tetrahedra from the template assembled into a large vector. In the studies we will consider, there will be shapes from cortical substructures on the right and left side of the brain. Templates are formed for each side separately. When both structures are consider together, the vectors on each side are simply concatenated.

In the following sections, we explore several issues that are important to shape-based classification including normalization by volume, alignment technique, and the choice of using a linear or a non-linear classifier.

In each section, we will indicate whether data have been normalized by volume, or not scaled at all. When normalization was used, each shape was isotropically scaled to the same interior volume. Note that since we will examined structures in the brain, normalizing by brain volume may be biologically relevant, but unfortunately brain volumes were not available for all subjects.

Except where stated, all results use a second order moment alignment of the data. In Section 4.3.3, we also align using the second order moments of the mesh,

and absolute orientation [Hor87] (removing global translations and rotations from a deformation field). Also in that section we test two additional representations that are independent of orientation. The first representation is a vector of the volumes of the tetrahedra used to describe each structure. That is, the tetrahedra in one shape are ordered and the volumes placed in a vector according to that order. The same vector can be formed for the other shapes using the volumes of the corresponding tetrahedra for each shape. Rather than using the volumes of tetrahedra, one can also use the edge lengths of the tetrahedra for a representation that is independent of global alignment. Thus, we consider a second representation that is a vector of the edge lengths of the tetrahedral meshes used to describe each shape.

We use support vector machines (SVM) as the basis for classification, using both RBF and linear kernels as described in Section 4.1.2.

4.3 Results: Amygdala-Hippocampus Study

The first study we examine consists of thirty hand segmented amygdala-hippocampus complexes. Half of the complexes came from subjects suffering from chronic schizophrenia and half from control subjects [SKJ+92]. This study was chosen because it has already been examined by other researchers so that comparisons of results can be made.

We compare our results using displacement fields to results provided by Polina Golland using signed distance maps. The distance maps are formed by labeling each voxel with its distance from a surface as described in [GGSK01]. The resulting representation is the vector of labels of the voxels.

We also report results by Gerig *et al.* [GSSL01] who examined the same data set. They formed a classifier explicitly comparing the asymmetry of the left and right amygdala. Each shape was normalized by volume and represented using spherical harmonics. The left and right complexes were overlayed, and points on one surface were set to correspond to the closest point on the other surface. Gerig *et al.* used a support vector machine (SVM) using radial basis functions (RBFs) to classify based

on correspondences. They furthermore added volume as a separate variable and made a classifier based on both volume and shape differences.

4.3.1 Comparison of Representation

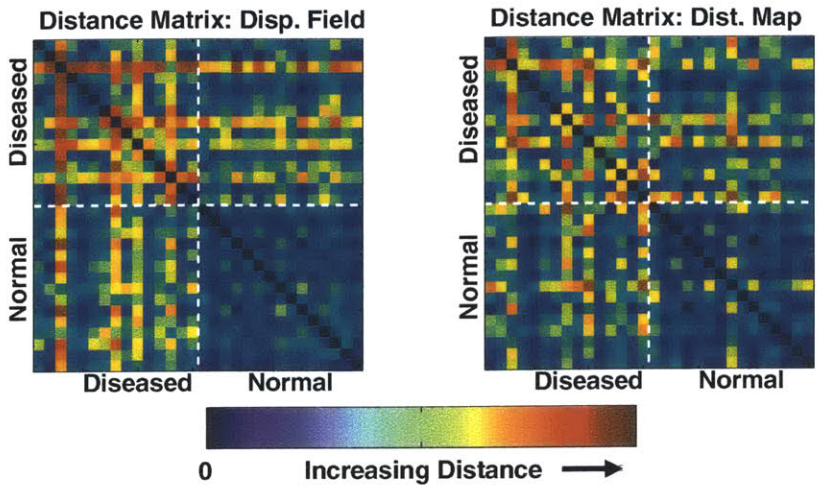


Figure 4-3: The squared distances between the 30 amygdala-hippocampi. The left matrix uses a displacement field representation; the right uses distance maps. Numbers are mapped linearly to the colorbar shown. Each row represents the squared distance between a particular complex and all the complexes. This data is used to compute the last row in Table 4.1. It uses both complexes, aligned by second-order moments of the label-maps, without any normalization for volume.

Once the representations of the input shapes have been formed, the first step in classification is to determine the distances between the shapes. To classify well, one would intutively expect that examples from one group would be similar to other example shapes from the same group, but be different from examples in the other group. Figure 4-3 shows some, but not all of those properties.

Figure 4-3 shows the distance between amygdala-hippocampus complexes using the displacement field representation and the distance map representation. The two distance matrices show strong similarities. In both cases the distances between the complexes from normal subjects (lower right quadrant) were relatively small. Conversely, examining the distances between complexes from diseased subjects (upper left quadrant), many of them strongly differ from each other. Examining the off-diagonal

Structure	Cross Validation Accuracy using Linear Classifier		
	Displacement Field	Distance Map	
Left Complex	$63 \pm 18\%$	$57 \pm 18\%$	
Right Complex	$63 \pm 18\%$	$53 \pm 18\%$	
Both Complexes	$67 \pm 18\%$	$53 \pm 18\%$	
Structure	Cross Validation Accuracy using RBF		
	Displacement Field	Distance Map	Gerig *et al.*
Left Complex	$80 \pm 17\%$	$73 \pm 17\%$	
Right Complex	$77 \pm 17\%$	$70 \pm 17\%$	
Both Complexes	$87 \pm 16\%$	$70 \pm 17\%$	$87 \pm 16\%$

Table 4.1: Cross-validation accuracy for the different representations using RBF and linear classifiers. For the third column, volume was added separately to shape data [GSSL01]. The range is the 95% confidence interval [Bey91].

Normalized Structure	Cross Validation Accuracy using Linear Classifier		
	Displacement Field	Distance Map	
Left Complex	$57 \pm 18\%$	$57 \pm 18\%$	
Right Complex	$70 \pm 18\%$	$53 \pm 18\%$	
Both Complexes	$70 \pm 18\%$	$53 \pm 18\%$	
Normalized Structure	Cross Validation Accuracy using RBF		
	Displacement Field	Distance Map	Gerig *et al.*
Left Complex	$73 \pm 18\%$	$70 \pm 17\%$	
Right Complex	$77 \pm 17\%$	$77 \pm 17\%$	
Both Complexes	$87 \pm 16\%$	$67 \pm 18\%$	$73 \pm 17\%$

Table 4.2: A comparison of cross-validation accuracy for RBF and linear classifiers. The data is normalized to the same volume. The range is the 95 % confidence interval.

quadrants of the matricies, one can see that in both representations, some of the complexes from diseased subjects are similar to complexes from normal subjects.

Because Figure 4-3 does not show all the properties one would expect to make a good classifier, one might expect that the resulting classifier would not perform perfectly. Table 4.1 illustrates this result. None of the classifiers have perfect cross-validation accuracy.

Table 4.1 shows that linear classifiers based on displacement fields seem to outperform linear classifiers based on distance maps by approximately 10 percentage points. For nonlinear classifiers that use RBFs, again those classifiers based on displacement fields performed better than those based on distance maps, by roughly 17 percentage points. Also note that classifiers based on displacement fields were able to improve

using information from the right and left complexes, while the same does not hold for classifiers based on distance maps.

Gerig *et al.* [GSSL01] examined the data set using a method that explicitly examined left right complex assymetry. They also used an RBF based classifier, and found similar results to the displacement field based classifier as shown in Table 4.1.

4.3.2 Effects of Volume Normalization

One of the issues faced in a shape study is whether or not normalizing by volume will affect the results. Table 4.2 shows classification results with all the input data normalized to the same volume and aligned by second order moments. For this study, normalizing by volume had very small effects. For most entries in tabels, there are only small differences between Table 4.2 and Table 4.1. Most of the differences are 3 percentage points or less. However, there are two entries that decreased by 6 percentage points. Note that the classifier by Gerig *et al.* decreased by 14 percentage points. Thus the methods used by Gerig seem to be more sensitive to volumetric effects for this study.

4.3.3 Effects of Alignment Technique

Using displacement fields, several different alignment techniques may be appropriate. We examined several alignment methods to see if they would produce significantly different results. Figure 4-4 shows the effect of three different alignment methods on the resulting classification accuracy for both linear and RBF based classifiers, without normalizing by volume. The effect of alignment caused a range generally 10 percentage points or less between the best and worst cross validation accuracy – though in one case, the linear classifier using both the left and right complexes, the range was 17 percentage points. The first two alignment methods are both based on second order moments and therefore one would expect would yield very similar results. For the non-linear RBF based classifier, this result was found; the largest effect was 3 percentage points. For the linear classifier, larger effects were observed. Thus, the

linear classifier shows results consistent with being more sensitive to alignment than the non-linear classifier.

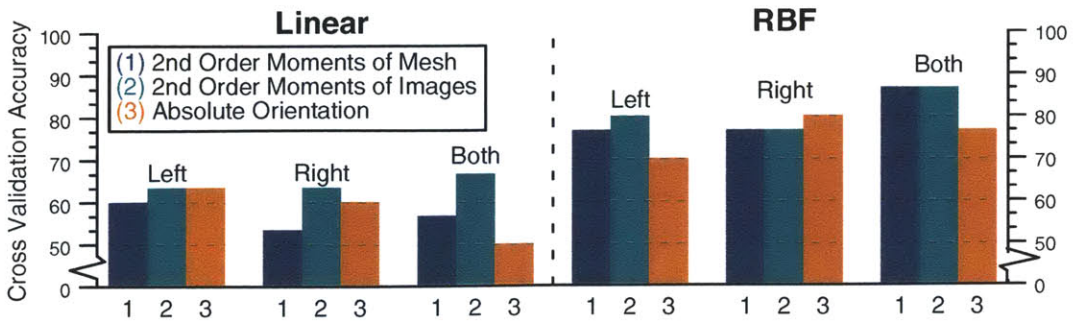


Figure 4-4: The effect of alignment method on classification accuracy using displacement fields as a representation. We used three different alignment techniques. Method (1) aligned the meshes on the basis of the second order moments of the meshes. Method (2) aligned the meshes by the second order moments of the labelmaps (which was used in Tables 4.2 and 4.1. Method (3) aligned the meshes by absolute orientation [Hor87]. For both linear and RBF based classifiers, "Left", "Right" and "Both" refers to a classifier based on just the left complex, just the right complex or both.

Because of the observed increased sensitivity of the linear classifier to rigid alignment technique, we hypothesized that a linear classifier based on a representation that was independent of orientation might show increased performance over the linear classifier based on displacement fields. We therefore created two new representations: the first a vector consisting of the volumes of the tetrahedra in the mesh, the second a vector consisting of the edge lengths of the tetrahedra. Those representations both yielded classification accuracies between 67 and 76 percent for the left, right and combined complexes. This performance is as good or better than the best linear classifier based directly on displacement fields – though, it is interesting to note that that RBF based classifiers based on these two representations yielded no performance improvement over linear classifiers.

4.3.4 Visualization of Differences

One of the important goals of a shape study is to visualize the differences between the classes of shapes. Figure 4-5 and Figure 4-6 show those differences for the right and left complexes. Those differences were found from derivatives of the RBF based classifier

evaluated at examples as described in Section 4.1.3. There are several interesting observations about the figures. First, for a non-linear classifier it is not guaranteed that the gradients computed for the two groups are equal and opposite. However, Figure 4-5 and Figure 4-6 shows that for both classifiers, the changes detected to make one group look more like the other are similar in nature but opposite in sign. This finding holds for both complexes, and both representations.

Ideally, the shape differences found between the classes would be indpendent of representations. For right complexes, (Figure 4-5), there are strong similarities in the shape differences found by the classifiers based on each displacement fields and distance maps. In particular, at the posterior portion of the tail of the hippocampus (the top of the shapes shown) and the base of the hippocampus (the bottom of the shapes shown), the shape differences found by the two classifiers appear very similar. For left complexes (Figure 4-6), there are disparities between shape differences found by the classifiers based on each representation. These disparities are particularly evident at the base of the amygdala (the bottom of the shapes in the figure) and the tail of the hippocampus (near the top of the shapes in the figure).

Given that the two representations yielded different morphological differences, we seek evidence as to which answer is the correct one. There are two factors which suggest the answer. Firstly, examining the gradients found at every left complex (not shown), the displacement field based classifier yielded visually similar gradients at each left complex. However, for many of the left complexes, the gradients found using the distance map based classifier were visually very different from each other. Thus, because the shape differences found were consistent across all left complexes for displacement field based methods, but not for distance map based methods, the shape differences found using displacement fields are more likely. Note that for the right complex, the classifiers based on both representation yielded visually similar gradients at each right complex.

For a second indication of which of the two classifiers yielded the correct answer, we can consider a third classifier. Figure 4-7 shows the morphological differences found between the two groups as determined by a linear classifier using displacement

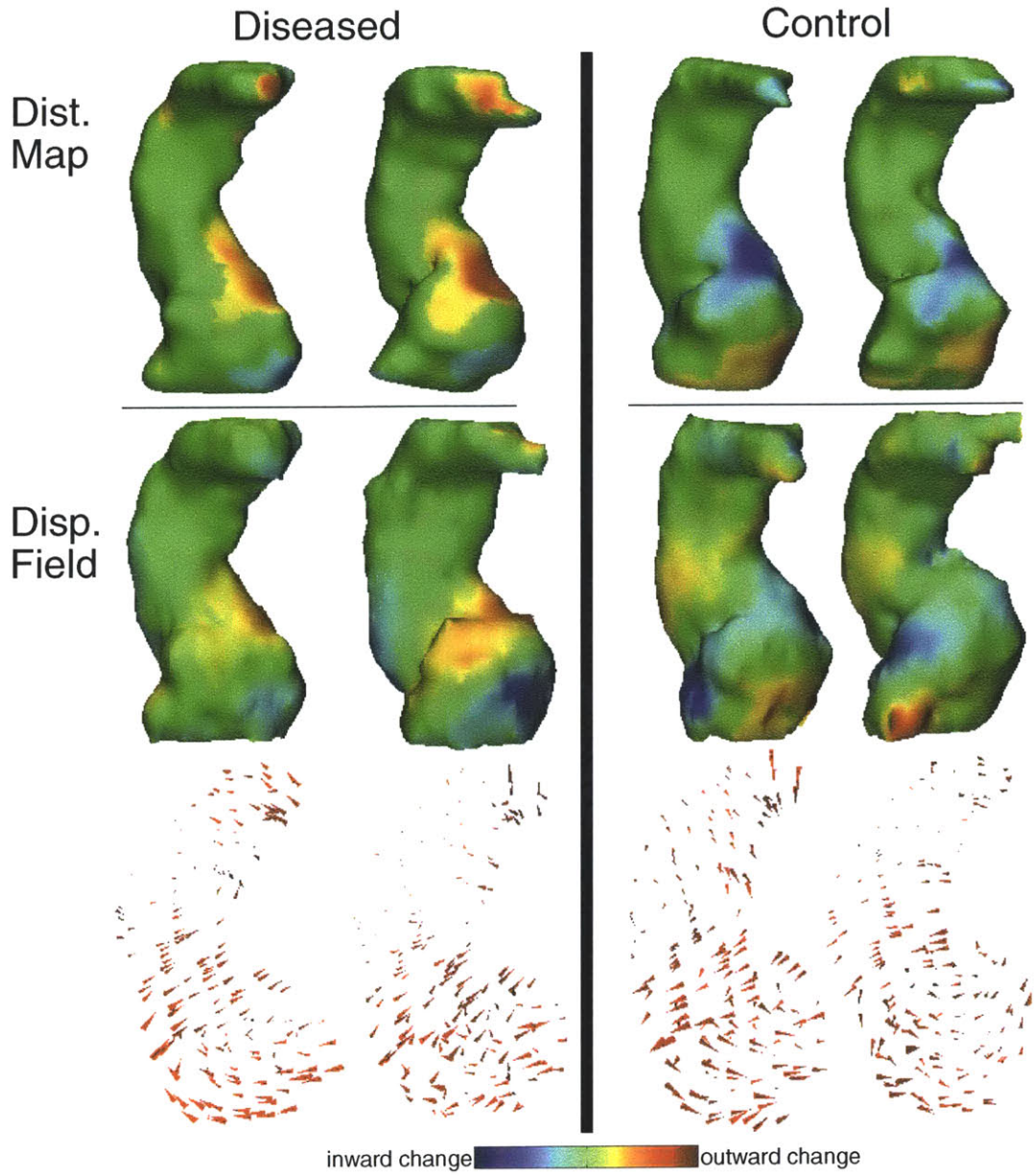


Figure 4-5: The top four plots show the "deformation" of the surfaces relative to the surface normal for the right amygdala-hippocampus complex. For diseased subjects, "deformation" indicates changes to make the complexes more like a normal control. For normal subjects, "deformation" indicates changes to make the complexes more diseased. The four pairs of surface plots show deformations of diseased/control subjects using distance maps/displacement fields as representations. In each pair of surface plots, the two examples with the largest deformations evaluated at the support vectors of the SVM classifier are shown; the larger one is on the left. The color coding is used to indicate the direction and magnitude of the deformation, changing from blue (inward) to green (no motion) to red (outward). The bottom two plots are the deformations fields used to generate the plots directly above them. Note that motion along the surface does not affect the colors in the surfaces.

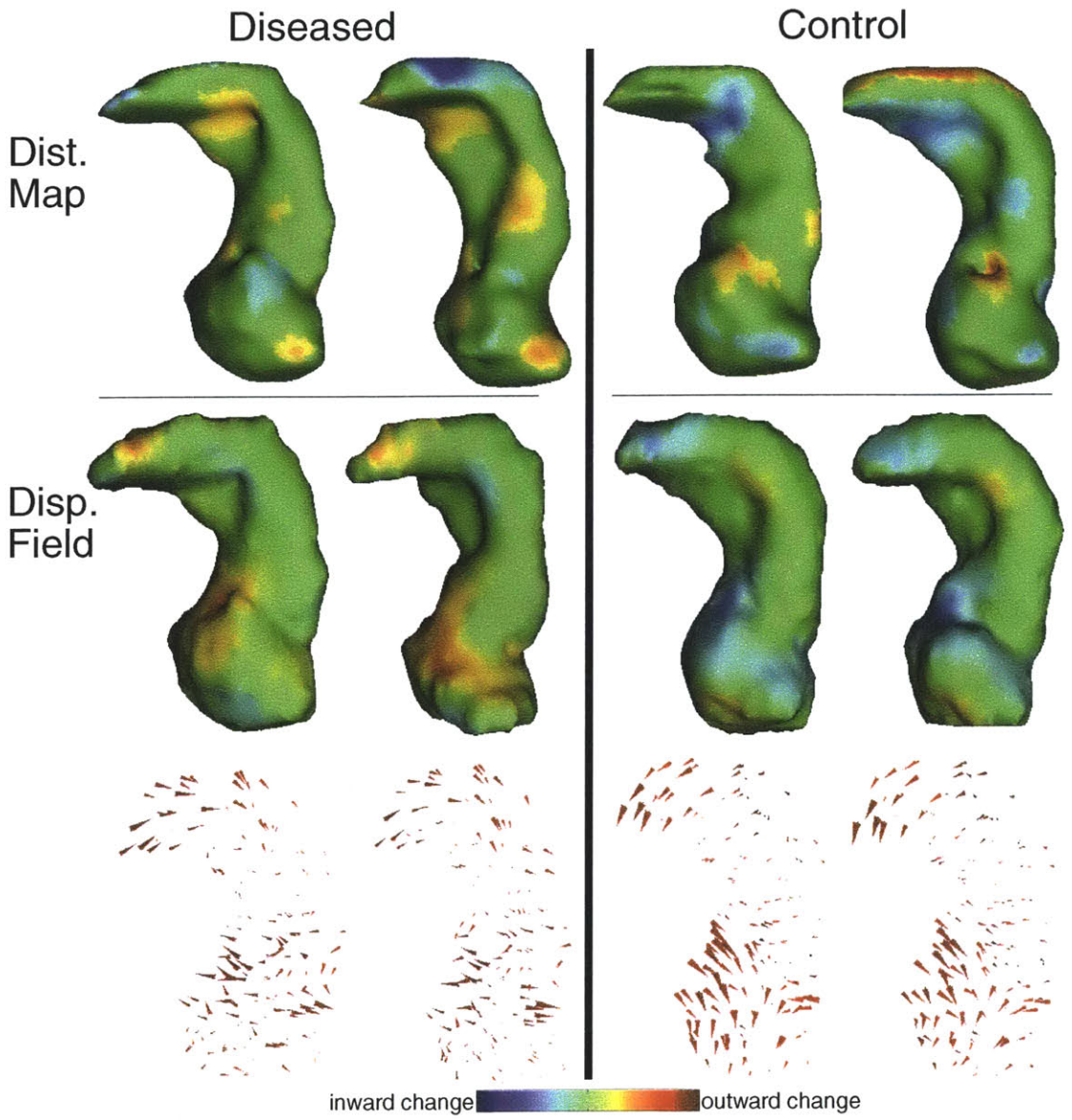


Figure 4-6: The top four plots show the deformation of the surfaces relative to the surface normal for the left amygdala-hippocampus complex. The figure is otherwise identical to Figure 4-5.

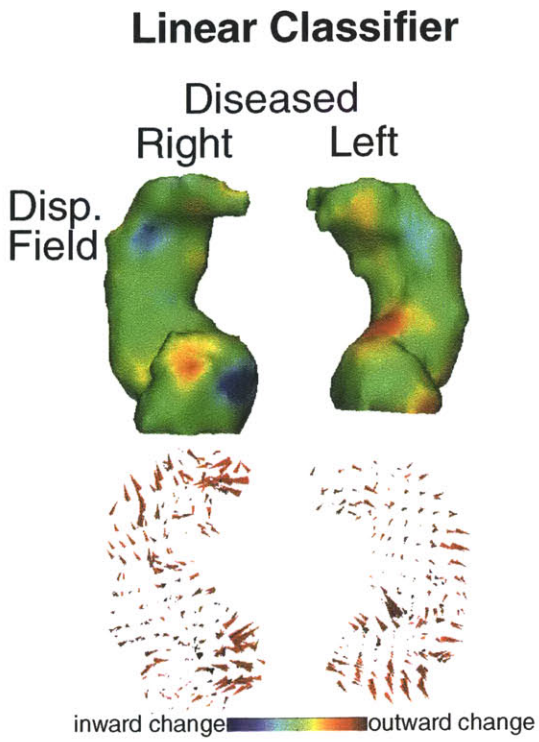


Figure 4-7: The "deformation" of the right and left complex determined using a linear classifier based on displacement fields. The results shown were determined by two separate classifiers, one for each side. Both example shapes are from the same diseased subject. The remainder of the figure is described in Figure 4-5.

fields. We consider only a linear classifer based on displacement fields and not a linear classifier based on distance maps due to the poor cross validation accuracy of the linear classifier based on distance maps. Comparing the shape differences in Figure 4-7 with those found in Figure 4-5 and Figure 4-6, there are several similarities in the shape differences found by the linear classifier and the non-linear classifier based on displacement fields, particularly at interface of the hippocampus and amygdala. Thus, the shape differences found using the linear classifier based on displacement fields supports the shape differences found by the non-linear classifier using displacement fields, not those of the non-linear classifier based on distance maps.

Displacements fields have the potential to yield information not present using distance maps. In the bottom of Figures 4-5, 4-6, and 4-7, the derivatives of the classifier based on displacement fields are shown in the form of a vector field. Because distance maps do not use correspondences, it is not possible to show motions tangential to the

surface using distance maps. The vector fields show that there is motion along the surface in several places. Most notably, in Figure 4-6, there is a clear rotation of the tail of the hippocampus. This effect is not clearly visible in the surface changes; it is only visible using the vector field.

4.4 Results: Thalamus Study

We apply our methods to a second study of shape. To our knowledge, no one has searched for morphological differences of the thalamus between schizophrenic and normal individuals. However, as the nuclei in the thalamus in part control and integrate cortical activity, the thalamus may play a role in schizophrenia [PGS+98]. In this section, we present preliminary results of a morphological study of the thalamus in 18 first episode schizophrenic subjects and 18 normal subjects. The results presented here are preliminary because some subjects may be excluded from the study in order to have relatively consistent economic status, education, and socioeconomic status across patients. Also, the manual segmenter who identified the thalami has indicated that she had difficulties locating the borders of the thalamus in some of the subjects due to blurriness in the original image [Wie03]. Those subjects may also be removed from the study.

The subjects in this study did not participate in the study of the amygdala-hippocampus complex [SKJ+92]. However, some of the subjects in this study participated in the study reported in [KSS+03] which examined superior temporal gyrus volumes.

Morphological differences were determined using displacement fields. We report results using an absolute orientation alignment; other alignment yielded nearly identical results. That is, cross validation accuracies differed by less than 3% using the different alignment techniques. Each thalamas was normalized to the same volume. In this study, normalizing volume consistently produced increased classification performance in both linear and non-linear classifiers by between 3 and 8 percentage points.

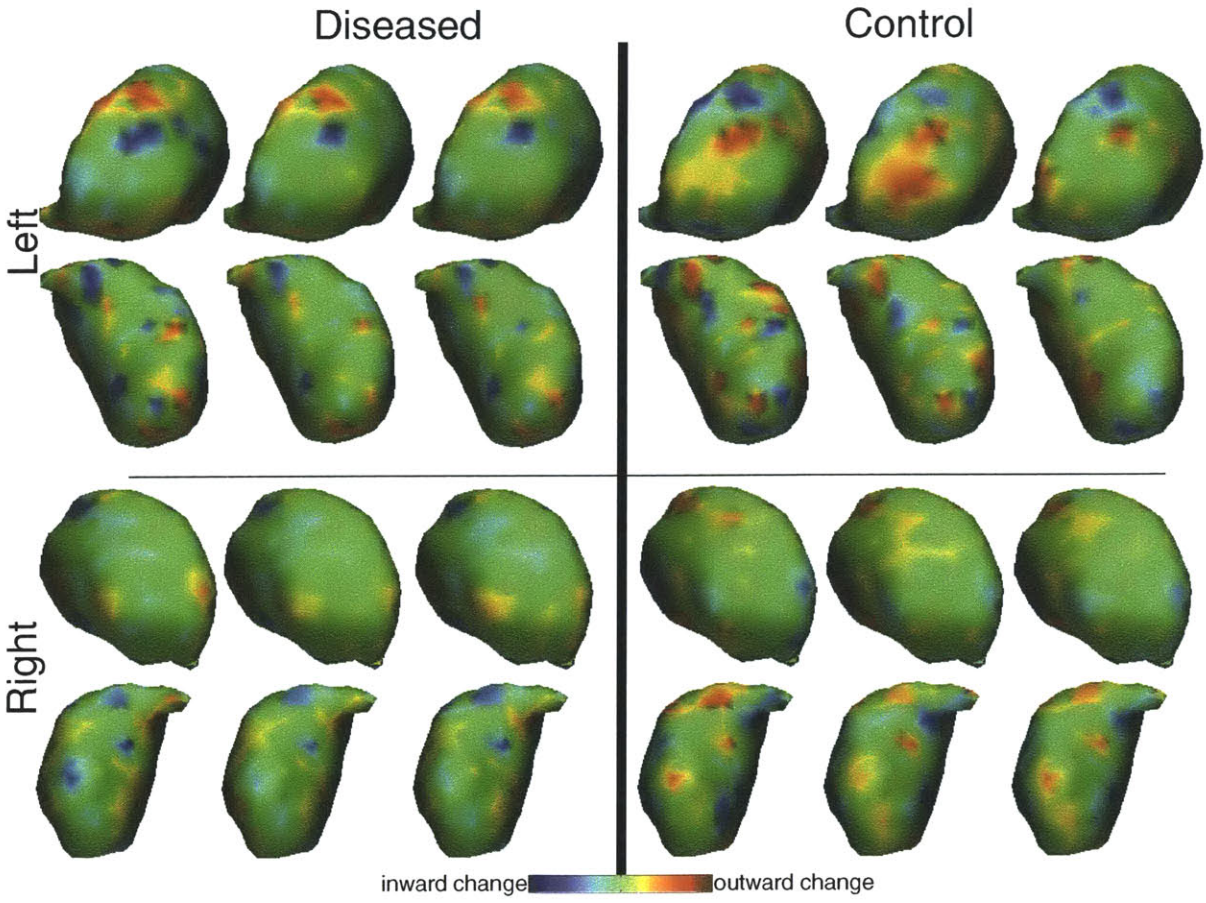


Figure 4-8: The "deformation" of the surfaces relative to the surface normal for the left and right thalamus. Deformation is as described in Figure 4-5. In each quadrant of the plot, the three largest deformation are shown, evaluated at the support vectors of the SVM; the largest vector is leftmost and the third largest is second from the left. The top two rows are for two different views of the left thalamus; the bottom two are for the same views of the right thalamus. An RBF classifier was used with displacement fields to generate the figure. Displacement fields are not shown because the projections of them are difficult to interpret for the thalamus.

Structure	Cross Validation Accuracy	
	Non-Linear Classifier	Linear classifier
Left Thalamus	$73 \pm 15\%$	$55 \pm 19\%$
Right Thalamus	$63 \pm 15\%$	$45 \pm 20\%$
Both Thalami	$71 \pm 15\%$	$55 \pm 19\%$

Table 4.3: Cross-validation accuracy for the thalamus using Linear and non-linear (RBF based) classifiers. The range is the 95% confidence interval.

Table 4.3 shows the cross-validation accuracy of linear and non-linear classifiers on the thalamus. The non-linear classifier was better able to separate left thalami (73% accuracy) than the right thalami (63% accuracy). The performance of the non-linear classifiers were significantly better than the performance of the linear classifiers, which achieved classification accuracies near 50%.

Shape differences are shown for the left and right thalamus in Figure 4-8. Examining the figure, many of the changes between the diseased and normal groups are localized indentations or protrusions, particularly on the inferior portions of the right and left thalamus. As with the amygdala-hippocampus complexes, the changes needed to make one group appear more like the opposite are similar in nature but opposite in sign. Examining gradients at the remaining complexes (not shown), there were strong similarities of the gradients evaluated at all the left complexes and all the right complexes.

4.5 Discussion

We compared classifiers based on displacement fields and distance maps for a shape study of the amygdala-hippocampus as it related to the presence of schizophrenia. It is clear from the similarity matrices in Figure 4-3 that both representations provide some information to separate shapes. However, classifiers based on displacement fields consistently outperformed those based on distance maps, by up to 20 percentage points. In particular, linear classifiers based on distance maps did little better than random while linear classifiers based on displacement fields performed as well as 70% cross-validation accuracy.

The shape differences found using the two different classifiers were visually similar for the right complex, but very different for the left complex. There were two pieces of evidence that suggested that the results based on displacement fields were more likely. First, the linear classifier based on displacement fields yielded very similar results to the non-linear classifier based on displacement fields. Second, examining the derivatives of the classifier based on displacement fields at all of the left complexes yielded similar results, while the derivatives of the classifier based on distance maps did not.

The difference in the answers found by the two classifiers is an indication of one of our main concerns: classifiers created from such small numbers of examples may not generalize to new data[Gol01]. That the displacement field based classifiers found a qualitatively similar class difference evaluated at many complexes may be a good indicator of those classifiers' ability to generalize to other data. Also, the observation that in several cases displacement-field based classifiers worked reasonably well with a simple linear classifier (Section 4.3.1) may also be such an indication.

We not only examined the effects of shape representation on classification accuracy, but we examined the effects of alignment, volume normalization, and classification choice. For the amygdala-hippocampus study, shape representation and classification method produced the largest effects on classification accuracy. That is, non-linear classifiers outperformed linear classifiers by 10 to 20 percentage points. Alignment was second most important, causing a range of plus or minus 5 percentage points. Volume normalization had the smallest effect, approximately 3 percentage points.

In Section 4.3.3, we examined two representations that were independent of global orientation: one based on the volumes of the tetrahedra in a description of an object and a second based on the edge lengths. The relatively good performance of the linear-classifiers based on volume element representations was surprising to us. Because of the lack of directional information in the volumes of the tetrahedra, we expected classifiers based on this representation to perform rather poorly. The success of this method, and its lack of directional information, suggests that a representation

that is independent of global orientation but maintains local orientation information might be very successful. In particular, the representations based on multi-resolution, progressive meshes [GSS99, DGSS99] seem to capture these ideas and are likely worth investigating.

Examining the thalamus study, we were initially skeptical of the shape differences found between the classes. We expected a smoothly varying shape difference, not relatively localized "bumpiness". However, the segmenter who identified the structures has specifically noted that a number of the examples appear bumpy [Wie03]. Furthermore, as the thalamus consists of collection of nuclei, the bumpiness could be an indication that only some nuclei are affected by the presense of disease. Thus, the results may be reasonable. When the study nears its completion, we will examine the data to see if a human can observe the bumpiness in one group and not the other. Currently, the study is blinded as to which anatomical structures come from diseased subjects and which come from normal controls.

4.6 Conclusion

The primary goal of this chapter is to demonstrate that the correspondances developed in the last chapter can yield a performance improvement in a mophological study over over representations. To accomplish this goal, we performed a shape study based on the correspondences and based on a very different representation, distance maps. We found that a classifier based on the shape matching algorithm was able to achieve a cross validation accuracy roughly 15 percent points higher than the classifier based on distance maps. Furthermore, we were able to achieve 70% seperation of the groups using a linear classier based on displacement fields; the classifier based on distance maps achieved accuracies slightly more than 50%. Thus, for the amygdala-hippocampus study, displacement fields outperformed distance maps.

We also examined the sensitivity of classification cross-validation accuracy to several other issues. The most important issue was the choice of classification method: non-linear RBF based methods outperformed linear classifiers by 10 to 20 percentage

points for both studies examined. Volume normalization and alignment method had smaller effects.

Finally, we used our methods to find new results correlating shape differences in the thalamus with first episode schizophrenia. We found statistically significant shape differences in the left and right thalamus and visually showed the differences.

Chapter 5

Application: Segmentation

In the previous chapter, segmented structures were used to correlate shape and disease. Those structures were segmented manually, an extremely tedious, time consuming, and error prone task [Dem02, KSG+92]. For studies involving tens of cases, manual segmentation may be a viable option. However, it is very challenging to use manual segmentation for studies involving hundreds or thousands of cases.

Automatically identifying tissue types in medical images has proven challenging. The challenge stems partly from confounding effects in imaging. MRI images, for example, suffer from noise, and from image inhomogeneities due to varying magnetic fields, as indicated by Figure 5-1. Adding difficulty of the segmentation task is the fact that the boundaries of structures are sometimes of extremely low contrast. In fact, some boundaries are defined by convention rather than intensity differences. For instance in the brain, the border between the plenum temporalis and Heschyl's gyrus is defined by a straight line across the grey matter [KMaJEA+99]. Other boundaries, such as those of the thalamus, are very weakly visible in MRI [PGS+98], as shown in Figure 5-1.

Most automatic segmentation algorithms are able to handle some, but not all, of these confounding issues. For example, intensity based methods have proven successful in correcting for imaging noise and image inhomogeneities [WGKJ96], but have difficulty with barely detectable or invisible boundaries. The addition of spatial information on the location of tissue classes is somewhat helpful to these methods, but

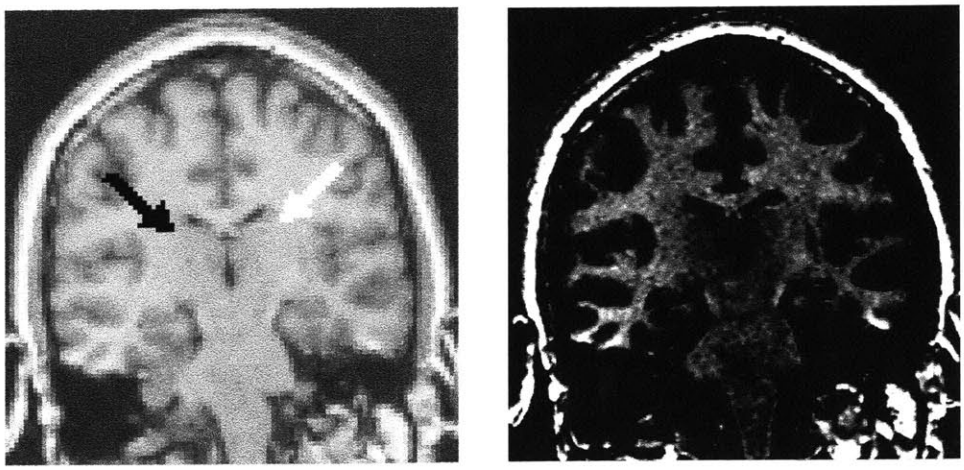


Figure 5-1: Left: MR image with arrows pointing to the right (black arrow) and left (white arrow) thalamus. The borders of these structures are very difficult to identify. Right: the same MR image displayed with a different linear intensity scale. Note that the white matter on the bottom left appears brighter than the remaining white matter, indicating imaging inhomogeneities

as we will show, intensity based methods have difficulties even with a spatial prior. Deformable model based methods use prior shape information to locate boundaries that intensity based methods do not [CHTH92, CET98, KSG99]. However, these methods have trouble with structures that show large variation in shape and size. Structures like gray matter, which can be found reasonably well using intensity based methods, are hard to capture using deformable models.

Both intensity based methods and deformable model based methods have their strengths and weaknesses. We introduce a new segmentation technique that incorporates these two methods to make a more robust and more accurate final method.

In this chapter, we continue to show that the shape matching algorithm given in Chapter 3 finds correspondences that are relevant and useful by using them as part of an automatic segmentation algorithm. We begin by providing a brief overview of existing segmentation methods. We then give a detailed description of the methods we will combine: an intensity based method that uses a spatial prior and a deformable model. To combine the two methods, we create a new non-rigid matching algorithm that adapts a deformable model to the results of an intensity based method. The results of the deformable model are then fed back into the intensity based method

through the spatial prior. By returning the results to the intensity based method, the latter method can incorporate the new information into the entire segmentation. We test our algorithm on MRI images of the brain, using shape information to segment the left and right thalamus. We end the chapter by validating our method, comparing thalamus segmentations of the new algorithm with those provided by a manual segmenter, and with those produced by an intensity based method alone. We show that adding shape information into an intensity based method has significant advantages over such a method alone.

5.1 Overview of Segmentation Techniques

Simple segmentation strategies, such as thresholding, have not produced results that are comparable to expert, manual segmentations of MRI. The medical community has therefore introduced more complicated methods. We divide the strategies used into three types: *intensity based* methods, *spatial distribution* algorithms, and *deformable shape* models.

Intensity based methods use only the voxels' gray values to segment MR images. They often focus on the estimation of intensity inhomogeneity artifacts by using fuzzy clustering techniques [SU00, PP99] or the expectation-maximization (EM) method [WGKJ96]. To decrease the effects of noise and intensity deviations within tissue classes, a Markov Random Field (MRF) can be added [Kap99, ZBS01]. MRFs model the relationship between neighboring voxels; for example, neighboring voxels often belong to the same tissue class.

As intensity based methods use no spatial information, they have difficulties separating out structures of similar intensity, even if these are well separated. To fix this problem, *spatial distribution* methods have been created. These methods use prior spatial distributions on the locations of tissue classes, also known as atlases. They do so by first registering the atlas to the patient and then modifying an intensity based framework to make use of the atlas information. Often they use the expectation-maximization algorithm in conjunction with a Markov Random Field (EM-MRF)

[LMVS99, PWG+02, MVB+02] or with K-nearest neighbor classifiers [WRK+00]. These methods still make large errors in segmenting structures [LMVS99]. As we will show in this chapter, when weakly visible boundaries appear in images, these methods can find unphysical results such as disconnected regions of the same tissue class, small protrusions or sharp edges.

A different class of segmentation methods is based on *deformable shape* models. These models generally incorporate a representation of shape and its allowed modes of variation, along with expected intensity distributions or gradients that the models will overlay. Shape surfaces have been modeled successfully in many different ways, including level sets [LFG00, TYW+01], spherical harmonics [KSG98], point based models [CHTH92, CET98, DCT01] and skeleton or medial representations [PGJA02]. Generally principal component analysis (PCA) is used to find the largest modes of variation across a particular representation [CHTH92]. A generative model is then formed which allows a shape to deform along the principal modes of variation.

Deformable models have been very successful in characterizing many anatomical structures. Using prior information about the shape of tissue classes, they are able to locate boundaries in images that are barely visible, and therefore not typically found by intensity based or spatial distribution methods. But, deformable models are limited by the model used to determine their modes of variation. And, they are limited by their training examples. For structures with tremendous variability, like gray matter, it is unlikely that a small number of training examples will sufficiently well populate the space of gray matter shapes to make a deformable model that accurately captures the variability of that structure. Conversely, intensity based methods capture grey matter reasonable well [WGKJ96].

In this chapter we combine a *spatial distribution* method and a *deformable shape* method to generate a new algorithm that combines the strengths of both methods. In particular, the EM-MRF algorithm generates probabilities that each voxel contains a particular tissue class. We adapt a deformable model to fit these results. The results are then given back to the EM-MRF method so that information from intensity and shape can be merged. In order to feed the shape information back into the EM-MRF

algorithm, we treat the results of the deformable model as a second patient-specific atlas. We then combine the initial atlas and the results of the deformable model, and give the combined atlas back to the EM-MRF algorithm.

5.2 Segmentation Techniques Used

We commence by describing in detail the two methods we will combine. We use a *spatial distribution* algorithm developed by Pohl *et al.* [PWG+02] which has been shown to be successful in segmenting some brain structures. We choose a deformable model that uses many of the ideas developed by Cootes and Taylor [CHTH92], which have also achieved numerous successes.

5.2.1 Atlas Matching

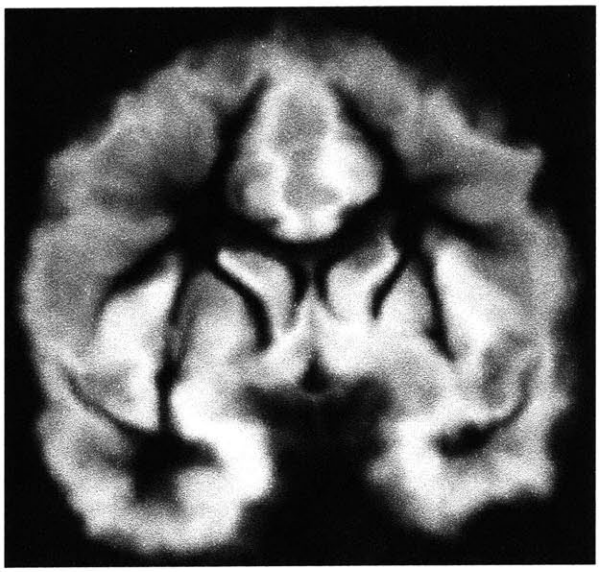


Figure 5-2: Sagittal slice through an atlas of white matter. Note that the atlas does not have sharp boundaries. The atlas is useful for generally locating white matter, but is not useful for accurately describing the boundaries of the structure.

Pohl *et al.* [PWG+02] begin the *spatial distribution* method by warping an atlas onto the image to be segmented. The main purpose of the atlas is to aid an EM-MRF algorithm in differentiating between spatially-separated structures with similar

intensities. To generate the atlas, up to 80 manual segmentations were warped onto a randomly chosen segmentation using a non-rigid registration procedure [WRH+01]. The MRI corresponding to the randomly chosen segmentation is the template case. From these aligned segmentations, the probability of each tissue class at each voxel is estimated. These spatial tissue distributions define the atlas.

The atlas is aligned onto the image to be segmented in order to generate an atlas specific to that image. To achieve this alignment, the template MRI is non-rigidly registered to the image to be segmented [GRAM01, Thi98]. The same warping is then applied to the atlas. The result is an atlas in the coordinate system of the subject of interest. We call the probability determined by the atlas of tissue γ at voxel x $P(\gamma|x)$. As an example of an atlas for a single tissue class, a slice through the probability of white matter is shown in Figure 5-2. A similar slice through the probability for the left-thalamus is shown in Figure 5-5. Note that in both examples, the atlas does not have sharp edges. The atlas is useful in roughly localizing a structure, not in accurately describing its shape.

The atlas has a secondary purpose besides providing localization information. In order to label a particular structure, it is necessary to know the expected intensity distribution of that structure. This intensity distribution will change depending on the particular imager used and imaging modality chosen. One way of obtaining the intensity distribution automatically is to examine the intensities of voxels at locations where the atlas is very confident that a particular tissue class is found. Based on this idea, Pohl [Poh03] estimates the probability $p(\mathcal{Y}|\gamma)$ that intensity $\mathcal{Y}$ comes from tissue class γ by a Gaussian distribution fit to the log intensity of the voxels where the atlas is at least 95% certain that tissue class γ will be found.

5.2.2 Inhomogeneity and Tissue Estimation

The core of the segmentation algorithm consists of an expectation maximization (EM) technique used in conjunction with a Markov random field (EM-MRF). The algorithm iterates between an expectation step (E-step) and a maximization step (M-step). For each tissue class, γ, the E-step computes the probability of the tissue class, $\mathcal{W}(\gamma|x)$,

at each voxel x. These probabilities are generally called *weights* in the EM-MRF framework. The estimate for the next iteration $n+1$ is based on the image intensities $\mathcal{Y}$, the current estimate of additive image inhomogeneities, or bias field, $\mathcal{B}^n$, the probabilities from the atlas $p(\gamma|x)$, and the estimated intensity distribution of each tissue class $p(\mathcal{Y}|\gamma)$. The estimate is also affected by a neighbor energy function $E_N(\gamma, x)$, which reflects the likelihood that two tissue classes occur in neighboring voxels

$$\mathcal{W}^{n+1}(\gamma|x) = \frac{1}{Z} p(\mathcal{Y}(x)|\mathcal{B}^n(x), \gamma) p(\gamma|x) E_N(\gamma, x), \tag{5.1}$$

where Z is a normalization factor. Equation 5.1 effectively calculates the posterior probability of a tissue class at a particular voxel, given the intensity distribution of the tissue class, the probability that a tissue class is assigned to a particular voxel according to the atlas, and the neighbor energies. Because of the neighbor energy function, Equation 5.1 can be viewed as a description of a Markov random field, and solved approximately using the mean-field approximation. Use of this approximation avoids having to calculate the normalization factor Z.

The M-step calculates the maximum a posteriori (MAP) estimate of the additive bias field, $\mathcal{B}$, which models intensity inhomogeneity artifacts. That is, one finds the value of $\mathcal{B}$ which maximally predicts the measured intensities at each voxel given a prior probability on $\mathcal{B}$:

$$\mathcal{B}^{n+1} = \arg\max_{\mathcal{B}} p(\mathcal{B}|\mathcal{Y}, \mathcal{W}^{n+1}) = \arg\max_{\mathcal{B}} p(\mathcal{B}) p(\mathcal{Y}|\mathcal{B}, \mathcal{W}^{n+1}). \tag{5.2}$$

The probability distribution of the additive bias field $p(\mathcal{B})$ is assumed to be a stationary, zero mean Gaussian with a given variance, and covariance across neighboring voxels. The covariance gives the additive bias field smoothness properties. The second term in Equation 5.2 asks how well a given bias field $\mathcal{B}$ and weights $\mathcal{W}$ predict the measured intensities. The solution to this equation can be shown to be [WGKJ96]

$$\mathcal{B}^{n+1} = H \cdot R \tag{5.3}$$

where H is a smoothing filter determined by the covariance of $\mathcal{B}$, and R is the mean residual field of the predicted image in relation to the real image. In words, the additive bias field is roughly a smoothed version of the difference between the predicted image using the weights and the measured image (see [WGKJ96, PWG+02] for more details).

The E- and M- Steps are repeated until the algorithm converges to a local maximum.

5.2.3 Deformable Models

We introduce shape-based information into the EM-MRF framework using a deformable model very similar to the ones presented by Cootes and Taylor [CHTH92]. Deformable model based segmentation methods use prior information on the shape of an object. They typically begin by describing a set of training shapes using a common description. Depending on the choice of shape descriptor, the representation can be very large, often using thousands of parameters to describe the object. However, researchers argue that the majority of the variations in the data can be captured by a few important modes of variation. Principal component analysis is commonly used to find these modes of variation [CHTH92, DHS01]. Given the representations of the examples, PCA finds an mean representation, and a set of orthonormal vectors $\vec{e}_i$, that maximally describe the variation of the training examples about their mean. PCA additionally finds the variances σ_i^2 of the examples in each direction $\vec{e}_i$.

To use the results of PCA as the basis of a segmentation process, the mean model is deformed to agree with an image. The mean model is only allowed to deform in the subspace defined by the vectors $\vec{e}_i$ with largest variance. Choosing enough vectors to account for 98% of the variation in a space is typical. When the model is deforming, the target image voxels inside the final model are identified as the voxels of interest.

The matching process is carried out within a maximum a posteriori (MAP) framework. An MAP estimator finds the representation that maximizes a combination of an agreement term between the model and the image, the prior probability on the representation and pose. The prior probability on the representation comes directly

from PCA, which is equivalent to forming a Gaussian prior on the changes in the model with covariance matrix $\sum_i \vec{e}_i \vec{e}_i^T \sigma_i^2$. The image agreement term can be in the form of the expected intensity that the deformable model will overlap, or a measure of "edge-ness" that the boundaries of the model will overlie. One typically further trains the models to include covariances between image intensity variations and shape, though we will not use these ideas here. Instead, we will fit the deformable model to the image weights $\mathcal{W}$, as we describe in the next section.

5.3 Methods

The main contribution of this chapter is to merge deformable models and spatial distribution methods into one segmentation technique. Our idea is to have a *spatial distribution* based method to capture all known information about intensities and a deformable model to capture all information about shape. The results of the deformable model can then be fed back into the EM-MRF algorithm by updating the atlas. That is, we consider the deformable model as essentially providing patient-specific atlas information with sharp boundaries.

The overall algorithm is characterized by three major steps (see Figure 5-3). First, an atlas representing tissue probability is non-rigidly matched to the image of interest, as discussed in Section 5.2.1. The EM-MRF method is used to estimate the bias field and tissue weights at each voxel, as described in Section 5.2.2. The EM-MRF method therefore captures all intensity information in the image. Finally, a deformable model aligns itself to the tissue class probabilities found by the EM-MRF algorithm, thereby adding shape information. The results of the deformable model are returned to the EM-MRF algorithm by updating the atlas. The latter algorithm can integrate the shape information with other available information. The EM-MRF algorithm iterates, taking shape information from a deformable model in each iteration.

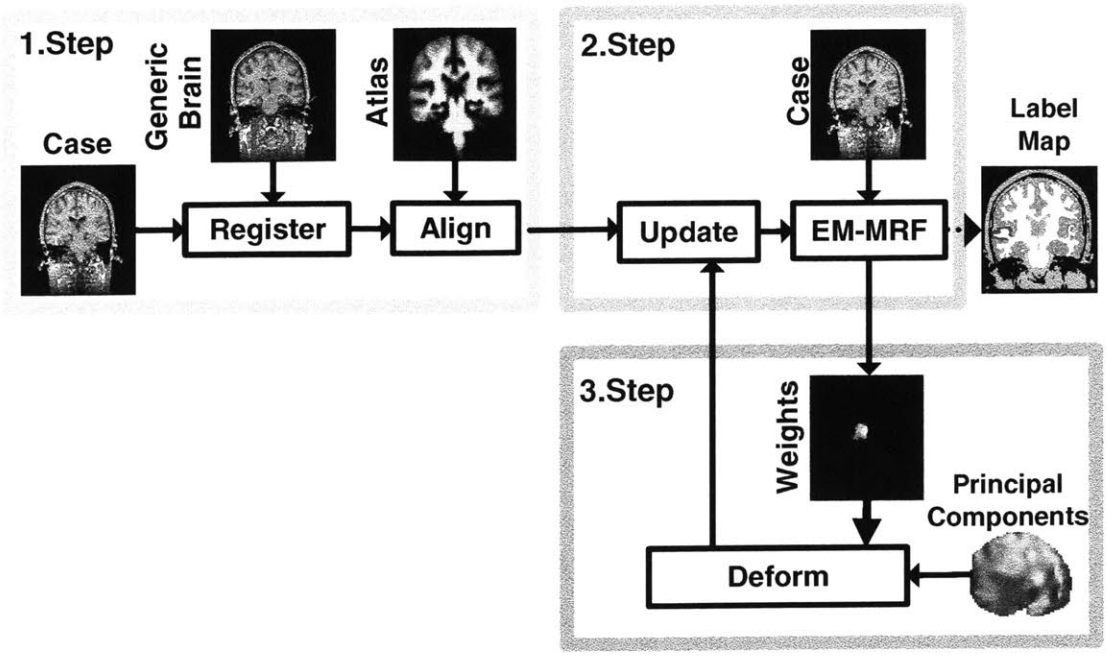


Figure 5-3: The three main steps of the algorithm.(1) Atlas Matching. The atlas is warped to the image of interest. One of the tissue classes in the atlas, white matter, is shown as an example. (2) the EM-MRF algorithm incorporates the atlas, estimates the weights of each tissue class at each voxel, and estimates image inhomogeneities. Bottom: the weights $\mathcal{W}$ of a particular tissue class is fit by a deformable model. The results feed back into the EM-MRF algorithm. (3) A deformable model of the right thalamus is shown. The gray values indicate the first principal component of variation. White indicates expansion of the model and black indicates compression.

We now describe the particulars of our deformable model as well as the details of the feedback of shape.

5.3.1 Deformable Model

In order to integrate the shape constraints into the segmentation process, a deformable model for tissue class γ is defined and matched to the weights $\mathcal{W}(\gamma|x)$. This matching problem is very similar to the one addressed in Chapter 3. The weights $\mathcal{W}(\gamma|x)$ vary between zero and one, just as interpolated binary intensity varies between zero and one in Chapter 3. The model here will deform according to prior information; the one in Chapter 3 deformed according to linear elasticity. There is one major difference between the two non-rigid registration problems: in Chapter 3, the target images

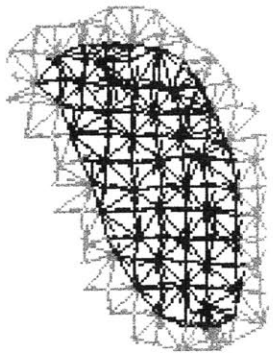


Figure 5-4: A thin slice through a tetrahedral mesh of a right thalamus. The dark tetrahedra are inside the thalamus. The lighter tetrahedra are outside the thalamus. During matching, the dark tetrahedra are attracted to region that are more likely to contain thalamus, the light tetrahedra are attracted to regions less likely to contain thalamus.

had well-defined boundaries, while there is no reason to expect the weights $\mathcal{W}$ to have such boundaries. However, the matching methods developed in Chapter 3 rely on a volumetric image agreement term so that clear boundaries are not needed for those methods to function. We therefore develop similar methods to the ones used in Chapter 3.

We use a separate mesh for each tissue class. While there are potential advantages to representing many different tissue classes with one mesh, there are disadvantages due to topology changes between subjects. For example, some subjects have a right and left thalamus that touch each other; in other subjects these are separated by a thin layer of the ventricles. A mesh that included all of these structures would have difficulty capturing these connectivity changes.

For each tissue class, a mesh is created by filling one randomly-chosen training example with tetrahedra. Tetrahedra are added to the mesh outside the segmented regions as shown in Figure 5-4. Each tetrahedron is labeled as to whether it is inside or outside the tissue class. The mesh is then warped to all the other examples using the techniques of Chapter 3, so that all the training examples are represented by the same mesh. To adjust for differences in patient positioning during imaging, all the meshes are aligned to one coordinate system using the absolute orientation of the mesh nodes [Hor87]. We use PCA on the node coordinates of the mesh. The results

are an average mesh, which will become a deformable model, and $P(\vec{r})$, a Gaussian probability on the displacements of the nodes $\vec{r}$ relative to the average mesh.

The deformable model is deformed onto the weights $\mathcal{W}(\gamma|x)$ using a maximum a posteriori (MAP) matcher. We find the pose $\hat{\mathcal{P}}$ and the displacements of the nodes of the mesh $\hat{\vec{r}}$ that maximize the posterior probability of the pose and displacements

$$\begin{aligned}\left(\hat{\vec{r}}, \hat{\mathcal{P}}\right) &= \arg\max_{\mathcal{P},\vec{r}} \log[P\left(\vec{r}, \mathcal{P} \mid \mathcal{W}(\gamma|x), \text{Mesh}\right)] \\ &= \arg\max_{\vec{r},\mathcal{P}} \log[P(\mathcal{W}(\gamma|x), \text{Mesh}|\vec{r})\ P(\vec{r})\ P(\mathcal{P})]\end{aligned} \tag{5.4}$$

where $P(\mathcal{W}(\gamma|x), \text{Mesh}|\vec{r})$ is the probability that the mesh overlaps the tissue weights, $P(\vec{r})$ is the Gaussian probability determined by PCA, and $P(\mathcal{P})$ is the probability of a given position and orientation of the model. To estimate the overlap probability of the mesh, we begin by using the standard assumption that the probability distribution of the label of each voxel is independent of every other voxel. Then, letting x_{T_j} represent the voxels inside tetrahedron T_j,

$$\log P(\mathcal{W}(\gamma|x), \text{Mesh}|\vec{r}) = \sum_{T_j} \sum_{x \in x_{T_j}} \log P(\mathcal{W}(\gamma|x), T_j|\vec{r}) \tag{5.5}$$

$$= \sum_{T_j} N_{T_j} \left(\frac{1}{N_{T_j}} \sum_{x \in x_{T_j}} \log P(\mathcal{W}(\gamma|x), T_j|\vec{r}) \right) \tag{5.6}$$

$$\approx \sum_{T_j} \frac{V_{T_j}}{V_{voxel}} \langle \log P(\mathcal{W}(\gamma|x), T_j|\vec{r}) \rangle_{T_j} \tag{5.7}$$

where $\langle \cdot \rangle_{T_j}$ represents the continuous average over tetrahedron T_j, V_{T_j} represents the volume of tetrahedron T_j and V_{voxel} is the volume of a voxel. The progression from Equation 5.6 to Equation 5.7 is the change from the discrete (voxels) to continuous space. To make that advance, we have effectively performed the thought experiment of upsampling the image many times. In the present case, there are many voxels within each tetrahedron, so that continuous approximation become appropriate.

The probability that a voxel x with weight $\mathcal{W}(\gamma|x)$ is overlapped by a tetrahedron initially inside the structure is exactly $\mathcal{W}(\gamma|x)$. If only tetrahedra representing the

structure of interest were included in the template, the mesh would have a tendency to collapse to the most likely region (that is, where $\mathcal{W}(\gamma|x)$ is greatest). The tetrahedra in the template that were originally outside the initial example prevent this problem from occurring. These tetrahedra are attracted to regions where $\mathcal{W}(\gamma|x)$ is small. The probability that a voxel with weight $\mathcal{W}(\gamma|x)$ is overlapped by a tetrahedron initially outside the structure is $1 - \mathcal{W}(\gamma|x)$. The image agreement term becomes,

$$\log P(\mathcal{W}, \text{Mesh}|\vec{r}) = \sum_{T_j \in \text{Tet}_{in}} \frac{V_{T_j}}{V_{voxel}} \langle \log \mathcal{W}(\gamma|x) \rangle_{T_j} + \sum_{T_j \in \text{Tet}_{out}} \frac{V_{T_j}}{V_{voxel}} \langle \log(1 - \mathcal{W}(\gamma|x)) \rangle_{T_j}, \tag{5.8}$$

where Tet_{in} and Tet_{out} are the set of tetrahedra labeled inside and outside respectively.

The last term to estimate is the probability of pose. To keep the overall method general, allowing images at any orientation, it is not appropriate to model absolute pose. We therefore currently use a uniform prior on position and orientation. Thus, the MAP estimator becomes

$$\begin{aligned} \left(\hat{\vec{r}}, \hat{\mathcal{P}}\right) &= \arg\max_{\vec{r}} \ \log P(\text{data}|\vec{r}) P(\vec{r}) P(\mathcal{P}) \\ &= \arg\max_{\vec{r}} \sum_{T_j \in \text{Tet}_{in}} \frac{V_{T_j}}{V_{voxel}} \langle \log \mathcal{W}(\gamma|x) \rangle_{T_j} + \sum_{T_j \in \text{Tet}_{out}} \frac{V_{T_j}}{V_{voxel}} \langle \log(1 - \mathcal{W}(\gamma|x)) \rangle_{T_j} \\ &\quad - \sum_i \frac{\vec{r}^T \vec{e}_i \vec{e}_i^T \vec{r}}{2\sigma_i^2} \end{aligned} \tag{5.9}$$

where the last term comes from the $P(\vec{r})$ as determined by PCA, and the nodes of the mesh are only allowed to deform along the principal components of the tetrahedra e_i.

The initial pose of the model can be determined by alignment to the outcome of the EM-MRF algorithm. In each iteration, the pose is updated by treating the model as a rigid object and moving the object in the direction of the torque and the total force due to image forces. After a new estimate a pose is found, a gradient descent algorithm is used to update the displacements of the deformable model.

5.3.2 Error in Weight Estimates

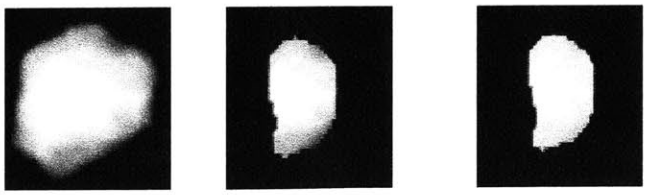
The derivation in the previous section implicitly assumed that the weights $\mathcal{W}$ were accurate. This is not necessarily the case. In fact, we expect the weights to have errors because, if they did not have errors, the EM-MRF algorithm would segment the tissue class of interest with no help from the deformable model.

Note that the objective function is very sensitive to errors. For example, consider a single voxel that has weight 0.1, that should actually be a weight of 0.3. The change in the image agreement term is roughly $ln(0.3) - ln(0.1) = 1.1$. This is the slightly more than the contribution to the objective function by moving the entire mesh one standard deviation in a mode, which is exactly 0.5. Thus, errors in one voxel are sufficient to make a significant difference in the mesh. Errors in several voxels can cause much larger effects.

To define an equation more robust to errors in the estimates of the weights $\mathcal{W}$, one can model errors in the estimates. Alternatively, one can introduce a prior on the probability of the weights. Both methods have the same effect. We assume a spatially stationary prior probability of 0.5 that a voxel will be assigned to the tissue class of interest. We can then form a convex combination of the two estimates of the weights of tissue at each voxel; we use a weighting of 0.5 for each estimate of weight. In practice, this means mapping $\mathcal{W}$ from the range 0 to 1, to the range 0.25 to 0.75. As the image agreement term is most sensitive to errors in probability near 0 and 1, this type of prior should mitigate the problems due to errors in the estimates of the weights.

5.3.3 Feedback of Shape Information

The deformable model returns a binary map of 1's inside the estimate of the shape and 0's outside. This information is essentially a very sharply defined patient-specific atlas. We therefore use this information to update the prior probability of the tissue class, $P(\gamma|x)$, which originated in the atlas. In particular, we update the atlas by



Original First Iteration Second Iteration

Figure 5-5: Changes in the atlas of the left thalamus through three iterations with $\lambda = 0.5$. From left to right: the initial atlas is somewhat diffuse. The updated atlas slowly converges toward a binary map. Black indicates 0 tissue probability; white indicates probability 1. After each iteration, the atlas becomes more sharply defined and patient-specific.

taking a convex combination of the atlas and the results of the deformable model:

$$p(\gamma|x) \leftarrow (1 - \lambda) \cdot p(\gamma|x) + \lambda \cdot S(x) \tag{5.10}$$

where $\lambda \in [0, 1]$ is a weight factor and $S(x)$ is the binary map returned from step three. The influence of the deformable model is defined by λ. It roughly states how confident we are in the deformable model. For shapes like gray matter that are difficult to model, λ should be set near 0. For shapes models in which we have high confidence, λ should be set near 1. Currently λ is set manually by trial and error on a few cases. For the cases we have examined using shape information with the thalamus, the algorithm returns nearly identical results for a sizable range of λ values, so we have found its exact value is not critical. For the thalamus, we use $\lambda = 0.5$. As shown in Figure 5-5, the spatial probability distribution of the atlas converges toward a binary map describing the tissue class region.

5.4 Experiments

The gray scale images tested were acquired using a standard SPGR MRI acquisition system. Each image has 256x256x234 voxels with 0.9375x0.9375x1.5 millimeter voxel spacing. In each image, we segmented skin, cerebrospinal fluid (including the ventricles), gray matter, white matter and the left and right ventricles. We use shape information for the right and left thalamus.

We used fifty-five manual segmentations of the thalami for training the deformable model and for validating the method. We used a leave-one-out validation method so that the manual segmentation of a case was not used to train the deformable model used to segment that case. Manual segmentations by one segmenter were used as a gold standard. Since manual segmenters are error prone [KSG+92], we would ideally have preferred not to use manual segmentations as a standard. However, for the thalamus, manual segmentation is the best gold standard we have.

5.5 Results

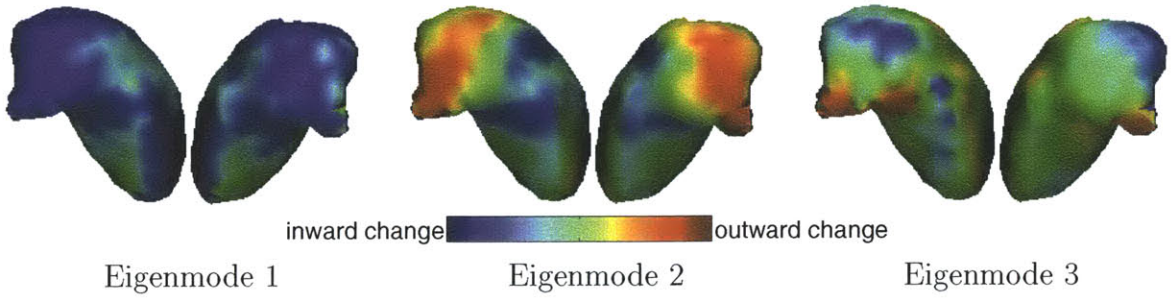


Figure 5-6: The three largest principal modes of variation found using PCA for the left thalamus and the three largest principal modes for the right thalamus. Colors indicate changes to the structure relative to the surface normal. Note that modes were found separately for the right and left thalamus. It is remarkable how similar the first three principal modes of the left thalamus are to the first three principal modes of the right thalamus.

Deformable models of the thalamus were trained on 54 right and 54 left thalami. After representing each thalamus with tetrahedra, the principal components of variation of the nodes of the meshes were found. The first three modes of variation of the right and left thalami are shown in Figure 5-6. It is remarkable that the first three modes of both sides are very similar. They are roughly an expansion/compression mode, a rotation of the anterior portion of the thalamus, and motion of the lateral geniculate (the hook-shaped object). To represent 98% of the variation of the right model (Figure 5-7) required 41 modes; the left required 38 modes. The fact that so many modes were required to describe the variations of 53 examples is a good indication that the variations in the model are not well-modeled by a Gaussian distribution.

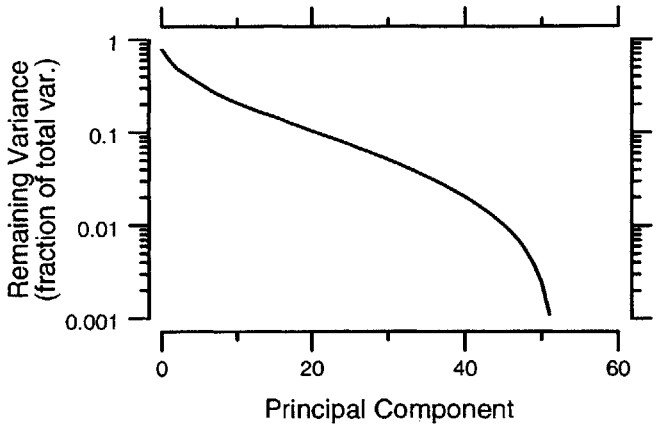


Figure 5-7: The fraction of the total variance of the space covered by the first n principal components of the right thalamus. One typically uses enough components to describe 98% of the space, which is 41 in this example.

5.5.1 Algorithm Sensitivities

We begin by examining one case that presents typical challenges to the alignment process. Figure 5-8 illustrates such a case. The purple outline indicates the gold standard boundaries, as determined by a manual segmenter. The gray scale image represents the weight $\mathcal{W}(\gamma|x)$ that thalamus tissue is present. One can see that there are several regions inside the gold standard boundary that have very low probability. Those weights are roughly 0.02. An examination of the data shows that the EM-MRF algorithm has underestimated the intensities that should be inside the thalamus. Furthermore, the nearest neighbor energy had the effect of making the weights change from almost 1 to close to 0 in a voxel or less.

This type of problem is not one that deformable modeling technologies were designed to address. In particular, deformable models are generally used when part of the boundary is well defined, and part of the boundary is not. Figure 5-8 shows sharp boundaries around the entirety of the weights, $\mathcal{W}(\gamma|x)$. For the matcher to work well in this type of example, the "correct boundary" found by the weights in the top right of the image should be better defined than the "incorrect" one defined by the weights in the lower left of the image. If this were the case, the objective function would have been higher aligning a boundary of the model to the top right of the image, then aligning the boundary of the model to the lower left of the image. However, both

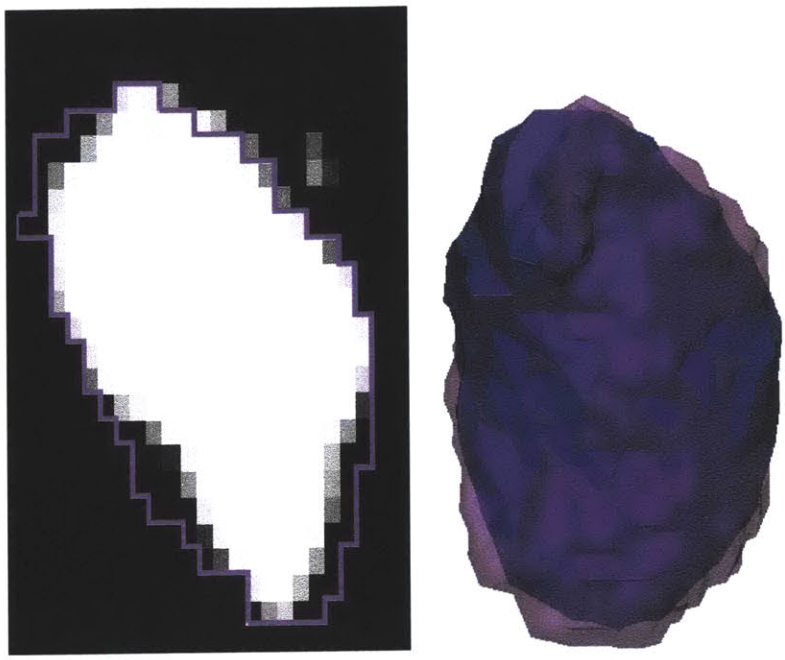


Figure 5-8: Illustration of the difficulty of the matching problem. Left: The purple line indicates the boundary of the gold standard manual segmentation for the left thalamus for a particular image. The image in the background shows the weights sent to the deformable matcher. White indicates weight close to 1; black indicates weight near 0. Surveying the "black" regions inside the border of the gold standard shows that the average weight in those regions is roughly 0.02. The average weight in the regions just outside the model are roughly the same. Thus the boundaries of the weights are roughly equally well defined on both sides. Right: the resulting error in the deformable model is mostly a pose error. The deformable model (blue) ends up slightly rotated to the gold standard (purple). That is, in the image, the lateral geniculate (hook shape) of the deformable model is to the left of the gold standard hook shape; the bottom of the deformable model is slightly to the right of the bottom of the gold standard. The two errors are indicative of a rotation of deformable model relative to the gold standard.

boundaries are equally well defined. Thus, the result is an error in the estimate of the thalamus. In particular, most of the error was an error in the estimate of pose, as shown by the right side of Figure 5-8.

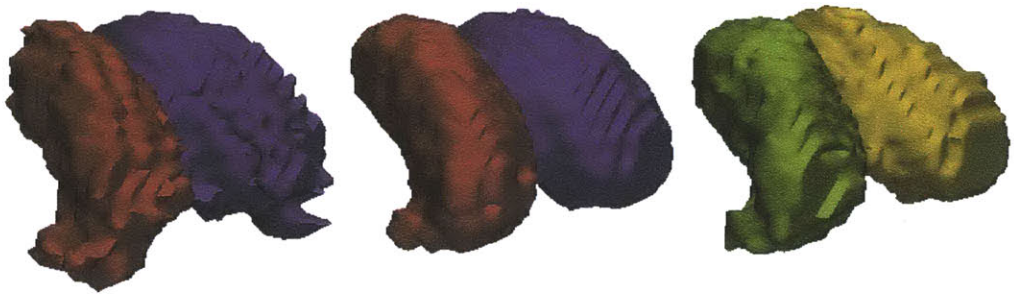
5.5.2 Segmentation without a Shape Prior

We begin by segmenting using the EM-MRF algorithm alone in order to create a baseline with which to compare our algorithm. For the EM-MRF algorithm, we effectively set λ to zero in Equation 5.10. In several of the cases we examined, the algorithm found a primary region of right thalamus, and one or more secondary regions detached from the primary region. An example of such a detached region is shown in the top right of Figure 5-10. Figure 5-10 and Figure 5-9 also shows that the EM-MRF algorithm resulted in jagged edges, and protrusions coming out of the surface. As we expect the surface of the thalamus to be fairly smooth, these surface features are unlikely.

The method made a large overestimate of the posterior portion of the thalamus (top of the image) as shown in the three dimensional surface model in Figure 5-10. It also overestimated the other end (anterior) of the thalamus, which is somewhat visible in the top portion of the 3D model of Figure 5-10. For that segmentation, the intensity based method found a thalamus several voxels longer than the manual segmentation. These results are consistent with the observations by Pohl *et al.* [PWG+02] who noted that EM-MRF algorithms tend to make large over or under estimates of the "ends" of small structures.

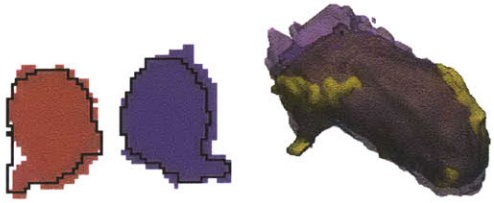
5.5.3 Segmentation with a Shape Prior

We segmented the brain using our new algorithm which includes shape information for the right and left thalami. Thus, after each iteration, the atlas was updated by the deformable model as shown in Figure 5-5. Unlike the EM-MRF algorithm, in every case the new algorithm found only one connected region of left thalamus and only one connected region of right thalamus. As shown in Figure 5-11 and Figure 5-9,



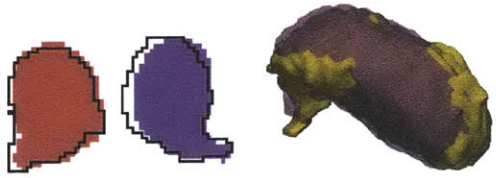
(a) no shape prior (b) using shape prior (c) manual segmentation

Figure 5-9: 3D models of the right (maroon) and left (violet) thalamus generated by (a) the segmentation algorithm without shape priors, (b) the new algorithm with shape priors, and (c) a manual segmentation. Note that the thalami produced with a shape prior (b) are significantly smoother than those produced without a shape prior, which have protrusion and sharp edges.



Without Shape Prior

Figure 5-10: Comparison between the automatic segmentation with no shape prior and the experts segmentation. The left image shows the right (maroon) and left (violet) automatic segmentations of the thalamus. The black line shows the border of the manual segmentation. Note that the automatic segmentation is sharp protrusions and a jagged surface. The right image shows a 3D model of the left thalamus (maroon) overlapping the manual segmentation (yellow). The automatic segmentation is somewhat translucent so that one may see the overshoot at the top of the image from the goal standard. Note also the jagged surface structure and the separated secondary region of thalamus in the top of the image.



With Shape Prior

Figure 5-11: The same as Figure 5-10 except using the new algorithm that incorporates shape. Note that surface of the result is much smoother than the manual one. The left thalamus (violet) is somewhat rotated from the goal standard position indicative of a pose estimation error in the deformable model. The 3D model produced by the new algorithm better matches the manual segmentation than the 3D model generated without the shape prior in Figure 5-10.

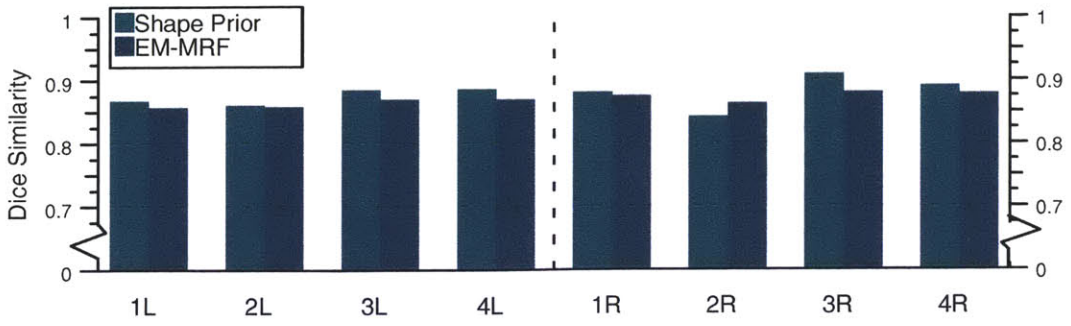


Figure 5-12: The Dice similarity metric for the left and right thalamus of four test cases compared to the gold-standards. For the Dice similarity measure, larger is better. The Dice similarity measure is generally slightly larger for the shape based algorithm than for the EM-MRF algorithm.

the resulting segmentations were smooth, which matches our expectation that the thalamus should be at least somewhat smooth. In fact, the resulting segmentations were smoother than the gold-standard manual segmentations.

Visually inspecting the "ends" of the thalami in Figure 5-11, the new method did well locating the lateral geniculate. It also did better than the EM-MRF algorithm at locating the other end of the thalamus (not visible in the image). Resulting segmentations were generally smoother that the manual or EM-MRF algorithm. The 3D models are shown in the center of Figure 5-9.

As described in Section 5.5.1, the largest errors made by the new method generally consistently of small errors in pose. One example of such an error is visible in Figure 5-11 for the left thalamus; the segmentation appears somewhat rotated and translated from the gold-standard.

5.5.4 Validation

Producing expected smooth results is only one important goal of a segmentation process; a second goal is producing results similar to manual segmenters. We validate the results of our algorithm by comparing the automatic segmentations to the manual segmentations using 2 metrics. The first metric is the Dice similarity coefficient which is the ratio $\frac{2|V_1 \cap V_2|}{|V_1|+|V_2|}$, where V_1 and V_2 are the two segmented regions and $|\cdot|$ indicates volume. The measure is 0 for non-overlapping regions and 1 for regions that perfectly

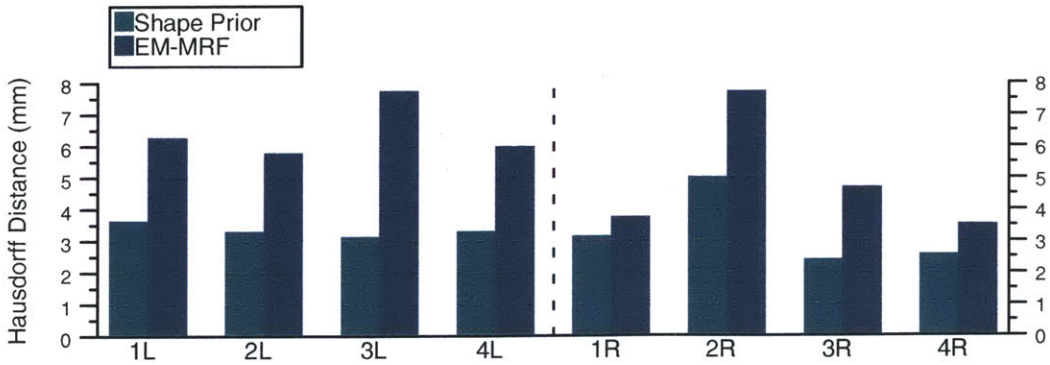


Figure 5-13: The Hausdorff distance for the left and right thalamus of four test cases compared to the gold-standards.. For the Hausdorff distance, smaller is better. The Hausdorff distance is generally significantly smaller for the shape based method than the EM-MRF algorithm.

overlap. The Dice Coefficient is a measure of general region agreement. The second measure we use is the symmetric Hausdorff distance, which is the maximum distance between the two surfaces of the segmentations. This metric is generally useful for finding outliers in the segmentation.

Figure 5-12 and Figure 5-13 show the measures for four cases. The Dice metric is roughly uniform across all subjects for both algorithms. Both algorithms yielded dice similarity coefficients of about 0.87; though the algorithm that used shape prior generally had a slightly larger Dice metric. While the new algorithm slightly outperformed the EM-MRF method using a Dice metric, the new method dramatically outperformed the EM-MRF algorithm using the Hausdorff metric. In many thalami, the Hausdorff metric was significantly smaller for the new shape based algorithm than for the EM-MRF algorithm, sometimes by as much as a factor of two. Qualitatively, the results suggests that the shape-based algorithm overlayed the manual segmentations slightly better than the EM-MRF based algorithm, and the largest errors made by the shape-based method were much smaller than the largest errors made by the EM-MRF method.

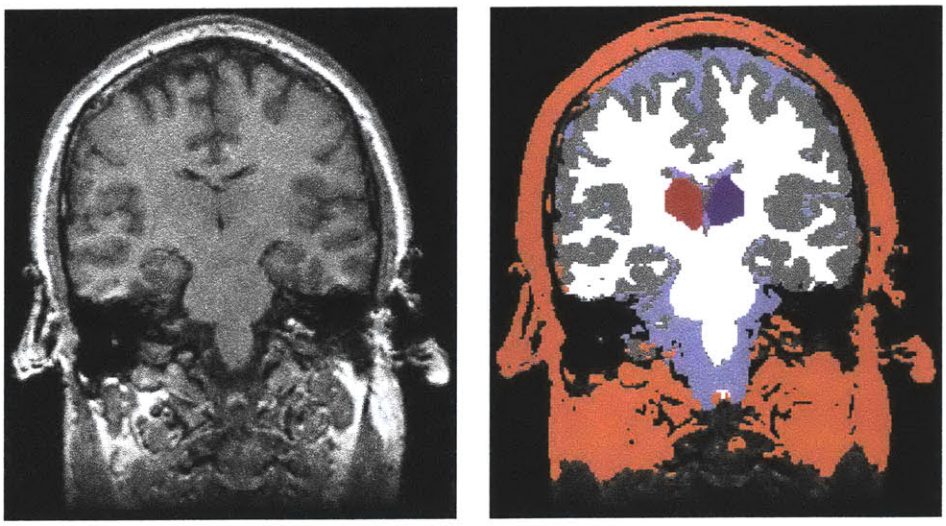


Figure 5-14: A slice through the full brain segmentation using the new algorithm with shape prior. White matter is labeled white. Gray matter is labeled gray. Cerebrospinal fluid is labeled blue. Maroon and Purple are the thalami. Pink is skin.

5.6 Discussion

There are numerous small structures in the brain whose boundaries are either weakly determined or not visible at all. The thalamus is an example of such a structure. Intensity based methods have difficulties finding the boundaries of such structures. As we have shown, such methods can produce protrusions, jagged surfaces, and even disconnected regions of tissue.

We introduced a new algorithm that is a merger of shape information information and intensity based methods. The new method produced significantly smoother surfaces than the old method, without protrusions and without disconnected regions of tissue – the surfaces produced by the new method were much closer to our expectations than the surfaces produced by the method without shape information. According to the Dice measure the volume overlap of the manual segmentations with the segmentations produced with this algorithm was slightly more than the purely intensity based EM-MRF algorithm. Furthermore, the Hausdorff distance clearly showed that the surfaces produced with our new method were significantly closer to the surfaces of the manual segmentation that those of the EM-MRF algorithm.

We identified errors in pose estimates as the largest source of error for the new

algorithm that used shape information. In particular, given the errors in the weights of tissues, $\mathcal{W}$, it was difficult to accurately estimate a pose. One way to significantly mitigate this problem is to add more information by considering adjacent structures. Rather than having a deformable model that consists of tetrahedra that are "thalamus" and "not thalamus", we can re-label the "not thalamus" tetrahedra by a probability distribution over what structures they should overlay. Some parts of the thalamus are always directly next to ventricles; others overlay gray matter. It should be straight forward to train a model to account for this information. Unfortunately, the data on which we trained the deformable model only contained thalamus segmentations. Thus, we were not able to pursue this idea here.

It is worth noting that the idea of using prior models on relative location is related to work by Kapur [KGWK98]. Kapur used histograms of the distances between structures to aid in the segmentation of structures. The distances she used were not directional. Our suggestion is effectively to use directional distances, which can be represented in a straightforward way using tetrahedra.

While we compared our method to an EM-MRF based method, we did not compare our method to "purely" deformable model based method. We have no reason to expect that our current method would outperform a deformable model based method. However, our method has two important advantages over a purely deformable model based method. First, if the image to be segmented were changed from MRI to CT, our methods would still function, starting with atlas registration and proceeding. A deformable model that directly incorporated intensity information would need to be retrained on the image intensities in the new image. A second advantage of our method over a pure deformable model is that our method allows feedback of the results of the deformable model into the EM-MRF framework so that other structures can benefit from the information found using shape.

5.7 Conclusion

We have introduced a new segmentation method based on a merger of deformable models and EM-MRF based algorithms. We have applied that method to full brain segmentations, using shape information to help identify the thalami. Unlike the EM-MRF algorithm we compared it to, the new algorithm using shape produces smooth surfaces, and connected regions of tissue. The new method also produces segmentations of the thalamus with larger Dice similarity coefficients with the gold-standard manual segmentations than the EM-MRF method, and much smaller Hausdorff Distances to those segmentations.

Chapter 6

Non-Rigid Registration of Medical Images

In this thesis, we have argued that using a representation that conforms to anatomy will facilitate the creation of efficient and effective algorithms. In each of the previous chapters, we began by conforming tetrahedral representations to already segmented anatomical structures. In this chapter, we assume that such segmentations are not available; instead, we develop methods that automatically adapt to the anatomy in an image.

In this chapter, we are primarily concerned with non-rigidly registering pre-operative images to intra-operative images. In the context of image guided surgery, intra-operative images give surgeons updated information of the locations of key anatomical structures. Intra-operative images are especially useful when tissue has deformed significantly between pre-operative imaging and actual surgery [NBG+01] – in this case pre-operative images no longer accurately represent the surgical situation. The disadvantage of using intra-operative images is that surgeons are typically unwilling to delay surgery to annotate these images with information such as surgical paths. This information is more easily added to pre-operative images because of the relative lack of time constraints. Furthermore many imaging modalities, such as diffusion tensor imaging (DTI) and function magnetic resonance imaging (fMRI), can be acquired before surgery, but are generally not available during surgery. Surgeons could

benefit greatly from an automatic procedure that warps pre-operative data onto intra-operative data, so that information available before surgery can be transformed to the current surgical situation [FWN+00].

Unfortunately, many medical image non-rigid registration tools take several hours to run, making them unsuitable for use during surgery. Hence, we aim to develop a fast, intensity-based, non-rigid registration technique that can map pre-operative data onto intra-operative data in a few minutes. To accomplish that goal, we develop a non-rigid registration algorithm that automatically adapts to anatomy and to the displacement field between images. Our key contribution is a method whose dominant computation cost need not scale linearly with the number of voxels in an image. Thus, even though medical images can easily have 256x256x124 (or roughly 8 million) voxels, for many non-rigid registration tasks, our algorithm can finish in a few minutes.

We begin this chapter by describing the key concepts that we will use to create the adaptive algorithm. We then review some of the research in the field of non-rigid registration, and describe the similarities and differences between our methods and others' methods. We describe the details of our method, paying special attention to the adaptive representation that we apply to non-rigid registration. We then examine the results of the algorithm and conclude by examining lessons learned.

6.1 Key Concepts

Our key contribution in this chapter is the development of a non-rigid registration algorithm whose dominant computational cost does not necessarily scale linearly with the number of voxels in an image. To accomplish that goal, we create compact, adaptive representations of displacement fields. There are two key concepts that we will use to create such representations. The first is that displacement fields between images need not be represented at the same resolution everywhere. The second is that in an image, the information most useful for non-rigid registration is generally not uniformly distributed; it is generally concentrated at the boundaries of anatomical structures.



Figure 6-1: The red box on the left can be matched to the region on the right. The edge in the first image very likely corresponds to the edge in the second image. However, the green box on the left is in a uniformly white region. It is clear that it matches somewhere in the white region on the right, but finding the exact correspondence is difficult. In particular, the green box might correlate best to the target position shown in the right image, but that correlation is only marginally higher than the correlation to surrounding regions.

We begin by noting that displacement fields between pre-operative images and intra-operative images are generally smooth[1] [GMVS98, HRS+99, RHS+98, WS98], and slowly varying in large portions of the image. For example, a cursory review of brain warping papers suggests that displacement fields often change very quickly near an incision, but much more slowly far away it [FMNW00, HRS+99, MPH+00, SD99]. Because displacement fields are generally smooth, regions of slowly varying displacements can be accurately described using a small number of vectors that are interpolated within those regions. Thus, one way to create a compact representation of a displacement field is to identify regions of slowly-varying displacements and to represent those regions with as few vectors as possible.

While our first observation suggests the density of vectors required to represent a displacement field, our second observation suggest where to make measurements of the displacement field. The second observation is that the information in an image that is most useful for matching is generally concentrated in regions of large intensity variations. One would expect that by finding the displacement field in those regions, one could accurately capture the majority of the displacement field, which could

[1]Deformation fields are often assumed to be smooth because tissue is generally well connected within and across anatomical structures. There are situations where slipping is known to occur; these situations can be dealt with explicitly such as in [MSH+02].

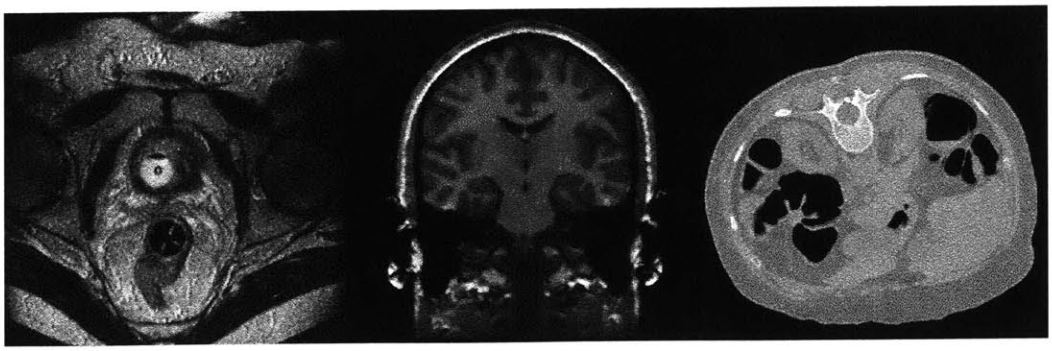


Figure 6-2: From left to right: Axial MRI of male pelvis, coronal MRI of a brain, and axial chest CT. These three images are examples of medical data appearing nearly uniform over large regions. In the pelvis, the muscle and fat has roughly uniform intensity or texture. The white matter in the brain also has nearly uniform intensity. In the CT image, the interior of the colon (in black), the liver (lower right), as well as other soft tissue all appear nearly uniform. In these regions it is difficult to accurately non-rigidly match between subjects.

later be refined by adding information from less information rich regions. Thus, we advocate a hierarchical scheme based on image intensity variations, rather than based on image resolution, as is normally done [DS88].

To quantify this concept mathematically, we build a non-rigid registration algorithm using a combination of a linear elastic energy and an image agreement term. In regions of nearly uniform intensity, correspondences found between one image and its target are known with very little confidence, as demonstrated in Figure 6-1. That is, a point found to correspond in such a region will match that correspondence marginally better than voxels close by. Therefore, image derived forces that drive a non-rigid matcher will be relatively weak in relation to elastic forces. The resulting displacement fields in such regions will therefore be smooth, so that an efficient representation would represent such regions with a small number of vectors.

In previous chapters, we have discussed the importance and usefulness of using representations that adapt themselves to anatomy. Our methods here effectively continue to use the same ideas. In particular, anatomical structures are often imaged as nearly uniform intensity (or uniform texture) regions, as shown in Figure 6-2. Regions of maximal intensity variations are often the borders between anatomical structures. Thus, representing the displacements in regions of the highest intensity

variations is often equivalent to representing the displacements of the borders of anatomical features.

To take advantage of our observations, we develop a representation of a displacement field using the nodes of a mesh of tetrahedra. The nodes of the tetrahedral mesh can be created using different densities in different parts of the image, and selectively adapted as needed. Furthermore, the surfaces of the tetrahedra can be aligned to the surfaces of anatomical structures in the image. If segmentations are available, it is straightforward to include the surfaces of such regions directly into the mesh. If segmented regions are not available, the nodes of the mesh can be moved to places of large intensity variations where information for non-rigid matching is maximized.

Our resulting algorithm therefore uses an adaptive mesh that adapts both to the anatomy of a structure and to the displacement field. In the event that the deformation field is only varying in a small part of an image, the algorithm will concentrate its attention in that part of the image. Thus, the primary determination of computation time will not be the number of voxels in the image because most of the image will not be used. Instead, the computation time be controlled by the number of nodes used in the mesh.

6.1.1 Previous Work

In this chapter, we are primarily concerned with representation. We therefore categorize previous work in non-rigid registration according to choice of representation. For example, there have been a number of grayscale based non-rigid registration methods that use a voxel based representation of a deformation field [CHF01, HNW+99, Thi98, WRD+98]. Another useful representation are b-splines, which typically place one control point at every voxel [RHS+98]. The work of Schnabel *et al.* [SRQ+01] using b-splines is particularly relevant to this paper. Schnabel used hierarchical b-splines to represent deformation fields. Schnabel was able to significantly reduce the number of control points needed by only selecting control points in regions of the image with large intensity variations. (The deformation field was represented at every voxel in the image.)

Our work is partially motivated by the successes of surface-based matching [SD99, MPH+00, FWN+00] and combined surface and grayscale based matching [HHCS+02, SD02]. In particular, Ferrant *et al.* [FWN+00] match surfaces of segmented regions and then interpolate the interior. The success of the method suggests that by accurately measuring the displacements of a few points on surfaces, the remaining displacement field can be interpolated.

Also relevant are the methods developed by Krysl, Grinspun and Schroder [GKS02, KGS01]. They demonstrated the use of hierarchical tetrahedral meshes to interpolate a deformation field of a surface into the interior. In this method, refinement is entirely dependent on the deformation field. Tetrahedra with a large stress-strain integral (high elastic energy) are refined.

We view our work as a combination of the ideas of Krysl *et al.* and Schnabel *et al.*. We combine the power of adaptive multi-resolution tetrahedral methods to minimally represent a deformation field, with the concepts of finding a deformation field using regions of the data with large intensity variations.

6.2 Methods

We begin by developing an initial configuration of tetrahedra to represent the displacement field. We then develop a multi-resolution description of a vector field that adapts to the initial representation. We use this representation to simultaneously find a displacement field and adapt the representation to it. Finally, we create methods to determine the displacement field given a particular tetrahedral mesh.

6.2.1 Initial Representation

We form a very coarse mesh by dividing space into cubes and subdividing each cube into five tetrahedra. We then subdivide each tetrahedron into 8 tetrahedra (The reason for this step will be explained in the following section.). After the mesh is created, we remove tetrahedra that entirely overlay the background of an image. We design the final mesh to have a roughly 5 mm spacing between nodes; we have found

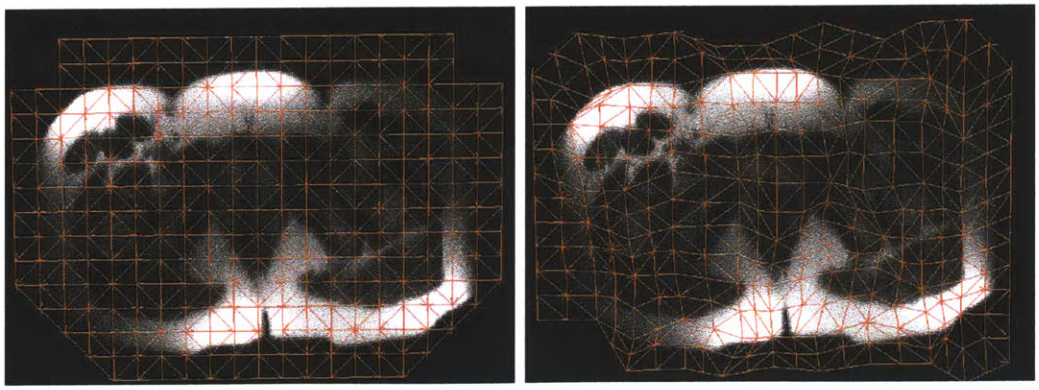


Figure 6-3: Left: a thin slice through an initial mesh overlayed on an MRI of a female pelvis. Right: the mesh, after being adapted to the surfaces of the pelvis. Note that many of the nodes lie on surfaces of the image.

this spacing is generally sufficient so that the algorithm will not overlook a localized displacement that occurs entirely within a tetrahedron.

When surfaces of anatomical objects have been found through segmentation, those surfaces can be incorporated directly into the tetrahedral mesh through intersection, and smoothing as described in Chapter 2. More often those surfaces will not be available. In that case, we locally move nodes to surfaces. For each node, we define a small box around it within which we move that node. The half-width of the box is set to one third the distance to a nearest node in each direction. We chose that distance so that the resulting tetrahedra are not overly skewed. Within that box, we find the voxel with the highest local intensity variations around it, where intensity variation is measured using local image entropies. (Local intensity variance is an alternative measure; our experience is that both measures work well for mesh adaptation.) We then move the node to that voxel. The positions of the nodes of the mesh end up at points which we expect will be maximally useful for non-rigid registration. In effect, we have written a simplistic surface finder. A slice through an example mesh is shown in the right of Figure 6-3. Note that many of the nodes in the mesh lie exactly on surfaces in the image.

The resulting mesh has properties that should be important for a good representation. In particular, the mesh covers the entirety of the image with nearly uniform spacing. And, most of the nodes are in regions of high intensity variation.

6.2.2 Adaptive Representation of a Vector Field

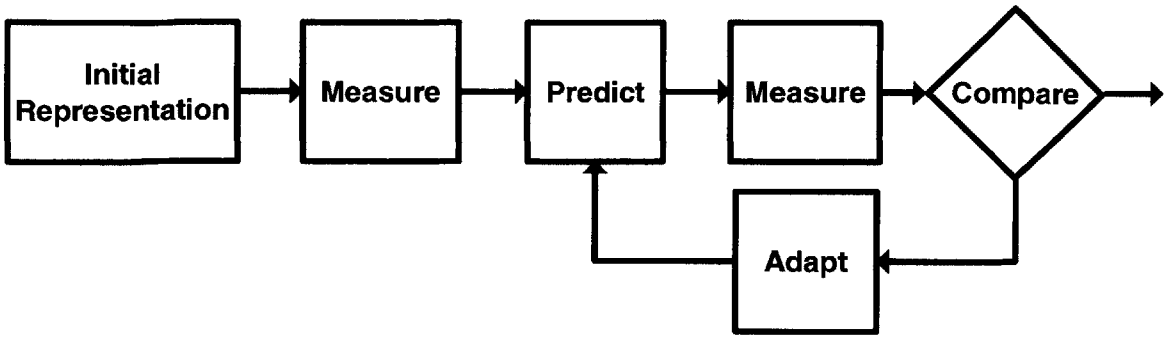


Figure 6-4: The schematic of the adaptive representation.

Our goal is to find the displacement field, while avoiding over-describing it. Our method for accomplishing this goal is described schematically in Figure 6-4, and pictorially in Figure 6-5. The method consists of three key steps: *Predict*, *Measure* and *Adapt*.

We initialize the representation of the displacement field with the initial tetrahedral representation developed in the last section. We then assume that a function is available that will take the nodes of the tetrahedral mesh and sample (or measure) a vector field at the nodes of that representation – we describe such a function in the next section. Calling that function, we measure an initial vector field.

In the *Predict* step, we introduce new nodes into the representation by subdividing the initial representation. We add nodes to the edges of each tetrahedron as shown in the second line of Figure 6-5. This subdivision corresponds to dividing a tetrahedron into eight (octasection), as described in Chapter 2. We then linearly interpolate the measured vectors to the new nodes, as a prediction of what the vectors will be at the new nodes. We have chosen a linear interpolation scheme because linear interpolation is particularly straightforward within a tetrahedron.

In the *Measure* step, we call the function that measures the vector field at the nodes of the subdivided mesh. We can then compare the measured vectors to the predicted vectors. If the vector on an edge agrees with the measured vector to within a certain tolerance, then subdividing the edge did not yield any new information about the displacement field. In the *Adapt* step, we remove all subdivisions that were

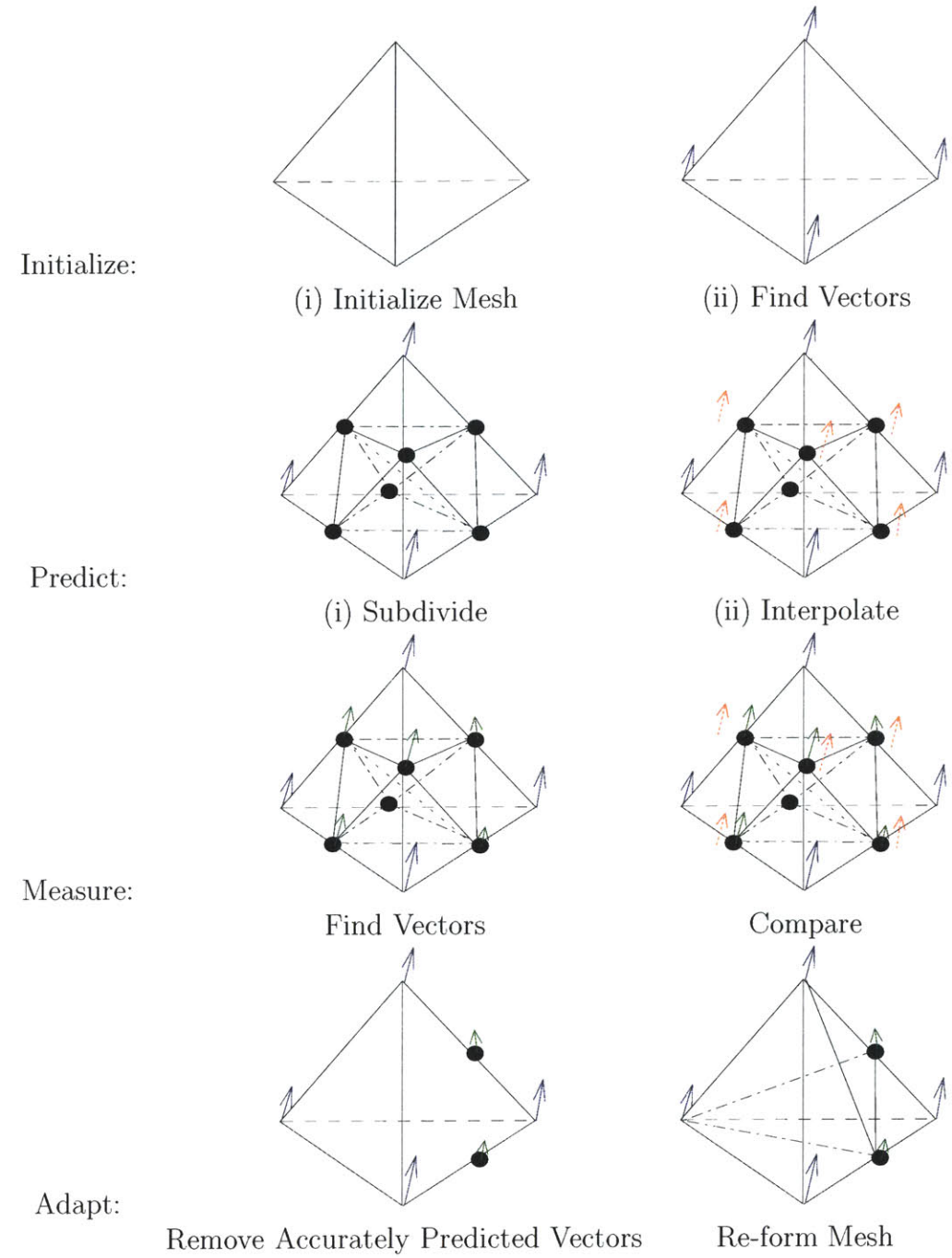


Figure 6-5: The adaptive representation in pictures. Top: the initial mesh is formed and vectors found. Predict Step: the mesh is subdivided and vectors predicted for the new nodes. Measure: the actual vector field is measured and compared to the prediction. Adapt: Accurately predicted vectors are removed and the mesh is reformed.

unnecessarily introduced, and reform the mesh, as shown in the last row of Figure 6-5. If a tetrahedron is not subdivided, it is marked so that it is not subdivided again. The algorithm recurses, until subdivision is not necessary anywhere.

When subdividing meshes, there are two key concerns: maintaining connectivity of the mesh and tetrahedron skewness. To maintain connectivity, tetrahedra neighboring subdivided tetrahedra may need to be subdivided themselves because some of their edges are cut. This can be done following the scheme presented by Ruprecht and Muller [RM98] which was reviewed in Chapter 2. Unfortunately, when a tetrahedron has 4 or 5 edges cut, the resulting subdivided tetrahedra tend to be particularly skewed. To avoid this problem, we cut additional edges on those tetrahedra. This idea is a variant on the red-green subdivision scheme discussed in Chapter 2.

Note that sometimes the initial mesh representation over-describes the displacement field, and should therefore be reduced in density. It is straightforward to detect this condition, though it is not straightforward to reduce the density of an arbitrary mesh. In the previous section, we formed a mesh using octasection. We used that method so that it would be straightforward to reduce the density of the mesh – that is by undoing the octasection.

Figure 6-6 illustrates an example of this algorithm in practice. A tetrahedral mesh was placed in a sinusoidal vector field and asked to find a representation of the vector field within 1% error. In this case, determining the displacement field is simply a function call. One would expect that the number of nodes needed to represent the field would be smaller in the linear regions and larger in the high curvature regions of the sinusoid. This expectation is consistent with the results in the figure.

It is worth pointing out that the representation of a vector field that we develop is very similar in nature to a wavelet representation. In particular, the lifting method for forming wavelets [DGSS99] includes the same subdivide and prediction steps that we include here. Methods and analysis developed that extend the lifting method to irregularly sampled data are particularly relevant [DGSS99], and would be important if we were to use a non-linear interpolation scheme.

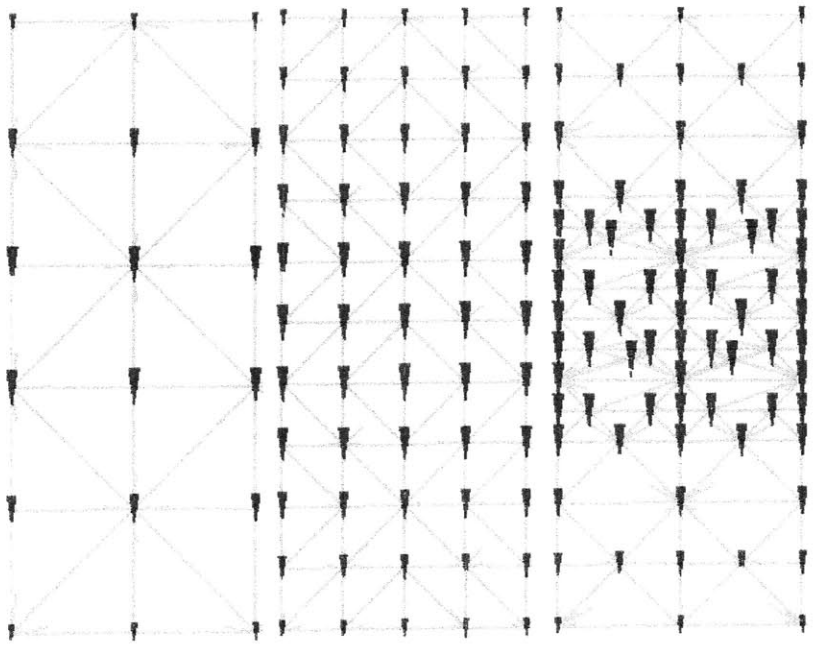
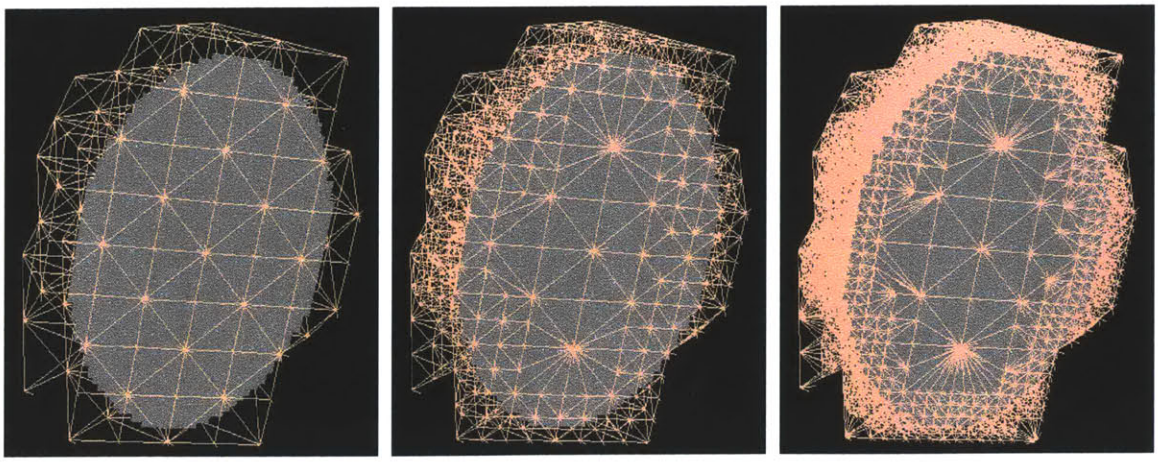


Figure 6-6: A slice of a tetrahedral mesh in a half-period of a sinusoidal vector field. The arrow size indicates vector length. (Note that the arrows are shortest at the top and bottom of the image, and longer in the middle.) Left: an initial mesh. Middle: the mesh after subdivision. Right: the mesh after the representation algorithm has converged.

6.2.3 Particulars of Image Matching

One reason to mark a tetrahedron to remain at the current resolution (without subdivision) is that the displacement is already sufficiently well described. When matching images, there are other reasons to mark a tetrahedron to remain without subdivision. First, if the tetrahedron in roughly the size of a voxel in the image, we do not subdivide it further. We make this decision because we expect there will be no further information in the image at a finer scale. A second reason to leave a tetrahedron without subdivision is that the tetrahedron lies entirely in a uniform intensity region. Such a region will almost definitely not contribute useful information to the matching problem. To determine if a tetrahedron is in a "uniform region", we pre-calculate the entropy in a small region around each voxel in an initial image and form a histogram of all the local image entropies. For the images we have examined, that histogram has a large peak near zero entropy, a local minimum, and then another peak at higher entropy. We define a tetrahedron as being in a uniform region if all the voxels it



Original First Iteration Second Iteration

Figure 6-7: Left: an initial tetrahedral mesh of an ellipsoid. Middle: subdivision of the mesh, only in regions of entropy greater than the threshold. Right: another level of subdivision. Tetrahedra were divided if any part overlapped a voxel with entropy higher than the threshold, or if the subdivision scheme necessitated they be subdivided to maintain mesh consistency.

overlaps have entropies less than the entropy of the local minimum in the histogram. Such a tetrahedron is marked to remain un-subdivided. Figure 6-7 shows an example mesh of an ellipsoid subdivided only in regions of entropy larger than the minimum.

There is one other minor change to the algorithm. When image matching, we will determine the displacements of all nodes simultaneously. Effectively, we do the *Predict* and *Measure* steps simultaneously. That is, we subdivide and determine the displacements of all nodes simultaneously. We can then ask if the displacements of the new nodes can be determined from the displacements of the old nodes. We therefore summarize the adaptive representation as follows:

```
I) Form coarse mesh of tetrahedra (unmarked).
      1) Adapt mesh to image.
      2) Mark tetrahedra in uniform regions.
II) While (there are unmarked tetrahedra remaining)
      1) Subdivide all unmarked tetrahedra.
      2) Subdivide extra tetrahedra to maintain consistency.
      3) Measure the displacement field at all nodes.
      4) Unsubdivide and mark tetrahedra that over-sample
         the deformation field.
      5) Mark small tetrahedra, and tetrahedra in uniform regions
```

6.2.4 Determining the Deformation Field

We use a probabilistic approach to find the displacement $\vec{r}$ that will transform one image V into another image U. To motivate that formulation, we define a random variable x that indexes all spatial positions, so that U and V become random variables, depending on x. We write the joint probability of U, V and $\vec{r}$ as

$$P(U, V, \vec{r}) = P(U, V_{\vec{r}}|\vec{r})P(\vec{r}). \tag{6.1}$$

where $V_{\vec{r}}(x_i)$ is the position x_i in the transformed image V. The first term $P(U, V_{\vec{r}}|\vec{r})$ is an image agreement term and the second term $P(\vec{r})$ is a prior probability which will ensure the smoothness of $\vec{r}$. We maximize the posterior probability of $\vec{r}$, which in this case is equivalent to maximizing the joint probability $P(U, V, \vec{r})$ because the marginal $P(U, V)$ is independent of $\vec{r}$.

To form $P(\vec{r})$, we look to statistical physics: the probability that a system is in a configuration is proportional to $e^{-E/(K_B T)}$ where $K_B T$ is a temperature (which will become a parameter in our model) and E is the linear elastic energy. We use a linear elastic energy because the energy is straightforward to calculate and because it has desirable properties which are similar to those of tissue. (Although, as pointed out in [MC02], tissue is actually a far more complicated material.)

As the images U and V may be of different modalities, a probabilistic image agreement term can be useful to align the images [BT01, PMV00, WVK95, ZFW03]. In certain cases, $P(U, V_{\vec{r}}|\vec{r})$ is known when the two images are perfectly aligned – this information can be obtained from previously solved matches. In this case maximizing Equation 6.1 is equivalent to minimizing an entropy term and a KL divergence. To show this let $P_{\vec{r}}(U(x_i), V_{\vec{r}}(x_i)|\vec{r})$ be the probability of joint intensity overlap as determined by histogramming and let $P_{\vec{r}_0}(U(x_i), V_{\vec{r}}(x_i)|\vec{r})$ be the probability of the current intensity overlap using the known intensity distribution at alignment. Let the joint intensity probability at every pixel be independent of every other pixel, and

identically distributed (IID) so that

$$\hat{\vec{r}} = \arg\max_{\vec{r}} \log P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r}) + \log P(\vec{r}) \tag{6.2}$$

$$= \arg\max_{\vec{r}} \sum_i \log P_{\vec{r}_0}(U(x_i), V_{\vec{r}}(x_i)|\vec{r}) + \log P(\vec{r}) \tag{6.3}$$

$$= \arg\max_{\vec{r}} n \sum_i \frac{1}{n} \log P_{\vec{r}_0}(U(x_i), V_{\vec{r}}(x_i)|\vec{r}) + \log P(\vec{r}) \tag{6.4}$$

$$\approx \arg\max_{\vec{r}} nE_{P_{\vec{r}}}[\log P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r})] + \log P(\vec{r}) \tag{6.5}$$

$$= \arg\max_{\vec{r}} nE_{P_{\vec{r}}}[\log \frac{P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r})}{P_{\vec{r}}(U, V_{\vec{r}}|\vec{r})} + \log P_{\vec{r}}(U, V_{\vec{r}}|\vec{r})] + \log P(\vec{r}) \tag{6.6}$$

$$= \arg\max_{\vec{r}} -nD(P_{\vec{r}}(U, V_{\vec{r}}|\vec{r})||P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r})) - nH(P_{\vec{r}}(U, V_{\vec{r}}|\vec{r})) + \log P(\vec{r}). \tag{6.7}$$

The advance from Equation 6.2 to Equation 6.3 uses the independence of the joint intensity distribution at every voxel. The advance from Equation 6.4 to Equation 6.5 uses the large of large numbers.

More commonly, one does not does have advance knowledge of $P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r})$. In this case, one can form an iterative algorithm, estimating $P_{\vec{r}_0}$ from the histogram intensities that overlap in the current pose, $\vec{r}_0$, to try to estimate a better pose $\vec{r}$. This is the method we use.

Using this type of iterative algorithm, the KL divergence term can be made as small as desired by taking $\vec{r}$ arbitrarily close to $\vec{r}_0$; that is by making small changes in $\vec{r}$ in each iteration. Thus the minimization is nearly a minimum entropy formulation.

It is worth noting that if n is large enough, the smoothing term, $\log P(T)$, becomes small compared to the image agreement term – it is generally true in MAP estimators that priors become less important as the amount of data increases. It is perhaps also worth noting that for large n and for a transformation T that makes the objective function larger than $\vec{r}_0$, $E[\log P_{\vec{r}_0}(U, V_{\vec{r}}|\vec{r})] > E[\log P_{\vec{r}_0}(U, V_{\vec{r}_0}|\vec{r}_0)]$. It seems doubtful that a new transformation would be more likely than an old transformation, using the probability distribution found under the old transformation. Some analysis suggests that spatial correlations can make this type of matcher work. A review of this method in comparison to other methods by Zollei *et al.* can be found in [ZFW03].

6.2.5 Locally Maximizing the Objective Function

Let K be the elasticity matrix of the mesh determined in the standard fashion [ZT89]. Inserting the linear elastic energy formula into the objective function, Equation 6.3 becomes:

$$\arg\max_{\vec{r}} \left[\sum_{x_i \in \text{All Voxels}} \log P_{\vec{r}_0}(U(x_i), V_{\vec{r}}(x_i)|\vec{r}) + \frac{-\vec{r}^T K \vec{r}}{2K_B T} \right]. \tag{6.8}$$

To solve this equation, we are motivated by the fact that many medical images are uniform over large areas. As shown in Figure 6-1, matching in uniform regions is difficult; any candidate displacement in a uniform intensity region will be only marginally better than a neighboring candidate displacement. To quantify this idea mathematically, we use block matching [GM90].

At each iteration, we use block matching to find a candidate displacement $\vec{a}_i$ of each voxel. That is, we take a small region around each voxel and test several possible displacements to see where in the other image that region best matches.

However, finding a candidate displacement is not sufficient for Equation 6.8. That equation looks to find a displacement field that trades off quality of match and smoothness. This tradeoff can be made by comparing the probability of each block matched result with the probability of matching to neighbors around that result. We use this information to approximate each term in the sum in Equation 6.8 as a quadratic, making the equation easily solvable. That is, we approximate

$$\log P_{\vec{r}_0}(U(x_i), V_{\vec{r}}(x_i)|\vec{r}) \approx \sum_i (\vec{r_i} - \vec{a_i})^T M_i (\vec{r_i} - \vec{a_i}) \tag{6.9}$$

where $\vec{r_i}$ is the displacement of voxel x_i and M_i is a 3x3 matrix evaluated by block matching. Re-examining Figure 6-1, it is clear that the red box matches to some point on the surface in the target image. Matching to a location in the normal direction to the target surface is unlikely. However, matching to slightly to the right or left on the target surface may be reasonable. The quadratic approximation to the image matching energy should capture this information.

Note that by approximating each term as a quadratic, we are effectively making a

Newton solver where we are approximating the Hessian as being zero for each node-node interactions. We expect this approximation to work very well for voxels that are well-separated; and poorly for neighboring voxels. However, for voxels that are near each other, the elastic matching energy should capture some of those interactions.

Equation 6.9 suggests an approximation to reduce the number of calculations needed to form the Hessian. In regions where the quadratic is known to have insignificant curvature relative to the curvature of the elasticity matrix, one can approximate M_i as being exactly 0 and avoid calculating it. In uniform regions, for example, if we were to spend the computational effort to find matches, the matches found would be insignificantly better than a match to a neighboring block. By ignoring voxels in nearly uniform regions, we effectively ignores terms in the summation in Equation 6.8 that are nearly independent of the displacement field $\vec{r}$. That is, the quadratic approximation of that term has insignificant curvature.

We therefore modify the summation in Equation 6.8 so that we only consider voxels in regions of high intensity variations. Equation 6.8 becomes

$$\arg\max_{\vec{r}} \left[\sum_{x_i \in \text{Selected Voxels}} (\vec{r_i} - \vec{a_i})^T M_i (\vec{r_i} - \vec{a_i}) + \frac{-\vec{r}^T K \vec{r}}{2K_B T} \right]. \tag{6.10}$$

In practice, we only consider voxels close to nodes of the current mesh. We start the first iteration using only those nodes that lie in the largest 50% of intensity variations (as determined by local image entropies). We then add nodes that lay in regions of smaller and smaller intensity variations until convergence. Equation 6.10 becomes a quadratic equation in $\vec{r}$, and is easily maximized. Although, as the algorithm is iterative, it will typically require several iterations before convergence of a displacement field.

Equation 6.10 has been examined by Papademetris *et al.* [POS+01]. They note that when image matching, one primarily desires the warped images to match, and secondarily for the transformation to have desired smoothness properties. They propose the use of a memory-less mesh – a memory-less mesh is not aware that it has already been stretched in previous iterations. By using a memory-less mesh, the

elastic energy is reset to 0 after each iteration so that after enough iterations, one effectively only maximizes the image agreement term. However, the final transformation is a sum of smooth transformation that are the solutions to Equation 6.10. Thus, the memory-less mesh concept leads to a solution with desirable smoothness properties, but a good final image agreement term. As these are properties we seek, we use the methods proposed by Papademetris *et al.*.

In order to extend the range of motions that the method can capture, we downsample the images for the lowest resolution mesh. For all other resolution meshes, we match using the original images.

6.2.6 Comparison to Chapter 3

The methods we develop here are very similar to those of Chapter 3. The MAP estimator in each chapter is nearly identical. Both methods use free form shape matching – that is both methods determine a non-rigid registration without landmarks. However, there are two key differences between the two sets of methods. First, we use adaptive methods here, while in Chapter 3, we used a mesh with an unchanging number of nodes. Second, here we solve by finding a candidate displacement, and then letting the node slide along a surface according to the confidence in the match in each direction. In Chapter 3, we applied image derived forces and elastic forces to the nodes and let the nodes move in the direction of the sum of those forces.

It is also worth realizing that in Chapter 3, we worked very hard to obtain convergence due to the ill-conditioned nature of the equations solved. Here, the equations are very well conditioned due to the approximation of the Hessians about maximums in the image agreement function for each node.

6.2.7 Summary

In summary, we create an algorithm for the non-rigid registration of images. To accomplish that goal, we form an initial mesh which we adapt to place nodes in the highest intensity regions of the image. We then solve Equation 6.10. and use the

Predict, *Measure* and *Adapt* protocols to find a new mesh to describe the displacement field.

6.3 Results

6.3.1 Shape Matching

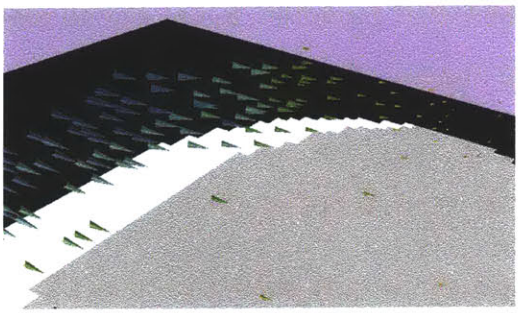


Figure 6-8: An oblique view of a slice through an ellipsoid. The white region indicates the compression of the ellipse. The vectors shown represent the deformation field. Note that nearly all the vectors are clustered around the surface that deformed, with more arrows in the regions of largest deformation. The arrows are scaled and colored by vector length.

We begin by matching shapes since the results are particularly easy to understand. Figure 6-8 shows part of the deformation field found between an initial ellipsoid and an ellipsoid compressed on one side. The maximum deformation was 3 voxels, and the maximum allowed error in the representation was 0.1 voxels. The resulting representation of the deformation field used approximately 7000 nodes. Though, at its densest, it used 14000 nodes. The entire process required less than 20 seconds to run on a single processor 2 Ghz processor. Note in the image how the vectors are only dense in the left portion of the image where the deformation is greatest. On the right portion of the image, where there is little deformation, there are very few vectors.

We have matched other shapes with similar results to the ellipsoid case. For three patients, we matched intra-operative segmentations of the prostate to pre-operative segmentations of the prostate; resulting correspondences are shown for one case in Figure 6-9. For these images, we allowed an error in the representation of the dis-

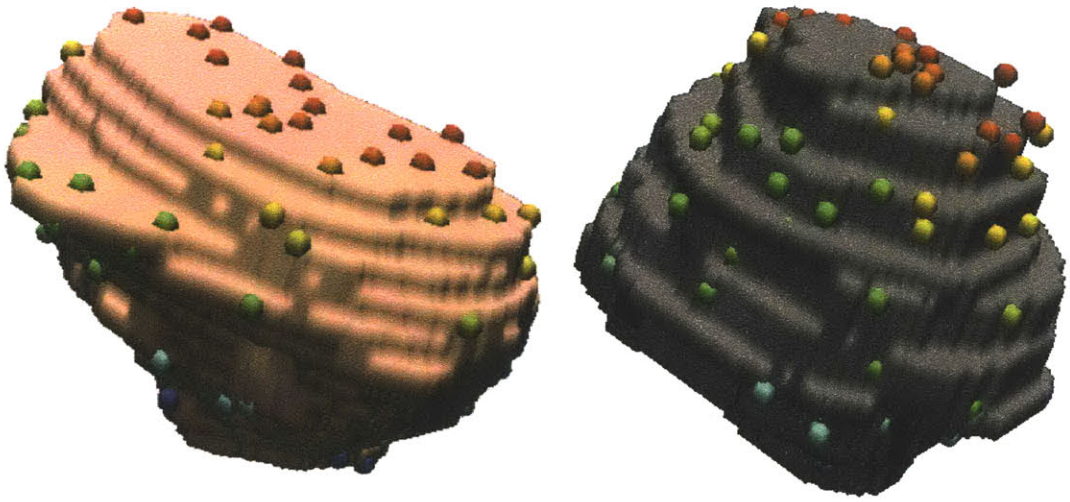


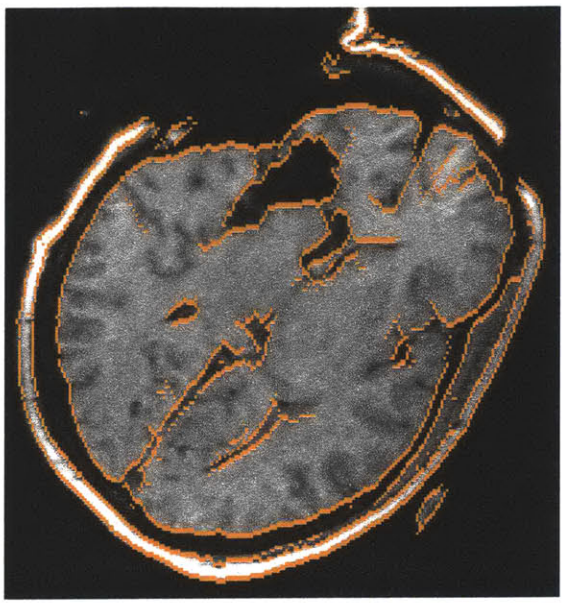
Figure 6-9: Non-rigid matching of intra-operative prostate (left) to pre-operative prostate (right). The colored spheres indicate points on the surface of the initial prostate and their corresponding location on the target prostate. Note the "bulge" in the prostate on the left. Methods in Chapter 3 were designed to not match to such protrusions that are not in the other image. However, methods in this chapter are designed to match to it.

placement field of 0.2 mm. Those matches used roughly 4000 nodes to describe the deformation field and were finished in less than one minute. Hausdorff distances between the resulting warped label map and the target label map were one voxel or less.

Note that one of the differences between the methods in Chapter 3 and this chapter is shown in Figure 6-9. In Chapter 3, the matcher was designed to use an elastic energy prior to avoid matching to protrusions. As the elastic prior in this chapter is memoryless, the matcher presented here is able to find a reasonable match.

6.3.2 Brain Images

Figure 6-10 shows two MRI images of the same brain taken during brain surgery. The images have 256x256x124 voxels with 0.9375x0.9375x1.5 millimeter spacing. A visual comparison of the two images shows that nearly all the deformation between the two images occurs near the incision. The result of the non-rigid registration algorithm is shown in the lower left of the figure. The warped image is remarkably similar to the initial target image. In particular, the small displacement of the left



Target

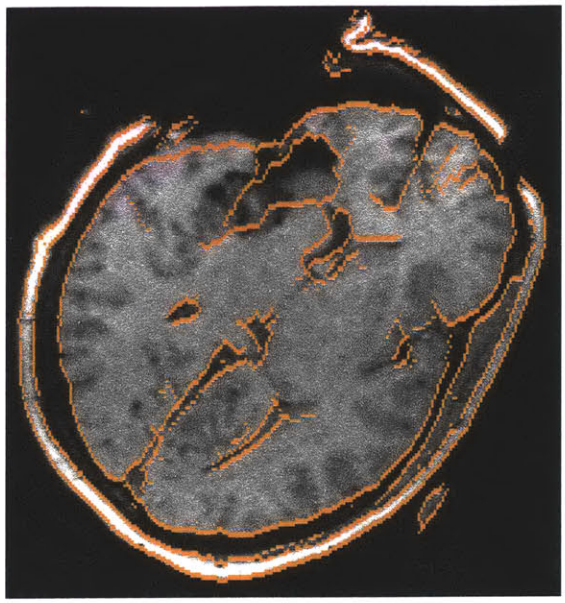
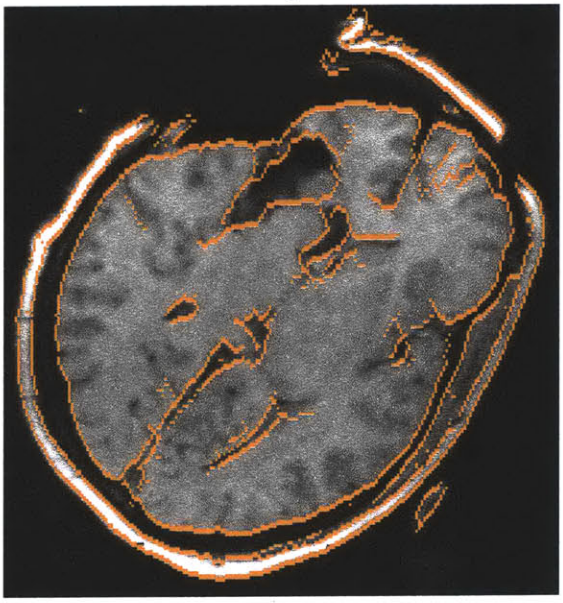
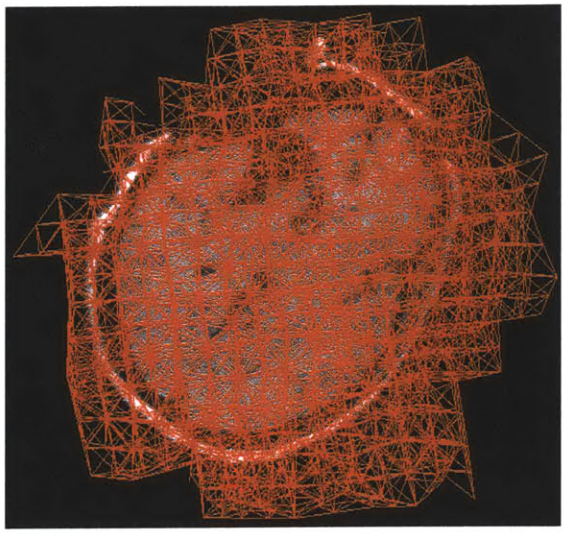


Image to be deformed

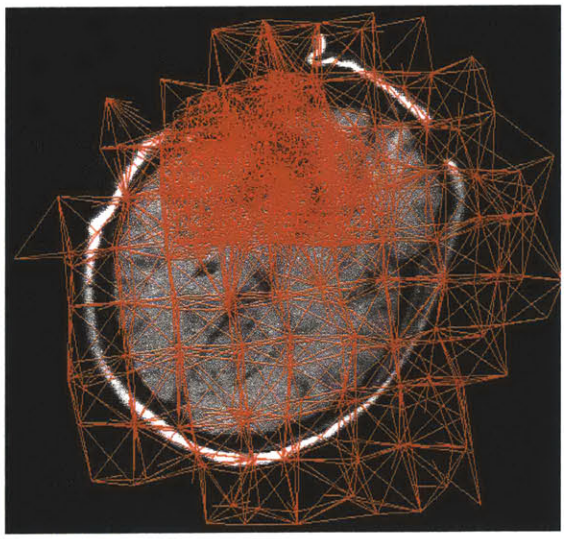


Result of deformation

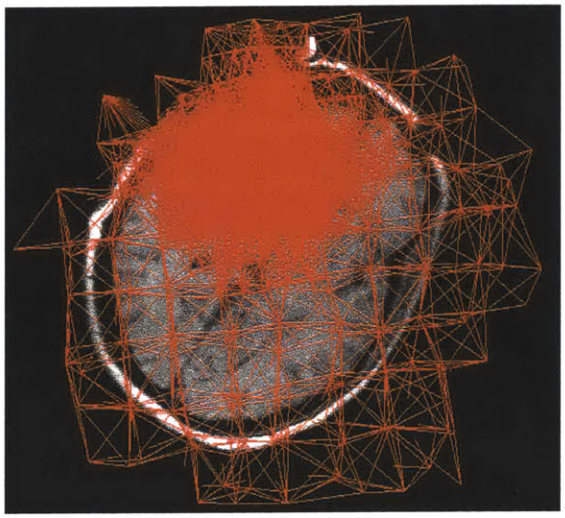
Figure 6-10: Top left: slice through MRI of a brain taking during surgery with edges highlighted. Top right: MRI taken of the same brain later in the surgery, with the edges from left image overlayed. Bottom left: the result of deforming the top right image onto the top left image, with the edges of the top-left image overlaid.



Initial Mesh



First Adaptation



Final Mesh

Figure 6-11: Top Left: projection of the initial mesh used for the non-rigid registration process onto a slice of the image. Top Right: projection of the first adaptation of the mesh. Bottom Right: projection of the final mesh. No further adaptation was needed because the tetrahedra would otherwise become too small.

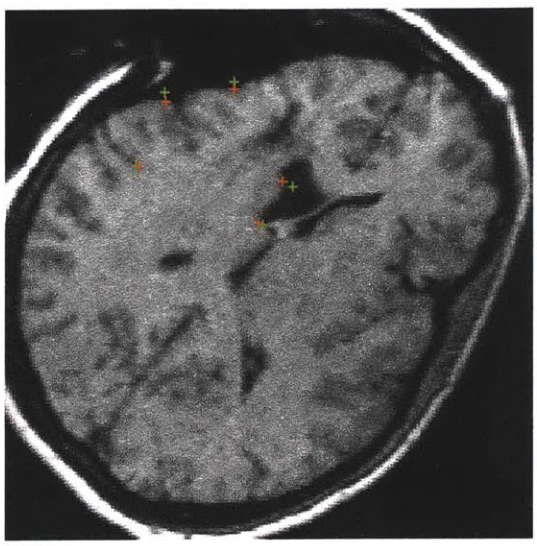
and right ventricles was captured. The boundaries near the incision appear to be in the appropriate places as well.

The plots in Figure 6-11 shows an initial coarse mesh used to represent the vector field, an intermediate description and the final representation of the vector field. The final mesh is heavily concentrated in the regions where the deformation field varies the most, and very sparse in all other regions. The final match used 8500 nodes, and the entire matching process was finished in less than 6 minutes on a 2 Ghz desktop computer. If the mesh had been represented uniformly everywhere, the mesh would have contained 5 million nodes and solving Equation 6.10 would not have been practical on a desktop machine.

While it is difficult to see in the images in Figure 6-10, there are small, slowly varying motions of 1 millimeter or less in the brain in the bottom right of the images, far away from the incision. The adaptive process was able to find these motions. However, the method determined that it was not necessary to represent those displacements finely.

To validate the found deformation field, we compare manually found landmarks in the initial images and the deformed image. Such landmarks are shown in a slice through the data in Figure 6-12. In the slice shown, motions are as large as 5 millimeters. The majority of that motion was captured by the non-rigid registration algorithm, with disagreements between manually chosen landmarks and the algorithm of 1 millimeter or less. These disagreements may not be significant because tests to re-locate landmarks generally found different locations than the original locations by 1 millimeter.

Examining errors in the remainder of the brain (not shown), the majority of the motion was captured by the non-rigid registration algorithm – that is, errors in the algorithm were generally significantly smaller than the motions of tissue. In particular, errors in the estimate of the motion of the ventricle closest to resection were generally 1 millimeter or less. However, errors at a small part of the posterior of right ventricle reached 2 millimeters. A careful examination of this portion of the ventricle found that the ventricle had collapsed to effectively zero width in the image to be



Target

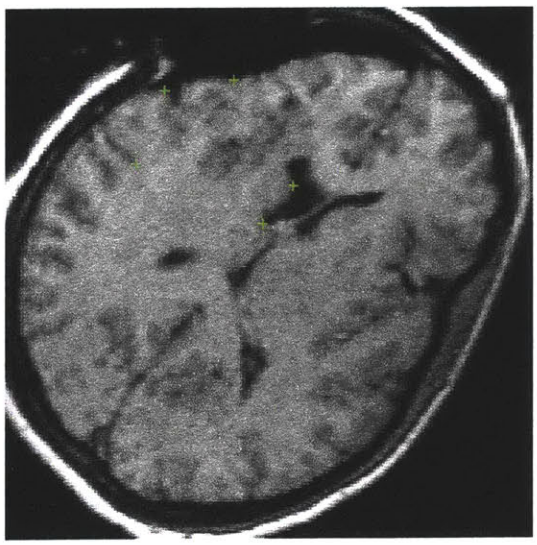
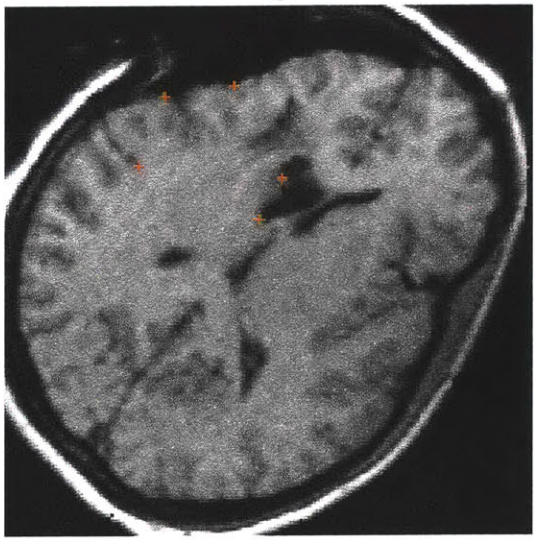


Image to be deformed



Deformed Image

Figure 6-12: Comparison of manually labeled fiducials on the initial images and deformed image. Top Right: Image to be deformed with fiducials in green. Top Left: Target image with fiducials in red. The green fiducials from the top right image are also shown. Bottom: the deformed image with the red fiducials from the target image overlayed on the fiducials found in the warped image, in green. Notice that most of the motion of the fiducials is captured. Maximum motions within the image are between 4 and 5 millimeters; errors in motion estimation are 1 millimeter or less.

deformed – the non-rigid registration algorithm was unable to "re-create" that region of the ventricle to do the matching. Nonetheless, as motion of that ventricle reached a maximum of 5 millimeters, the majority of the motion was captured. Motion of gyri and sulci not directly in the resection region were captured to within 1 millimeter in all parts of the brain; motions of those structured were between 5 and 6 millimeters near the incision and 0 and 2 millimeters far away from the incision.

Near the incision, the outer cortical surface motions were captured very accurately, mostly to within 1 millimeter. However there were errors at the surface of resection that reached 2 to 3 millimeters. It was challenging for the non-rigid registration algorithm to correctly determine the motion because tissue was resected between images. However, while the method made errors near the resection, it was able to capture a majority of the motion in that region, which reached a maximum of 13 millimeters. Furthermore, such errors are approximately the same as those made by other methods designed to capture brain tissue deformation during surgery [FNM+01].

6.3.3 Female Pelvis Images

The top of Figure 6-13 shows two images of a female pelvis. The images are MRI acquisitions with 512x512x8 voxels and 0.78125x0.78125x5 millimeter spacing. In one of the images, the subject was asked to engage her pelvic muscles. This image sequence was taken to aide a researcher in understanding the motion of pelvis muscles. The bottom left image of Figure 6-13 shows the resulting warped image overlayed by edges of the target image. The majority of edges of the warped image appear underneath edges of the initial image.

A slice through the representation of the vector field is shown in the bottom right image of Figure 6-13. Note that there are regions where there are very few vectors because very little deformation occurred in those regions. The resulting representation used 4000 vectors and required roughly 5 minutes to compute.

To validate the matching algorithm, corresponding points in the warped and target image were found manually by a single individual at 10 points, as shown in Figure 6-14. Because of the difficulties of finding landmarks in three dimensions for

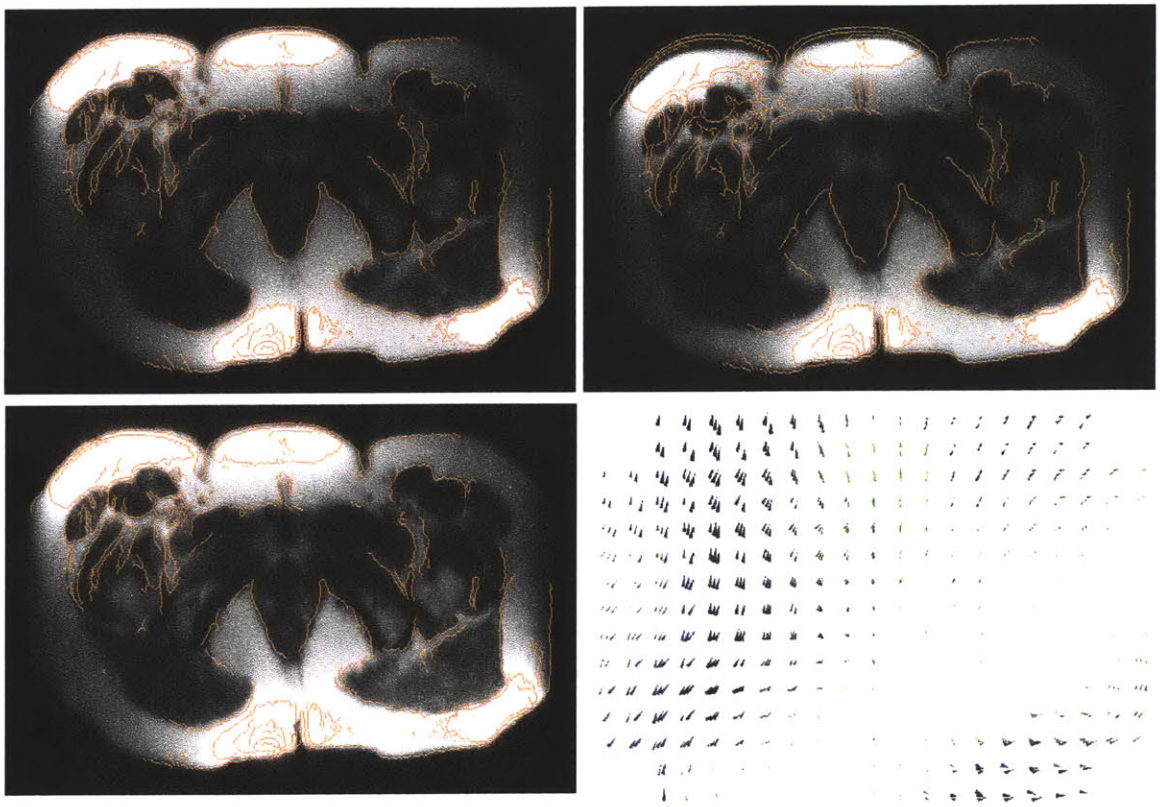


Figure 6-13: Top left: MRI of a female pelvis. Orange lines indicate edges of this image and have been placed over all images for easy comparison. Top Right: MRI of the same subject after being asked to squeeze. Bottom Left: Warped version of the picture in the top right. Bottom Right: vector representation of the deformation field. Note the large region where the deformation field is sparsely represented.

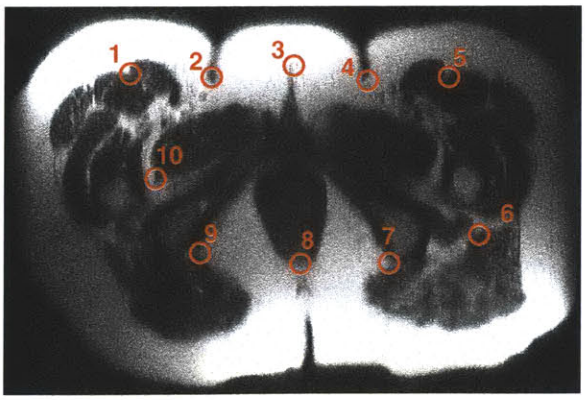


Figure 6-14: The points in the image indicate points at which the pelvis correspondences were validated.

Point	1	2	3	4	5	6	7	8	9	10
Error(mm)	1	3	1	1	1	0	0	7	2	2
Disp.(mm)	10.0	6.4	4.3	2.2	3.1	2.6	4.1	0.6	8.2	6.3

Table 6.1: Errors in the non-rigid registration algorithm at the 10 points in Figure 6-14. The table shows numbers rounded to the nearest millimeter as it is doubtful that any further digits are meaningful. Axial Displacements of the points found using the non-rigid registration algorithm are shown to demonstrate that errors in the method were generally much smaller than the local displacement field.

this data set, those comparisons were made in two dimensions only, within an axial slice. Attempts to re-locate already chosen correspondence resulted in errors up to 1.5 millimeters, indicating that differences between the algorithm and the manually located correspondences below 1.5 mm may not be significant.

Table 6.3.3 shows the errors made by the algorithm in comparison to the manually segmented data. Most errors are 1 millimeter or less, consistent with other non-rigid registration algorithms [FNM+01]. However, there is a larger error made at Point 2, where the correspondence found was slightly shifted to the actual correspondence. Also, the algorithm made a large error at the most posterior boundary between the levator muscle and fat (Point 8 in Figure 6-14) where the method failed to capture the motion of the bottom of the levator. Both errors are visible in examining the bottom left image in Figure 6-13.

Examining the results of the algorithm carefully showed that the errors were not directly due to the representation. There were nodes of the tetrahedral mesh in both

regions where errors were made. The image-agreement term simply found that the correct correspondence was not as good a match as another region.

6.3.4 Computational Analysis

We divide the time that we use to solve the algorithm into three parts: time to manipulate the mesh, time to block match, and time to solve each iteration. For a 5 minute solve time with the roughly 10,000 nodes the algorithm has been using, we have found that mesh manipulation uses no more than a few seconds. Solving the equations in all the iterations together takes about 30 seconds of the solve time. Most of the remainder of the time involves block matching. Thus, to speed the algorithm further, effort should be put into block matching methods. For example, caching block matched results from one iteration to the next could reduce the run time of the method.

Our algorithm does access every voxel in the image, including once in each iteration to calculate image entropies and once to place the nodes of the mesh. Thus the algorithm does have computational complexity $O(N)$ where N is the number of voxels in the image. However, in practice, the amount of time the algorithm requires to complete is proportional to $O(M)$ where M is the number of nodes in the mesh. That is, $O(M)$ calculations are done to form the system of equations using block matching. Solving the system of equations is worst case $O(M^2)$ because the matrix is symmetric and sparse. However, as the equations are well-conditioned, in practice solving the system of equations requires $O(M)$ time.

6.4 Discussion

Our aim was to develop an intensity-based non-rigid matcher that would be fast enough to use during image guided surgery. To that end, we created a non-rigid registration tool based on an adaptive multi-resolution representation of a deformation field. For the deformed ellipsoid, the prostate shapes, the female pelvis data set, and the brain data set, the matcher was able to find deformation fields in a few minutes.

The matcher showed the ability to adapt to the matching problem, adding nodes in regions where the nodes were needed, instead of uniformly subdividing everywhere. The resulting computational cost savings allow matches to be found with less than 10,000 nodes and in less than 10 minutes. Without the multi-resolution abilities the brain-data set, for example, would have needed roughly 5 million nodes to find the deformation field. Solving such a system of equations would not have been possible on a desktop computer.

Unlike other intensity-based non-rigid registration algorithms with which we are familiar, the time our algorithm takes to converge is not dominated by the number of voxels in the image; it is instead dominated by the number of nodes in the mesh. For displacement fields that slowly vary over much of the image, like those often found between pre-operative and intra-operative images, this type of representation is very efficient.

While the representation we use is efficient for registering pre-operative and intra-operative images, for other types of non-rigid registration problems, the representation may not be an optimal one. In particular, when matching across subjects, one could imagine that there would be large deformations everywhere. If the adaptive algorithm must adapt everywhere, then the method loses any computational advantage over voxel based methods. In fact, when the number of nodes in the mesh approaches the number of voxels in the image, voxel based methods can be significantly faster than the methods we present here due to the overhead of solving on an unstructured grid rather than a regular grid.

Comparing the results of our method to manually located correspondences, the algorithm performed very well, generally making errors less than 1 millimeter. These errors are roughly the same size of errors found by other non-rigid registration algorithms [FNM+01]. However, the algorithm did make one particularly large error: missing the motion of the bottom of the levator muscle. This error resulted from the image agreement term. Statistical image agreement terms can make errors and it seems this error was a manifestation of that fact.

One might expect that our algorithm could make further errors by initially sam-

pling the displacement field too coarsely. That is, the algorithm can entirely miss deformations to an image localized within a tetrahedron. We view this problem as a standard one in hierarchical methods. For a hierarchical matcher, if a coarse resolution representations of a displacement field fails to find a reasonable approximation of the final displacement field, then finer representations typically have the same problems. While the danger of missing motions is present, it can be minimized by starting with a mesh that is sufficiently fine to capture all expected displacements. In our experience, starting with 5 millimeter spacing is more than sufficient for the non-rigid matching problems we explore.

Finally, it is worth point out that the fact that intra-operative data matching is the application of interest could be more fully exploited by the algorithm. In particular, as pre-operative data is often collected many hours before surgery, there is a lot of time for pre-processing. One of the most important improvements that could be made with that additional time is in the creation of the initial mesh. One could make a very complicated algorithm to adapt the mesh to the surfaces in the pre-operative image, rather than the simple method we have presented here. Such an adaptation might lead to further speed gains due to an improved representation of anatomy, and therefore perhaps reduced need to block-match elsewhere, or convergence in fewer iterations of the non-rigid matching algorithm.

6.5 Conclusion

Our aim was to develop an intensity-based non-rigid matcher that would be fast enough to use during image guided surgery. In particular, our aim was to develop methods whose dominant computational complexity do not scale like the number of voxels in an image. We accomplished those goals. We developed methods based on an adaptive tetrahedral mesh that is used to represent a displacement field. Those methods were able to non-rigidly warp three dimensional images into each other in roughly 5 minutes. The algorithm made errors that generally less than 2 millimeters, which is comparable to other methods [FNM+01].

Chapter 7

Conclusion

In this thesis, we argue that by making compact representations of anatomical features, we can make more efficient and more effective algorithms. We began by noting that voxel based representations inefficiently describe uniform regions found in segmented anatomical structures. Furthermore those representations describe the surface of anatomical structures as unrealistic jagged edges. We therefore developed a compact tetrahedra-based representation of an anatomical structure. The representations we created used fewer tetrahedra than the voxel-based representations used voxels. Furthermore, the tetrahedra-based representation could straightforwardly describe a smooth surface.

We then used the compact representations we developed to create free-form non-rigid shape matcher using a linear elastic regularization term. We overcame the ill-conditioned equations that resulted to create a non-rigid matcher that typically converged in thirty seconds to a minute on a desktop machine.

We used the results of that non-rigid registration algorithm to perform morphological studies of shape. We showed that the correspondences found by the non-rigid registration algorithm led to classifiers with improved performance over classifiers based on distance maps. We furthermore showed that the correspondences we found led to effective linear classifiers, while distance maps did not.

Not only did the non-rigid shape registration algorithm lead to effective morphological studies, it also led to an effective deformable model for segmentation. We cre-

ated a deformable model based on the non-rigid registration of a group of anatomical structures and merged that deformable model with an intensity based segmentation method. The result was a more accurate final method than intensity based methods alone. In particular, the segmentations found by intensity based method alone included disconnected regions of tissue, jagged edges, and unrealistic protrusions. Our combined method produced smooth surfaces and one connected region of tissue. Furthermore, our method achieved greater agreement with manual segmentations than the intensity based method alone.

Finally, we created a novel, adaptive method to non-rigidly register medical images. That technique uses a tetrahedral representation of a displacement field that adapts both to anatomy and to the displacement field between images. The resulting method had a computational cost that was dominated by the number of nodes in the mesh, rather than the number of voxels in the image. The resulting method converged in about five minutes, making it fast enough for use during surgery.

Clinically, we validated other's results on finding the correlation of shape of the amygdala-hippocampus with Schizophrenia. Furthermore, we presented new results showing a correlation between the shape of the thalamus and first episode schizophrenia. We developed a full brain segmentation algorithm and validated the accuracy of that method in segmenting the thalamus. Finally, we developed a non-rigid registration tool for use during surgery.

7.1 Future Directions of Research

In this thesis, we discussed representational issues in medical images. We plan to continue pursuing that direction of research. In particular, we have pointed out that segmented anatomical shapes have uniform intensity regions. Medical images have nearly uniform textures or uniform intensity over large regions; it seems inefficient to describe such regions using a voxel representation. We plan to investigate representing a gray scale image using volumetric elements – it should be possible to represent an image as low order polynomials over tetrahedra. The large regions of uniform intensity

could be represented very efficiently. Though, it will be very challenging to find an efficient representation in regions of the image where the intensity varies a lot; in these regions volumetric elements may not be an efficient representation.

A second direction of research we would like to pursue concerns hierarchical volumetric representations of anatomical objects. Such a representation could be very powerful for non-rigidly matching complicated shapes between subjects such as the cortical folds. In particular, the non-rigid shape matcher developed in Chapter 3 has very little chance of succeeding in matching the cortical fold because of the strong likelihood that a cortical fold in the deforming structure would be attracted to the wrong cortical fold in the target structure. A hierarchical method, where an alignment is done one level at a time, may be able to overcome that problem. Furthermore, much of the technology needed to create such a representation was already developed in Chapter 6.

A hierarchical representations may also be particularly useful for comparing anatomical structures. Nodes in a hierarchical mesh in one level of the mesh can be introduced based on local coordinate system relative to the coarser level of the mesh. Such a representation has the advantage of having a dependence on global orientation only in its coarsest level. As discussed in Chapter 4, choosing the "correct" orientation is often a problem when comparing shapes. A hierarchical description using this type of relative coordinate system could strongly mitigate these problems; on the coarsest level would have any dependence on orientation.

There is a second direction of research we intend to pursue. In our view, choosing a good representation is only part of the challenge of developing efficient and effective methods. Another part of the challenge is using a good metric to compare those representations. In Chapter 4 and Chapter 5, we used Euclidean metrics to compare displacement fields or node locations. As displacement fields form a vector space (Section 4.1.3), a Euclidean metric is probably appropriate.

However, we are also interested in comparing tensors. In particular, we desire to compare local strain or stress tensors between subjects, rather than comparing the vector field. Furthermore, researchers at Brigham and Women's hospital are collecting

diffusion tensor data across many subjects which they would like to compare. For tensor comparisons, we believe that the Amari metric [Ama98] or the metric recently introduced by Miller and Chefd'hotel [MC03] are likely appropriate choices to compare strain matrices. Both of these choices have been designed to account for the group structure of tensors. We expect improved performance using metrics that account for group structure over metrics that do not.

7.2 Conclusion

We argued that incorporating anatomical features directly into representations would lead to more compact representations and more efficient algorithms. We created non-rigid registration algorithms using this idea. We demonstrated the efficiency and effectiveness of those methods.

Bibliography

[ABE97] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1997.

[Ama98] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[AMP00] D. Arnold, A. Mukherjee, and L. Pouly. Locally adapted tetrahedral meshes using bisection. *SIAM*, 22:431–338, 2000.

[Ber99] M. Berzins. Mesh quality: A function of geometry, error estimates or both? *Engineering With Computers*, 15(3):236–47, 1999.

[Bey91] W. H. Beyer. *CRC Standard Probability and Statistics.* CRC Press, Boca Raton, Florida, 1991.

[BPar] M. Bern and P. Plassmann. Mesh generation. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry.* Elsevier Science, to appear.

[BT01] T. Butz and J.-P. Thiran. Affine registration with feature space mutual information. In *MICCAI*, pages 549–556, October 2001.

[CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proc. European Conference on Computer Vision*, 1407:484–498, 1998.

[CHF01] C. Chefd'Hotel, G. Hermosillo, and O. Faugeras. A variational approach to multi-modal image matching. In *Variational and Level Set Methods in Computer Vision*, pages 21–28, 2001.

[CHTH92] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):9–18, 1992.

[CRM96] G. E. Christensen, R. D. Rabbit, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, 1996.

[CTCG92] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, pages 9–18, 1992.

[Dav96] C. Davatzikos. Spatial normalization of 3d brain images using deformable models. *Journal of Computer Assisted Tomography*, 20(4):65–665, 1996.

[DCT01] R. H. Davies, T. F. Cootes, and C. J. Taylor. A minimum description length approach to statistical shape modeling. In *IPMI*, pages 50–63, Davis, CA, June 2001.

[Dem02] S. Demeo. (Research Assistant and Expert Segmenter, Brigham and Women's Hospital, Boston, Ma.) Personal Communication, 2002.

[DGSS99] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens. Wavelets on irregular point sets. *Phil. Trans. R. Soc. Lond. A*, 357(1760):2397–2413, 1999.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, Berkshire, England, 2001.

[Dji00] H. Djidjev. Force-directed methods for smoothing unstructured triangular and tetrahedral meshes. In *Proceedings of the 9th International Meshing Roundtable*, pages 395–406, October 2000.

[DP94] C. Davatzikos and J. Prince. Brain image registration based on curve mapping. *IEEE Workshop Biomedical Image Anal*, pages 245–254, 1994.

[DS88] J. Dengler and M. Schmidt. The dynamic pyramid — a model for motion analysis with controlled continuity. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:275–286, 1988.

[DVR+96] C. Davatzikos, M. Vaillant, S. M. Resnick, J. L. Prince, S. Letovski, and R. N. Bryan. A computerized approach for morphological analysis of the corpus callosum. *Journal Computer Assisted Tomography*, 20(1):88–97, 1996.

[ES96] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.

[FCM+01] M. Ferrant, O. Cuisenaire, B. Macq, J.-P. Thiran, and S. K. Warfield. Surface based atlas matching of the brain using deformable surfaces and volumetric finite elements. In *MICCAI*, pages 1352–1353, 2001.

[Fer97] M. Ferrant. *Physics-based Deformable Modeling of Volumes and Surfaces for Medical Image Registration, Segmentation and Visualization.* PhD thesis, Universite catholique de Louvain, Louvain-la-Neuve, Belgium, 1997.

[FJP95] L. Freitag, M. Jones, and P. Plassmann. An efficient parallel algorithm for mesh smoothing. In *Proceedings, 4th International Meshing Roundtable*, pages 47–58, Sandia National Laboratories, 1995.

[FMNW00] M. Ferrant, B. M. Macq, A. Nabavi, and S. K. Warfield. Deformable modeling for characterizing biomedical shape changes. In *DGCI*, pages 235–248, 2000.

[FNM+01] M. Ferrant, A. Nabavi, B. M. Macq, F. A. Jolesz, , R. Kikinis, and S. K. Warfield. Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model. 20(12):1384–1397, December 2001.

[FOG96] L. Freitag and C. Ollivier-Gooch. A comparison of tetrahedral mesh improvement techniques. In *Fifth International Meshing Roundtable*, pages 87–100, Sandia National Laboratories, 1996.

[Fre92] W. T. Freeman. *Steerable Filters and Local Analysis of Image Structure.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[FWN+00] M. Ferrant, S. K. Warfield, A. Nabavi, F. A. Jolesz, and R. Kikinis. Registration of 3D intraoperative MR images of the brain using a finite element biomechanical model. In *MICCAI*, pages 19–28, 2000.

[GBBH95] J. Gee, L. Briquer, C. Barillot, and D. Haynor. Probabilistic matching of brain images. *IPMI*, pages 113–125, 1995.

[Gei93] B. Geiger. Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. Technical Report RR-2105, INRIA, France, December 1993.

[GGSK01] P. Golland, W. E. L. Grimson, M. E. Shenton, and R. Kikinis. Deformation analysis for shape based classification. In *IPMI*, pages 517–530, Davis, CA, June 2001.

[GKS02] E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A simple framework for adaptive simulation. *Computer Graphics Proceedings (SIGGRAPH)*, 21(3):281–290, July 2002.

[GLM96] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer Graphics Proceedings (SIGGRAPH)*, 30:171–180, 1996.

[GM90] H. Gharavi and M. Mills. Block matching motion estimation algorithms: New results. *IEEE Trans. Circuits Syst.*, 37:649–651, 1990.

[GMVS98] T. Gaens, F. Maes, D. Vandermeulen, and P. Suetens. Non-rigid mulitimodal image registration using mutual information. In *MICCAI*, pages 1099–1106, Cambridge, MA, October 1998.

[GNK+01] D. Gering, A. Nabavi, R. Kikinis, N. Hata, L. Odonnell, W. E. L. Grimson, F. Jolesz, P. Black, and W. M. Wells III. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. *Journal of Magnetic Resonance Imaging*, 13:967–975, June 2001.

[Gol01] P. Golland. *Statistical Shape Analysis of Anatomical Structures.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[GRAM01] A. Guimond, A. Roche, N. Ayache, and J. Meunier. Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. *IEEE Transactions on Medical Imaging*, 20(1):58–69, January 2001.

[GRB93] J. Gee, M. Reivich, and R. Bajcsy. Elastically deforming a three-dimensional atlas to match anatomical brain images. *J. Computed Assisted Tomography*, 17:225–236, 1993.

[GSS99] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH)*, pages 325–334, 1999.

[GSSL01] G. Gerig, M. Styner, M. E. Shenton, and J. A. Lieberman. Shape versus size: Improved understanding of the morphology of brain structures. In *MICCAI*, pages 24–32, Utrecht, October 2001.

[HHCS+02] T. Hartkens, D. L. G. Hill, A. D. Castellano-Smith, D. J. Hawkes, C. R. Maurer Jr., A. J. Martin, W. A. Hall, H. Liu, and C. L. Truwit. Using points and surfaces to improve voxel-based non-rigid registration. In *MICCAI*, pages 565–572, 2002.

[HKR93] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions PAMI*, 15:850–863, 1993.

[HNW+99] N. Hata, A. Nabavi, S. Warfield, W. M. Wells III, R. Kikinis, and F. A. Jolesz. A volumetric optical flow method for measurement of brain deformation from intraoperative magnetic resonance images. In *MICCAI*, pages 928–935, Cambridge, UK, October 1999.

[Hor87] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.

[HRS+99] A. Hagemann, K. Rohr, H. S. Stiehl, U. Spetzger, and J. M. Gilsbach. Biomechanical modeling of the human head for physically based, nonrigid image registration. *IEEE Transactions on Medical Imaging*, 18(10):875–84, October 1999.

[Kap99] T. Kapur. *Model based three dimensional Medical Imaging Segmentation.* PhD thesis, Massachusetts Institute of Technology, 1999.

[KGS01] P. Krysl, E. Grinspun, and P. Schröder. Natural hierarchical refinement for finite element methods. *International Journal for Numerical Methods in Engineering*, page To appear, 2001.

[KGWK98] T. Kapur, W. E. L. Grimson, W. M. Wells III, and R. Kikinis. Enhanced spatial priors for segmentation of magnetic resonance imagery. In *MICCAI*, Oct 1998.

[KMaJEA+99] J. S. Kwon, R. W. McCarley, Y. H. and J. E. Anderson, I. A. Fischer, R. Kikinis, F. A. Jolesz, and M. E. Shenton. Left planum temporale volume reduction in schizophrenia. *Arch. Gen. Psychiatry*, 56(2):142–148, 1999.

[KO98] P. Krysl and M. Ortiz. Generation of tetrahedral finite element meshes: Variational delaunay approach. In *Proceedings 7th International Meshing Roundtable*, pages 273–284, Sandia National Lab, October 1998.

[KSG+92] R. Kikinis, M. E. Shenton, G. Gering, M. An derson J. Martin, D. Metcalf, C. Guttmann, R. W. McCarley, W. Lorensen, H. Line, and F A. Jolesz. Routine quantitative analysis of brain and cerebrospinal fluid spaces with MR imaging. *Journal of Magnetic Resonance Imaging*, 2(6):619–629, 1992.

[KSG98] A. Kelemen, G. Szekely, and G. Gerig. Three-dimensional model-based segmentation of brain MRI. In *Mathematical Methods in Biomedical Image Analysis*, pages 4–13, 1998.

[KSG99] A. Kelemen, G. Szekely, and G. Gerig. Elastic model-based segmentation of 3-d neuroradiological data sets. *IEEE Transactions on Medical Imaging*, 18(10):828 –839, 1999.

[KSS+03] K. Kasai, M. E. Shenton, D. F. Salisbury, Y. Hirayasu, C. U. Lee, A. A. Ciszewski, D. Yurgelun-Todd, R. Kikinis, and F. A. Joleszand R. W. McCarley. Progressive decrease of left superior temporal gyrus gray matter volume in patients with first-episode schizophrenia. *Am J Psychiatry*, 160(1):156–64, 2003.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics Proceedings (SIGGRAPH)*, 21(3), 1987.

[LDGC99] P. Labbe, J. Dompierre, F. Guibault, and R. Camarero. On element shape measures for mesh optimization. In *2nd Symposium on Trends in Unstructured Mesh Generation,5th US Congress on Computational Mechanics*, University of Colorado, Boulder, August 1999.

[Lev00] M. Leventon. *Statistical Models in Medical Image Analysis.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[LFG00] M. Leventon, O. Faugeraus, and W. E. L. Grimson. Level set based segmentation with intensity and curvature priors. In *Mathematical Methods in Biomedical Image Analysis*, pages 4–11, 2000.

[LMVS99] K. Van Leemput, F. Maes, D. Vanermeulen, and P. Suetens. Automated model-based bias field correction of MR images of the brain. *IEEE Transactions on Medical Imaging*, 18(10):885–895, 1999.

[MBF03] N. Molino, R. Bridson, and R. Fedkiw. Tetrahedral mesh generation for deformable bodies. In *Submission*, 2003.

[MC02] K. Miller and K. Chinzei. Mechanical properties of brain tissue in tension. *Journal of Biomechanics*, 35(4):483–490, 2002.

[MC03] E. G. Miller and C. Chefd'hotel. Practical non-parametric density estimation on a transformation group for vision. In *CVPR*, 2003.

[MG98] A. M. C. Machado and J. C. Gee. Atlas warping for brain morphometry. In *SPIE Medical Imaging, Image Processing*, pages 642–651, 1998.

[MPH+00] M. I. Miga, K. D. Paulsen, P. J. Hoopes, F. E. Kennedy, A. Hartov, and D. W. Roberts. In vivo quantification of a homogeneous brain

deformation model for updating preoperative images during surgery. *IEEE Transactions on Medical Imaging*, 47(2):266–273, February 2000.

[MPK94] J. Martin, A. Pentland, and R. Kikinis. Shape analysis of brain structures using physical and experimental models. In *CVPR*, pages 752–755, 1994.

[MSH+02] J. Masumoto, Y. Sato, M. Hori, T. Murakami, T. Johkoh, H. Nakamura, and S. Tamura. A new similarity measure for nonrigid volume registration using known joint distribution of target tissue: Application to dynamic ct data of the liver. In *MICCAI*, pages 493–500, Tokyo, October 2002.

[MVB+02] J. L. Marroquin, B. C. Vemuri, S. Botello, F. Calderon, and A. Fernandez-Bouzas. An accurate and efficient bayesian method for automatic segmentation of brain MRI. *IEEE Transactions on Medical Imaging*, 21:934–945, 2002.

[NBG+01] A. Nabavi, P. Black, D. Gering, C. Westin, V. Mehta, R. Pergolizzi, M. Ferrant, S. Warfield, N. Hata, R. Schwartz, W. Kikinis, and F. Jolesz. Serial intraoperative MR imaging of brain shift. *Neurosurgery*, 48:787–798, April 2001.

[Owe98] S. Owen. A survey of unstructured mesh generation technology. In *Proceedings 7th International Meshing Roundtable*, pages 239–267, 1998.

[PFY+96] S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V.E Johnson, and E. L. Chaney. Segmentation, registration and measurement of shape variation via image object shape. *IEEE Trans. on Medical Imaging*, 18(10):851–865, 1996.

[PGJA02] S. M. Pizer, G. Gerig, S. Joshi, and S. R. Aylward. Multiscale medial shape-based analysis of image objects. In *Proceedings IEEE, to appear*, 2002.

[PGS+98] C. M. Portas, J. M. Goldstein, M. E. Shenton, H. H. Hokama, C. G. Wible, I. Fischer, R. Kikinis, R. Donnino, F. A. Jolesz, and R. W. McCarley. Volumetric evaluation of the thalamus in schizophrenic male patients using magnetic resonance imaging. *Biological Psychiatry*, 43:649 – 659, 1998.

[Phi97] J. R. Phillips. *Rapid solution of potential integral equations in complicated 3-dimensional geometries.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1997.

[PMV00] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Multi-modality image registration by maximization of mutual information. In *MICCAI*, pages 567–578, 2000.

[Poh03] K. Pohl. (Graduate Student, MIT Artificial Intelligence Lab, Cambridge, Ma.) Personal Communication, 2003.

[POS+01] X. Papademetris, E. Turan Onat, A. J. Sinusas, D. P. Dione, R. T. Constable, and J. S. Duncan. The active elastic model. In *IPMI*, pages 36–49, Davis, CA, June 2001.

[PP99] D. L. Pham and J. L. Prince. Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18:737 – 752, 1999.

[PWG+02] K. M Pohl, W. M. Wells III, A. Guimond, K. Kasai, M. E. Shenton, R. Kikinis, W. E. L. Grimson, and S. K. Warfield. Incoperating non-rigid registration into expectation maximization algorithm to segment MR images. In *MICCAI*, pages 564–572, 2002.

[RHS+98] D. Rueckert, C. Hayes, C. Studholme, P. Summers, M. Leach, and D. J. Hawkes. Non-rigid registration of breast MR images using mutual information. *MICCAI*, 1496:1144–1152, 1998.

[RM98] D. Ruprecht and H. Müller. A scheme for edge-based adaptive tetrahedron subdivision. In Hans-Christian Hege and Konrad Polthier, editors, *Mathematical Visualization*, pages 61–70. Springer Verlag, Heidelberg, 1998.

[Ros02] P. Ross. (Clinical Radiologist, Brigham and Women's Hospital, Boston, Ma.) Personal Communication, 2002.

[SD99] O. S. Skrinjar and J. S. Duncan. Real time 3D brain shift compensation. In *IPMI*, volume 3, pages 42–55, Visegrad, Hungary, July 1999. Springer-Verlag.

[SD02] D. Shen and C. Davatzikos. HAMMER: Hierarchical attribute matching mechanism for elastic registration. *IEEE Trans. on Medical Imaging*, 21(11):1421–1439, November 2002.

[SFdC+95] M. Shephard, J. Flaherty, H. de Cougny, C. Ozturan, C. Bottasso, and M. Beall. Parallel automated adaptive procedures for unstructured meshes, 1995.

[SGA+84] N. W. Shock, R. C. Greulich, R. Andres, D. Arenberg, P. T. Costa Jr., E. Lakatta, and J. D. Tobin. Normal Human Aging: The Baltimore Logitudinal Study of Aging. U. S. Printing Office. Washington, D.C., 1984.

[SKJ+92] M. E. Shenton, R. Kikinis, F. A. Jolesz, S. D. Pollak, M. LeMay, C. G. Wible, H. Hokama, J. Martin, D. Metcalf, M. Coleman, and R. W. McCarley. Abnormalities in the left temporal lobe and thought disorder in schizophrenia: A quantitative magnetic resonance imaging study. *New England Journal of Medicine*, 327:604–612, 1992.

[SMLT98] W. Schroeder, K. Martin, B. Lorensen, and V. Toolkit. An object-oriented approach to 3D graphics. The Visualization Toolkit, 1998.

[SRQ+01] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G. Hill, and D. J. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In *MICCAI*, pages 573–581, 2001.

[SU00] P. K. Saha and J. K. Udupa. Iterative relative fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Mathematical Methods in Biomedical Image Analysis*, 2000.

[TGG+02] S. J. Timoner, P. Golland, W. E. L. Grimson, R. Kikinis, and W. M. Wells III. Performance issues in shape classification. In *MICCAI*, Tokyo, September 2002.

[TGKW01] S. J. Timoner, W. E. L. Grimson, R. Kikinis, and W. M. Wells III. Fast linear elastic matching without landmarks. In *MICCAI*, pages 1358–60, Utretcht, October 2001.

[Thi98] J.-P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[TYW+01] A. Tsai, A. Yezzi, W. M. Wells III, C. Tempany, A. Fan, W. E. L. Grimson, and A. Willsky. Model-based curve evolution techniques for image segmentation. In *CVPR*, 2001.

[WGKJ96] W. M. Wells III, W. E. L. Grimson, R. Kikinis, and F.A Jolesz. Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, 15:429–442, 1996.

[Wie03] L. Wiegand. (Research Assistant and Expert Segmenter, Brigham and Women's Hospital, Boston, Ma.) Personal Communication, 2003.

[WRD+98] S. Warfield, A. Robatino, J. Dengler, F. Jolesz, and R. Kikinis. *Nonlinear Registration and Template Driven Segmentation.* Progressive Publishing Alternatives, 1998.

[WRH+01] S. K. Warfield, J. Rexilius, P. S. Huppi, T. E. Inder, E. G. Miller, W. M. Wells III, G. P. Zientara, F. A. Jolesz, and R. Kikinis. A binary entropy measure to assess nonrigid registration algorithm. In *MICCAI*, pages 266–274, October 2001.

[WRK+00] S. K. Warfield, J. Rexilius, M. Kaus, F. A. Jolesz, and R. Kikinis. Adaptive, template moderated, spatial varying statistical classification. *Medical Image Analysis*, 4(1):43–55, 2000.

[WS98] Y. Wang and L. H. Staib. Elastic model based non-rigid registration incorporating statistical shape information. In *MICCAI*, pages 1162–1173, Cambridge, MA, October 1998.

[WVK95] W. M. Wells III, P. Viola, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. In *Symposium on Medical Robotics and Computer Assisted Surgery*, pages 55–62, 1995.

[ZBS01] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20:45–57, 2001.

[ZFW03] L. Zollei, J. W. Fisher, and W. M. Wells III. A unified statistical and information theoretic framework for multi-modal image registration. In *IPMI*, page Accepted for publication., June 2003.

[ZT89] O. C. Zenkiewicz and R. L. Taylor. *The Finite Element Method*, volume 1. McGraw-Hill, Berkshire, England, Fourth edition, 1989.